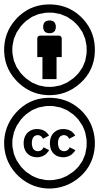
THE OPEN LOGIC TEXT

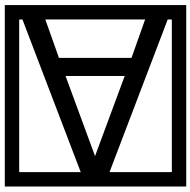
Debug Build

Open Logic Project

Revision: 9f12419 (master)
2026-05-25



The Open Logic Text by the Open Logic Project is licensed under a Creative Commons Attribution 4.0 International License.



Contents

I Naïve Set Theory **20**

1 Sets **20**
1.1 Extensionality 20
1.2 Subsets and Power Sets 21
1.3 Some Important Sets 23
1.4 Unions and Intersections 24
1.5 Pairs, Tuples, Cartesian Products 27
1.6 Russell's Paradox 28

2 Relations **30**
2.1 Relations as Sets 30
2.2 Philosophical Reflections 32
2.3 Special Properties of Relations 33
2.4 Equivalence Relations 34
2.5 Orders 35
2.6 Graphs 37
2.7 Trees 38
2.8 Operations on Relations 40

3 Functions **41**
3.1 Basics 41
3.2 Kinds of Functions 43
3.3 Functions as Relations 45
3.4 Inverses of Functions 46
3.5 Composition of Functions 48
3.6 Partial Functions 49

4 The Size of Sets **50**
4.1 Introduction 51
4.2 Enumerations and Enumerable Sets 51
4.3 Cantor's Zig-Zag Method 55
4.4 Pairing Functions and Codes 57
4.5 An Alternative Pairing Function 58
4.6 Non-enumerable Sets 60
4.7 Reduction 63

4.8 Equinumerosity . . . 65
4.9 Sets of Different Sizes, and Cantor's Theorem . . . 66
4.10 The Notion of Size, and Schröder-Bernstein . . . 68
4.11 Enumerations and Enumerable Sets . . . 68
4.12 Non-enumerable Sets . . . 70
4.13 Reduction . . . 72

5 Arithmetization **74**
5.1 From $\mathbb{N}$ to $\mathbb{Z}$ . . . 74
5.2 From $\mathbb{Z}$ to $\mathbb{Q}$ . . . 76
5.3 The Real Line . . . 77
5.4 From $\mathbb{Q}$ to $\mathbb{R}$ . . . 79
5.5 Some Philosophical Reflections . . . 80
5.6 Ordered Rings and Fields . . . 82
5.7 Appendix: the Reals as Cauchy Sequences . . . 85

6 Infinite Sets **89**
6.1 Hilbert's Hotel . . . 89
6.2 Dedekind Algebras . . . 90
6.3 Arithmetical Induction . . . 92
6.4 Dedekind's "Proof" . . . 93
6.5 Appendix: Proving Schröder-Bernstein . . . 95

II Propositional Logic **97**

7 Syntax and Semantics **97**
7.1 Introduction . . . 97
7.2 Propositional Formulas . . . 99
7.3 Preliminaries . . . 100
7.4 Formation Sequences . . . 102
7.5 Valuations and Satisfaction . . . 104
7.6 Semantic Notions . . . 106

8 Derivation Systems **107**
8.1 Introduction . . . 107
8.2 The Sequent Calculus . . . 109
8.3 Natural Deduction . . . 110
8.4 Tableaux . . . 111
8.5 Axiomatic Derivations . . . 112

9 The Sequent Calculus **114**
9.1 Rules and Derivations . . . 114
9.2 Propositional Rules . . . 115
9.3 Structural Rules . . . 116
9.4 Derivations . . . 117

9.5 Examples of Derivations . . . 119
9.6 Proof-Theoretic Notions . . . 123
9.7 Derivability and Consistency . . . 125
9.8 Derivability and the Propositional Connectives . . . 126
9.9 Soundness . . . 128

10 Natural Deduction **132**
10.1 Rules and Derivations . . . 132
10.2 Propositional Rules . . . 133
10.3 Derivations . . . 134
10.4 Examples of Derivations . . . 136
10.5 Proof-Theoretic Notions . . . 141
10.6 Derivability and Consistency . . . 142
10.7 Derivability and the Propositional Connectives . . . 144
10.8 Soundness . . . 145

11 Tableaux **149**
11.1 Rules and Tableaux . . . 149
11.2 Propositional Rules . . . 150
11.3 Tableaux . . . 151
11.4 Examples of Tableaux . . . 152
11.5 Proof-Theoretic Notions . . . 157
11.6 Derivability and Consistency . . . 160
11.7 Derivability and the Propositional Connectives . . . 161
11.8 Soundness . . . 164

12 Axiomatic Derivations **166**
12.1 Rules and Derivations . . . 166
12.2 Axioms and Rules for the Propositional Connectives . . . 168
12.3 Examples of Derivations . . . 168
12.4 Proof-Theoretic Notions . . . 170
12.5 The Deduction Theorem . . . 172
12.6 Derivability and Consistency . . . 173
12.7 Derivability and the Propositional Connectives . . . 174
12.8 Soundness . . . 175

13 The Completeness Theorem **176**
13.1 Introduction . . . 176
13.2 Outline of the Proof . . . 177
13.3 Complete Consistent Sets of Sentences . . . 178
13.4 Lindenbaum's Lemma . . . 180
13.5 Construction of a Model . . . 181
13.6 The Completeness Theorem . . . 181
13.7 The Compactness Theorem . . . 182
13.8 A Direct Proof of the Compactness Theorem . . . 183

III First-order Logic **185**

14 Introduction to First-Order Logic **185**
14.1 First-Order Logic . . . 185
14.2 Syntax . . . 187
14.3 Formulas . . . 187
14.4 Satisfaction . . . 189
14.5 Sentences . . . 190
14.6 Semantic Notions . . . 191
14.7 Substitution . . . 192
14.8 Models and Theories . . . 192
14.9 Soundness and Completeness . . . 193

15 Syntax of First-Order Logic **194**
15.1 Introduction . . . 195
15.2 First-Order Languages . . . 195
15.3 Terms and Formulas . . . 197
15.4 Unique Readability . . . 199
15.5 Main operator of a Formula . . . 202
15.6 Subformulas . . . 203
15.7 Formation Sequences . . . 205
15.8 Free Variables and Sentences . . . 208
15.9 Substitution . . . 209

16 Semantics of First-Order Logic **211**
16.1 Introduction . . . 211
16.2 Structures for First-order Languages . . . 212
16.3 Covered Structures for First-order Languages . . . 213
16.4 Satisfaction of a Formula in a Structure . . . 214
16.5 Variable Assignments . . . 219
16.6 Extensionality . . . 223
16.7 Semantic Notions . . . 225

17 Theories and Their Models **227**
17.1 Introduction . . . 227
17.2 Expressing Properties of Structures . . . 229
17.3 Examples of First-Order Theories . . . 229
17.4 Expressing Relations in a Structure . . . 232
17.5 The Theory of Sets . . . 234
17.6 Expressing the Size of Structures . . . 236

18 Derivation Systems **237**
18.1 Introduction . . . 238
18.2 The Sequent Calculus . . . 239
18.3 Natural Deduction . . . 240
18.4 Tableaux . . . 241

18.5 Axiomatic Derivations 243

19 The Sequent Calculus **244**
19.1 Rules and Derivations 244
19.2 Propositional Rules 245
19.3 Quantifier Rules 246
19.4 Structural Rules 247
19.5 Derivations 248
19.6 Examples of Derivations 250
19.7 Derivations with Quantifiers 255
19.8 Proof-Theoretic Notions 256
19.9 Derivability and Consistency 258
19.10 Derivability and the Propositional Connectives 260
19.11 Derivability and the Quantifiers 261
19.12 Soundness 262
19.13 Derivations with Identity predicate 267
19.14 Soundness with Identity predicate 267

20 Natural Deduction **268**
20.1 Rules and Derivations 268
20.2 Propositional Rules 269
20.3 Quantifier Rules 270
20.4 Derivations 272
20.5 Examples of Derivations 273
20.6 Derivations with Quantifiers 278
20.7 Proof-Theoretic Notions 282
20.8 Derivability and Consistency 284
20.9 Derivability and the Propositional Connectives 285
20.10 Derivability and the Quantifiers 287
20.11 Soundness 287
20.12 Derivations with Identity predicate 291
20.13 Soundness with Identity predicate 293

21 Tableaux **294**
21.1 Rules and Tableaux 294
21.2 Propositional Rules 295
21.3 Quantifier Rules 296
21.4 Tableaux 297
21.5 Examples of Tableaux 298
21.6 Tableaux with Quantifiers 303
21.7 Proof-Theoretic Notions 307
21.8 Derivability and Consistency 310
21.9 Derivability and the Propositional Connectives 311
21.10 Derivability and the Quantifiers 314
21.11 Soundness 315
21.12 Tableaux with Identity predicate 318

21.13 Soundness with Identity predicate . . . 319

22 Axiomatic Derivations 320
22.1 Rules and Derivations . . . 320
22.2 Axioms and Rules for the Propositional Connectives . . . 322
22.3 Axioms and Rules for Quantifiers . . . 322
22.4 Examples of Derivations . . . 323
22.5 Derivations with Quantifiers . . . 325
22.6 Proof-Theoretic Notions . . . 325
22.7 The Deduction Theorem . . . 327
22.8 The Deduction Theorem with Quantifiers . . . 328
22.9 Derivability and Consistency . . . 330
22.10 Derivability and the Propositional Connectives . . . 330
22.11 Derivability and the Quantifiers . . . 331
22.12 Soundness . . . 332
22.13 Derivations with Identity predicate . . . 333

23 The Completeness Theorem 334
23.1 Introduction . . . 334
23.2 Outline of the Proof . . . 335
23.3 Complete Consistent Sets of Sentences . . . 337
23.4 Henkin Expansion . . . 339
23.5 Lindenbaum's Lemma . . . 341
23.6 Construction of a Model . . . 342
23.7 Identity . . . 344
23.8 The Completeness Theorem . . . 347
23.9 The Compactness Theorem . . . 348
23.10 A Direct Proof of the Compactness Theorem . . . 350
23.11 The Löwenheim–Skolem Theorem . . . 351

24 Beyond First-order Logic 352
24.1 Overview . . . 353
24.2 Many-Sorted Logic . . . 353
24.3 Second-Order logic . . . 355
24.4 Higher-Order logic . . . 358
24.5 Intuitionistic Logic . . . 361
24.6 Modal Logics . . . 364
24.7 Other Logics . . . 366

IV Model Theory 367

25 Basics of Model Theory 367
25.1 Reducts and Expansions . . . 367
25.2 Substructures . . . 368
25.3 Overspill . . . 369

25.4 Isomorphic Structures . . . 369
25.5 The Theory of a Structure . . . 371
25.6 Partial Isomorphisms . . . 372
25.7 Dense Linear Orders . . . 374

26 Models of Arithmetic **376**
26.1 Introduction . . . 376
26.2 Standard Models of Arithmetic . . . 377
26.3 Non-Standard Models . . . 379
26.4 Models of **Q** . . . 381
26.5 Models of **PA** . . . 383
26.6 Computable Models of Arithmetic . . . 386

27 The Interpolation Theorem **388**
27.1 Introduction . . . 388
27.2 Separation of Sentences . . . 388
27.3 Craig's Interpolation Theorem . . . 390
27.4 The Definability Theorem . . . 392

28 Lindström's Theorem **394**
28.1 Introduction . . . 394
28.2 Abstract Logics . . . 394
28.3 Compactness and Löwenheim–Skolem Properties . . . 396
28.4 Lindström's Theorem . . . 398

V Computability **400**

29 Recursive Functions **400**
29.1 Introduction . . . 400
29.2 Primitive Recursion . . . 401
29.3 Composition . . . 403
29.4 Primitive Recursion Functions . . . 405
29.5 Primitive Recursion Notations . . . 407
29.6 Primitive Recursive Functions are Computable . . . 408
29.7 Examples of Primitive Recursive Functions . . . 409
29.8 Primitive Recursive Relations . . . 412
29.9 Bounded Minimization . . . 414
29.10 Primes . . . 415
29.11 Sequences . . . 416
29.12 Trees . . . 419
29.13 Other Recursions . . . 420
29.14 Non-Primitive Recursive Functions . . . 421
29.15 Partial Recursive Functions . . . 422
29.16 The Normal Form Theorem . . . 424
29.17 The Halting Problem . . . 425

29.18 General Recursive Functions 426

30 Computability Theory **427**
30.1 Introduction 427
30.2 Coding Computations 428
30.3 The Normal Form Theorem 429
30.4 The *s-m-n* Theorem 430
30.5 The Universal Partial Computable Function 431
30.6 No Universal Computable Function 432
30.7 The Halting Problem 432
30.8 Comparison with Russell's Paradox 434
30.9 Computable Sets 435
30.10 Computably Enumerable Sets 435
30.11 Definitions of C. E. Sets 436
30.12 There Are Non-Computable Sets 438
30.13 Union and Intersection of C.E. Sets 439
30.14 Computably Enumerable Sets not Closed under Complement . 440
30.15 Reducibility 441
30.16 Properties of Reducibility 442
30.17 Complete Computably Enumerable Sets 443
30.18 An Example of Reducibility 444
30.19 Totality is Undecidable 445
30.20 Rice's Theorem 446
30.21 The Fixed-Point Theorem 448
30.22 Applying the Fixed-Point Theorem 451
30.23 Defining Functions using Self-Reference 452

VI Turing Machines **454**

31 Turing Machine Computations **454**
31.1 Introduction 454
31.2 Representing Turing Machines 456
31.3 Turing Machines 460
31.4 Configurations and Computations 461
31.5 Unary Representation of Numbers 463
31.6 Halting States 466
31.7 Disciplined Machines 467
31.8 Combining Turing Machines 469
31.9 Variants of Turing Machines 471
31.10 The Church–Turing Thesis 472

32 Undecidability **473**
32.1 Introduction 474
32.2 Enumerating Turing Machines 475
32.3 Universal Turing Machines 477

32.4 The Halting Problem 480
32.5 The Decision Problem 482
32.6 Representing Turing Machines 482
32.7 Verifying the Representation 485
32.8 The Decision Problem is Unsolvable 490
32.9 Trakhtenbrot's Theorem 491

VII Incompleteness **495**

33 Introduction to Incompleteness **495**
33.1 Historical Background 495
33.2 Definitions 499
33.3 Overview of Incompleteness Results 504
33.4 Undecidability and Incompleteness 505

34 Arithmetization of Syntax **507**
34.1 Introduction 507
34.2 Coding Symbols 509
34.3 Coding Terms 510
34.4 Coding Formulas 512
34.5 Substitution 513
34.6 Derivations in **LK** 514
34.7 Derivations in Natural Deduction 517
34.8 Axiomatic Derivations 522

35 Representability in Q **526**
35.1 Introduction 526
35.2 Functions Representable in **Q** are Computable 528
35.3 The Beta Function Lemma 529
35.4 Simulating Primitive Recursion 532
35.5 Basic Functions are Representable in **Q** 533
35.6 Composition is Representable in **Q** 536
35.7 Regular Minimization is Representable in **Q** 537
35.8 Computable Functions are Representable in **Q** 540
35.9 Representing Relations 541
35.10 Undecidability 542
35.11 Σ_1 completeness 543

36 Theories and Computability **546**
36.1 Introduction 546
36.2 **Q** is C.e.-Complete 547
36.3 ω-Consistent Extensions of **Q** are Undecidable 548
36.4 Consistent Extensions of **Q** are Undecidable 548
36.5 Axiomatizable Theories 549
36.6 Axiomatizable Complete Theories are Decidable 550

36.7 **Q** has no Complete, Consistent, Axiomatizable Extensions . . 550
36.8 Sentences Provable and Refutable in **Q** are Computably Inseparable 551
36.9 Theories Consistent with **Q** are Undecidable 552
36.10 Theories in which **Q** is Interpretable are Undecidable 552

37 Incompleteness and Provability 553
37.1 Introduction 554
37.2 The Fixed-Point Lemma 555
37.3 The First Incompleteness Theorem 557
37.4 Rosser's Theorem 559
37.5 Comparison with Gödel's Original Paper 560
37.6 The Derivability Conditions for **PA** 561
37.7 The Second Incompleteness Theorem 562
37.8 Löb's Theorem 564
37.9 The Undefinability of Truth 567

VIII Second-order Logic 569

38 Syntax and Semantics 569
38.1 Introduction 569
38.2 Terms and Formulas 570
38.3 Satisfaction 571
38.4 Semantic Notions 574
38.5 Expressive Power 574
38.6 Describing Infinite and Enumerable Domains 576

39 Metatheory of Second-order Logic 577
39.1 Introduction 578
39.2 Second-order Arithmetic 578
39.3 Second-order Logic is not Axiomatizable 580
39.4 Second-order Logic is not Compact 581
39.5 The Löwenheim–Skolem Theorem Fails for Second-order Logic 581

40 Second-order Logic and Set Theory 582
40.1 Introduction 582
40.2 Comparing Sets 583
40.3 Cardinalities of Sets 584
40.4 The Power of the Continuum 585

IX The Lambda Calculus 588

41 Introduction 588
41.1 Overview 588
41.2 The Syntax of the Lambda Calculus 589

41.3 Reduction of Lambda Terms . . . 591
41.4 The Church–Rosser Property . . . 592
41.5 Currying . . . 592
41.6 λ-Definable Arithmetical Functions . . . 593
41.7 λ-Definable Functions are Computable . . . 594
41.8 Computable Functions are λ-Definable . . . 594
41.9 The Basic Primitive Recursive Functions are λ-Definable . . . 595
41.10 The λ-Definable Functions are Closed under Composition . . . 595
41.11 λ-Definable Functions are Closed under Primitive Recursion . 596
41.12 Fixed-Point Combinators . . . 598
41.13 The λ-Definable Functions are Closed under Minimization . . 599

42 Syntax 600
42.1 Terms . . . 600
42.2 Unique Readability . . . 601
42.3 Abbreviated Syntax . . . 602
42.4 Free Variables . . . 603
42.5 Substitution . . . 604
42.6 α-Conversion . . . 607
42.7 The De Bruijn Index . . . 611
42.8 Terms as α-Equivalence Classes . . . 612
42.9 β-reduction . . . 613
42.10 η-conversion . . . 615

43 The Church–Rosser Property 616
43.1 Definition and Properties . . . 617
43.2 Parallel β-reduction . . . 618
43.3 β-reduction . . . 620
43.4 Parallel $\beta\eta$-reduction . . . 621
43.5 $\beta\eta$-reduction . . . 623

44 Lambda Definability 624
44.1 Introduction . . . 624
44.2 λ-Definable Arithmetical Functions . . . 625
44.3 Pairs and Predecessor . . . 628
44.4 Truth Values and Relations . . . 628
44.5 Primitive Recursive Functions are λ-Definable . . . 630
44.6 Fixpoints . . . 632
44.7 Minimization . . . 635
44.8 Partial Recursive Functions are λ-Definable . . . 636
44.9 λ-Definable Functions are Recursive . . . 636

X Many-valued Logic 638

45 Syntax and Semantics 638

45.1 Introduction . . . 638
45.2 Languages and Connectives . . . 639
45.3 Formulas . . . 640
45.4 Matrices . . . 640
45.5 Valuations and Satisfaction . . . 641
45.6 Semantic Notions . . . 642
45.7 Many-valued logics as sublogics of **C** . . . 642

46 Three-valued Logics **644**
46.1 Introduction . . . 644
46.2 Łukasiewicz logic . . . 644
46.3 Kleene logics . . . 648
46.4 Gödel logics . . . 650
46.5 Designating not just $\mathbb{T}$ . . . 652

47 Infinite-valued Logics **655**
47.1 Introduction . . . 655
47.2 Łukasiewicz logic . . . 656
47.3 Gödel logics . . . 657

48 Sequent Calculus **658**
48.1 Introduction . . . 659
48.2 Rules and Derivations . . . 660
48.3 Structural Rules . . . 661
48.4 Propositional Rules for Selected Logics . . . 661

XI Normal Modal Logics **665**

49 Syntax and Semantics **665**
49.1 Introduction . . . 665
49.2 The Language of Basic Modal Logic . . . 667
49.3 Simultaneous Substitution . . . 668
49.4 Relational Models . . . 669
49.5 Truth at a World . . . 670
49.6 Truth in a Model . . . 672
49.7 Validity . . . 673
49.8 Tautological Instances . . . 674
49.9 Schemas and Validity . . . 676
49.10 Entailment . . . 678

50 Frame Definability **679**
50.1 Introduction . . . 680
50.2 Properties of Accessibility Relations . . . 680
50.3 Frames . . . 683
50.4 Frame Definability . . . 683

50.5 First-order Definability . . . 686
50.6 Equivalence Relations and **S5** . . . 687
50.7 Second-order Definability . . . 689

51 Axiomatic Derivations **691**
51.1 Introduction . . . 691
51.2 Normal Modal Logics . . . 693
51.3 Derivations and Modal Systems . . . 694
51.4 Proofs in **K** . . . 696
51.5 Derived Rules . . . 698
51.6 More Proofs in **K** . . . 700
51.7 Dual Formulas . . . 701
51.8 Proofs in Modal Systems . . . 701
51.9 Soundness . . . 703
51.10 Showing Systems are Distinct . . . 704
51.11 Derivability from a Set of Formulas . . . 705
51.12 Properties of Derivability . . . 706
51.13 Consistency . . . 706

52 Completeness and Canonical Models **707**
52.1 Introduction . . . 707
52.2 Complete Σ-Consistent Sets . . . 708
52.3 Lindenbaum's Lemma . . . 710
52.4 Modalities and Complete Consistent Sets . . . 711
52.5 Canonical Models . . . 713
52.6 The Truth Lemma . . . 713
52.7 Determination and Completeness for **K** . . . 715
52.8 Frame Completeness . . . 715

53 Filtrations and Decidability **718**
53.1 Introduction . . . 719
53.2 Preliminaries . . . 721
53.3 Filtrations . . . 722
53.4 Examples of Filtrations . . . 724
53.5 Filtrations are Finite . . . 726
53.6 **K** and **S5** have the Finite Model Property . . . 727
53.7 **S5** is Decidable . . . 728
53.8 Filtrations and Properties of Accessibility . . . 728
53.9 Filtrations of Euclidean Models . . . 730

54 Modal Tableaux **731**
54.1 Introduction . . . 731
54.2 Rules for K . . . 732
54.3 Tableaux for **K** . . . 734
54.4 Soundness for **K** . . . 735
54.5 Rules for Other Accessibility Relations . . . 738

54.6 Soundness for Additional Rules . . . 740
54.7 Simple Tableaux for **S5** . . . 742
54.8 Completeness for **K** . . . 743
54.9 Countermodels from Tableaux . . . 746

55 Modal Sequent Calculus **748**
55.1 Introduction . . . 748
55.2 Rules for **K** . . . 748
55.3 Sequent Derivations for **K** . . . 749
55.4 Rules for Other Accessibility Relations . . . 751

XII Applied Modal Logic **753**

56 Temporal Logics **753**
56.1 Introduction . . . 753
56.2 Semantics for Temporal Logic . . . 754
56.3 Properties of Temporal Frames . . . 756
56.4 Additional Operators for Temporal Logic . . . 757
56.5 Possible Histories . . . 757

57 Epistemic Logics **759**
57.1 Introduction . . . 759
57.2 The Language of Epistemic Logic . . . 760
57.3 Relational Models . . . 761
57.4 Truth at a World . . . 762
57.5 Accessibility Relations and Epistemic Principles . . . 764
57.6 Bisimulations . . . 765
57.7 Public Announcement Logic . . . 766
57.8 Semantics of Public Announcement Logic . . . 768

XIII Intuitionistic Logic **770**

58 Introduction **770**
58.1 Constructive Reasoning . . . 770
58.2 Syntax of Intuitionistic Logic . . . 771
58.3 The Brouwer–Heyting–Kolmogorov Interpretation . . . 773
58.4 Natural Deduction . . . 775
58.5 Axiomatic Derivations . . . 778

59 Semantics **779**
59.1 Introduction . . . 780
59.2 Relational models . . . 781
59.3 Semantic Notions . . . 782
59.4 Topological Semantics . . . 783

60 Soundness and Completeness **784**
60.1 Soundness of Axiomatic Derivations . . . 784
60.2 Soundness of Natural Deduction . . . 785
60.3 Lindenbaum's Lemma . . . 787
60.4 The Canonical Model . . . 789
60.5 The Truth Lemma . . . 790
60.6 The Completeness Theorem . . . 791
60.7 Decidability . . . 791

61 Intuitionistic Tableaux **792**
61.1 Introduction . . . 792
61.2 Rules for Intuitionistic Logic . . . 793
61.3 Tableaux for Intuitionistic Logic . . . 795
61.4 Soundness for Intuitionistic Tableaux . . . 795

XIVCounterfactuals **799**

62 Introduction **799**
62.1 The Material Conditional . . . 799
62.2 Paradoxes of the Material Conditional . . . 801
62.3 The Strict Conditional . . . 801
62.4 Counterfactuals . . . 804

63 Minimal Change Semantics **805**
63.1 Introduction . . . 805
63.2 Sphere Models . . . 806
63.3 Truth and Falsity of Counterfactuals . . . 808
63.4 Antecedent Strengthening . . . 810
63.5 Transitivity . . . 811
63.6 Contraposition . . . 812

XV Set Theory **814**

64 The Iterative Conception **814**
64.1 Extensionality . . . 814
64.2 Russell's Paradox (again) . . . 814
64.3 Predicative and Impredicative . . . 816
64.4 The Cumulative-Iterative Approach . . . 817
64.5 Urelements or Not? . . . 819
64.6 Appendix: Frege's Basic Law V . . . 820

65 Steps towards Z **821**
65.1 The Story in More Detail . . . 821
65.2 Separation . . . 822
65.3 Union . . . 823

65.4 Pairs . . . 824
65.5 Powersets . . . 825
65.6 Infinity . . . 825
65.7 $\mathbf{Z}^-$: a Milestone . . . 827
65.8 Selecting our Natural Numbers . . . 828
65.9 Appendix: Closure, Comprehension, and Intersection . . . 829

66 Ordinals **830**
66.1 Introduction . . . 830
66.2 The General Idea of an Ordinal . . . 830
66.3 Well-Orderings . . . 831
66.4 Order-Isomorphisms . . . 832
66.5 Von Neumann's Construction . . . 834
66.6 Basic Properties of the Ordinals . . . 835
66.7 Replacement . . . 838
66.8 $\mathbf{ZF}^-$: a milestone . . . 839
66.9 Ordinals as Order-Types . . . 839
66.10 Successor and Limit Ordinals . . . 840

67 Stages and Ranks **842**
67.1 Defining the Stages as the V_αs . . . 842
67.2 The Transfinite Recursion Theorem(s) . . . 843
67.3 Basic Properties of Stages . . . 845
67.4 Foundation . . . 846
67.5 **Z** and **ZF**: A Milestone . . . 847
67.6 Rank . . . 848

68 Replacement **850**
68.1 Introduction . . . 850
68.2 The Strength of Replacement . . . 850
68.3 Extrinsic Considerations . . . 851
68.4 Limitation-of-size . . . 853
68.5 Replacement and "Absolute Infinity" . . . 854
68.6 Replacement and Reflection . . . 856
68.7 Appendix: Results surrounding Replacement . . . 856
68.8 Appendix: Finite axiomatizability . . . 859

69 Ordinal Arithmetic **861**
69.1 Introduction . . . 861
69.2 Ordinal Addition . . . 861
69.3 Using Ordinal Addition . . . 864
69.4 Ordinal Multiplication . . . 866
69.5 Ordinal Exponentiation . . . 867

70 Cardinals **868**
70.1 Cantor's Principle . . . 868

70.2 Cardinals as Ordinals . . . 869
70.3 **ZFC**: A Milestone . . . 871
70.4 Finite, Enumerable, Non-enumerable . . . 871
70.5 Appendix: Hume's Principle . . . 873

71 Cardinal Arithmetic **875**
71.1 Defining the Basic Operations . . . 875
71.2 Simplifying Addition and Multiplication . . . 877
71.3 Some Simplifications . . . 879
71.4 The Continuum Hypothesis . . . 880
71.5 ℵ-Fixed Points . . . 882

72 Choice **884**
72.1 Introduction . . . 884
72.2 The Tarski–Scott Trick . . . 884
72.3 Comparability and Hartogs' Lemma . . . 885
72.4 The Well-Ordering Problem . . . 887
72.5 Countable Choice . . . 888
72.6 Intrinsic Considerations about Choice . . . 890
72.7 The Banach–Tarski Paradox . . . 891
72.8 Appendix: Vitali's Paradox . . . 893

XVI Methods **897**

73 Proofs **897**
73.1 Introduction . . . 897
73.2 Starting a Proof . . . 898
73.3 Using Definitions . . . 899
73.4 Inference Patterns . . . 900
73.5 An Example . . . 906
73.6 Another Example . . . 909
73.7 Proof by Contradiction . . . 911
73.8 Reading Proofs . . . 914
73.9 I Can't Do It! . . . 916
73.10 Other Resources . . . 917

74 Induction **917**
74.1 Introduction . . . 917
74.2 Induction on ℕ . . . 918
74.3 Strong Induction . . . 920
74.4 Inductive Definitions . . . 921
74.5 Structural Induction . . . 924
74.6 Relations and Functions . . . 925

XVIII History **928**

75 Biographies **928**
75.1 Georg Cantor . . . 928
75.2 Alonzo Church . . . 929
75.3 Gerhard Gentzen . . . 930
75.4 Kurt Gödel . . . 931
75.5 Emmy Noether . . . 932
75.6 Rózsa Péter . . . 934
75.7 Julia Robinson . . . 935
75.8 Bertrand Russell . . . 937
75.9 Alfred Tarski . . . 938
75.10 Alan Turing . . . 939
75.11 Ernst Zermelo . . . 940

76 History and Mythology of Set Theory **942**
76.1 Infinitesimals and Differentiation . . . 942
76.2 Rigorous Definition of Limits . . . 944
76.3 Pathologies . . . 945
76.4 More Myth than History? . . . 947
76.5 Cantor on the Line and the Plane . . . 948
76.6 Appendix: Hilbert's Space-filling Curves . . . 949

XVIII Reference **952**

77 The Greek Alphabet **952**

78 The Fraktur Alphabet **953**

Photo Credits **953**

Bibliography **955**

This file loads all content included in the Open Logic Project. Editorial notes like this, if displayed, indicate that the file was compiled without any thought to how this material will be presented. If you can read this, it is probably *not advisable* to teach or study from this PDF.

The Open Logic Project provides many mechanisms by which a text can be generate which is more appropriate for teaching or self-study. For instance, by default, the text will make all logical operators primitives and carry out all cases for all operators in proofs. But it is much better to leave some of these cases as exercises. The Open Logic Project is also a work in progress. In an effort to stimulate collaboration and improvement, material is included even if it is only in draft form, is missing exercises, etc. A PDF produced for a course will exclude these sections.

To find PDFs more suitable for teaching and studying, have a look at the sample courses available on the OLP website. To make your own, you might start from the sample driver file or look at the sources of the derived textbooks for more fancy and advanced examples.

Part I

Naïve Set Theory

The material in this part is an introduction to basic naive set theory. With the inclusion of Tim Button's Open Set Theory, this also covers the construction of number systems, and discussion of infinity, which are not required for the logical parts of the OLP.

Chapter 1

Sets

content/sets-functions-relations/sets/basics.tex

1.1 Extensionality

A *set* is a collection of objects, considered as a single object. The objects making up the set are called *elements* or *members* of the set. If x is an element of a set A, we write $x \in A$; if not, we write $x \notin A$. The set which has no elements is called the *empty* set and denoted "$\emptyset$". sfr:set:bas:sec

explanation It does not matter how we *specify* the set, or how we *order* its elements, or indeed how *many times* we count its elements. All that matters are what its elements are. We codify this in the following principle.

Definition 1.1 (Extensionality). If A and B are sets, then $A = B$ iff every element of A is also an element of B, and vice versa.

Extensionality licenses some notation. In general, when we have some objects $a_1, \ldots, a_n$, then $\{a_1, \ldots, a_n\}$ is *the* set whose elements are $a_1, \ldots, a_n$. We emphasise the word "*the*", since extensionality tells us that there can be only *one* such set. Indeed, extensionality also licenses the following:

$$\{a, a, b\} = \{a, b\} = \{b, a\}.$$

This delivers on the point that, when we consider sets, we don't care about the order of their elements, or how many times they are specified.

Example 1.2. Whenever you have a bunch of objects, you can collect them together in a set. The set of Richard's siblings, for instance, is a set that contains one person, and we could write it as $S = \{\text{Ruth}\}$. The set of positive integers less than 4 is $\{1, 2, 3\}$, but it can also be written as $\{3, 2, 1\}$ or even as $\{1, 2, 1, 2, 3\}$. These are all the same set, by extensionality. For every element of $\{1, 2, 3\}$ is also an element of $\{3, 2, 1\}$ (and of $\{1, 2, 1, 2, 3\}$), and vice versa.

Frequently we'll specify a set by some property that its elements share. We'll use the following shorthand notation for that: $\{x : \varphi(x)\}$, where the $\varphi(x)$ stands for the property that x has to have in order to be counted among the elements of the set.

Example 1.3. In our example, we could have specified S also as

$$S = \{x : x \text{ is a sibling of Richard}\}.$$

Example 1.4. A number is called *perfect* iff it is equal to the sum of its proper divisors (i.e., numbers that evenly divide it but aren't identical to the number). For instance, 6 is perfect because its proper divisors are 1, 2, and 3, and $6 = 1 + 2 + 3$. In fact, 6 is the only positive integer less than 10 that is perfect. So, using extensionality, we can say:

$$\{6\} = \{x : x \text{ is perfect and } 0 \leq x \leq 10\}$$

We read the notation on the right as "the set of x's such that x is perfect and $0 \leq x \leq 10$". The identity here confirms that, when we consider sets, we don't care about how they are specified. And, more generally, extensionality guarantees that there is always only one set of x's such that $\varphi(x)$. So, extensionality justifies calling $\{x : \varphi(x)\}$ *the* set of x's such that $\varphi(x)$.

Extensionality gives us a way for showing that sets are identical: to show that $A = B$, show that whenever $x \in A$ then also $x \in B$, and whenever $y \in B$ then also $y \in A$.

Problem 1.1. Prove that there is at most one empty set, i.e., show that if A and B are sets without elements, then $A = B$.

---

content/sets-functions-relations/sets/subsets.tex

1.2 Subsets and Power Sets

sfr:set:sub:sec We will often want to compare sets. And one obvious kind of comparison one might make is as follows: *everything in one set is in the other too.* This situation is sufficiently important for us to introduce some new notation. explanation

Definition 1.5 (Subset). If every element of a set A is also an element of B, then we say that A is a *subset* of B, and write $A \subseteq B$. If A is not a subset of B we write $A \nsubseteq B$. If $A \subseteq B$ but $A \neq B$, we write $A \subsetneq B$ and say that A is a *proper subset* of B.

Example 1.6. Every set is a subset of itself, and $\emptyset$ is a subset of every set. The set of natural even numbers is a subset of the set of natural numbers. Also, $\{a, b\} \subseteq \{a, b, c\}$. But $\{a, b, e\}$ is not a subset of $\{a, b, c\}$.

Example 1.7. The number 2 is an element of the set of integers, whereas the set of even numbers is a subset of the set of integers. However, a set may happen to *both* be an element and a subset of some other set, e.g., $\{0\} \in \{0, \{0\}\}$ and also $\{0\} \subseteq \{0, \{0\}\}$.

Extensionality gives a criterion of identity for sets: $A = B$ iff every element of A is also an element of B and vice versa. The definition of "subset" defines $A \subseteq B$ precisely as the first half of this criterion: every element of A is also an element of B. Of course the definition also applies if we switch A and B: that is, $B \subseteq A$ iff every element of B is also an element of A. And that, in turn, is exactly the "vice versa" part of extensionality. In other words, extensionality entails that sets are equal iff they are subsets of one another.

Proposition 1.8. *$A = B$ iff both $A \subseteq B$ and $B \subseteq A$.*

Now is also a good opportunity to introduce some further bits of helpful notation. In defining when A is a subset of B we said that "every element of A is ...," and filled the "..." with "an element of B". But this is such a common *shape* of expression that it will be helpful to introduce some formal notation for it.

Definition 1.9. $(\forall x \in A)\varphi$ abbreviates $\forall x(x \in A \to \varphi)$. Similarly, $(\exists x \in A)\varphi$ abbreviates $\exists x(x \in A \wedge \varphi)$. sfr:set:sub:forallxina

Using this notation, we can say that $A \subseteq B$ iff $(\forall x \in A)x \in B$.

Now we move on to considering a certain kind of set: the set of all subsets of a given set.

Definition 1.10 (Power Set). The set consisting of all subsets of a set A is called the *power set of* A, written $\wp(A)$.

$$\wp(A) = \{B : B \subseteq A\}$$

Example 1.11. What are all the possible subsets of $\{a, b, c\}$? They are: $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$, $\{a, b, c\}$. The set of all these subsets is $\wp(\{a, b, c\})$:

$$\wp(\{a, b, c\}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}\}$$

Problem 1.2. List all subsets of $\{a, b, c, d\}$.

Problem 1.3. Show that if A has n elements, then $\wp(A)$ has 2^n elements.

---

content/sets-functions-relations/sets/important-sets.tex

1.3 Some Important Sets

sfr:set:imp:sec

Example 1.12. We will mostly be dealing with sets whose elements are mathematical objects. Four such sets are important enough to have specific names:

$$\mathbb{N} = \{0, 1, 2, 3, \ldots\}$$
the set of natural numbers

$$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$$
the set of integers

$$\mathbb{Q} = \{m/n : m, n \in \mathbb{Z} \text{ and } n \neq 0\}$$
the set of rationals

$$\mathbb{R} = (-\infty, \infty)$$
the set of real numbers (the continuum)

These are all *infinite* sets, that is, they each have infinitely many elements.

As we move through these sets, we are adding *more* numbers to our stock. Indeed, it should be clear that $\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}$: after all, every natural number is an integer; every integer is a rational; and every rational is a real. Equally, it should be clear that $\mathbb{N} \subsetneq \mathbb{Z} \subsetneq \mathbb{Q}$, since -1 is an integer but not a natural number, and $1/2$ is rational but not integer. It is less obvious that $\mathbb{Q} \subsetneq \mathbb{R}$, i.e., that there are some real numbers which are not rational.

We'll sometimes also use the set of positive integers $\mathbb{Z}^+ = \{1, 2, 3, \ldots\}$ and the set containing just the first two natural numbers $\mathbb{B} = \{0, 1\}$.

Example 1.13 (Strings). Another interesting example is the set A^* of *finite strings* over an alphabet A: any finite sequence of elements of A is a string over A. We include the *empty string* Λ among the strings over A, for every alphabet A. For instance,

$$\mathbb{B}^* = \{\Lambda, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, \ldots\}.$$

If $x = x_1 \ldots x_n \in A^*$is a string consisting of n "letters" from A, then we say *length* of the string is n and write $\text{len}(x) = n$.

Example 1.14 (Infinite sequences). For any set A we may also consider the set A^ω of infinite sequences of elements of A. An infinite sequence $a_1 a_2 a_3 a_4 \ldots$ consists of a one-way infinite list of objects, each one of which is an element of A.

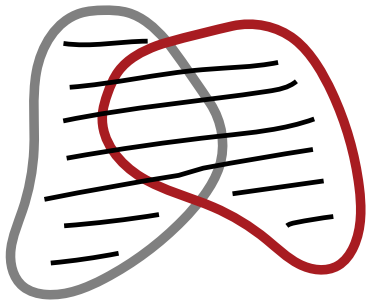


Figure 1.1: The union $A \cup B$ of two sets is set of elements of A together with those of B.

sfr:set:uni:fig:union

---

content/sets-functions-relations/sets/unions-and-intersections.tex

1.4 Unions and Intersections

explanation In section 1.1, we introduced definitions of sets by abstraction, i.e., definitions of the form $\{x : \varphi(x)\}$. Here, we invoke some property φ, and this property can mention sets we've already defined. So for instance, if A and B are sets, the set $\{x : x \in A \vee x \in B\}$ consists of all those objects which are elements of either A or B, i.e., it's the set that combines the elements of A and B. We can visualize this as in Figure 1.1, where the highlighted area indicates the elements of the two sets A and B together. sfr:set:uni:sec

This operation on sets—combining them—is very useful and common, and so we give it a formal name and a symbol.

Definition 1.15 (Union). The *union* of two sets A and B, written $A \cup B$, is the set of all things which are elements of A, B, or both.

$$A \cup B = \{x : x \in A \vee x \in B\}$$

Example 1.16. Since the multiplicity of elements doesn't matter, the union of two sets which have an element in common contains that element only once, e.g., $\{a, b, c\} \cup \{a, 0, 1\} = \{a, b, c, 0, 1\}$.

The union of a set and one of its subsets is just the bigger set: $\{a, b, c\} \cup \{a\} = \{a, b, c\}$.

The union of a set with the empty set is identical to the set: $\{a, b, c\} \cup \emptyset = \{a, b, c\}$.

Problem 1.4. Prove that if $A \subseteq B$, then $A \cup B = B$.

explanation We can also consider a "dual" operation to union. This is the operation that forms the set of all elements that are elements of A and are also elements

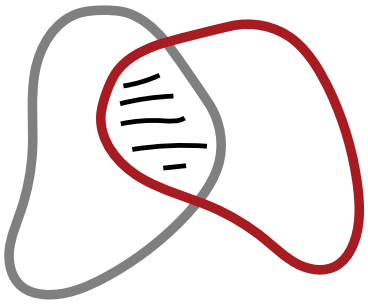


Figure 1.2: The intersection $A \cap B$ of two sets is the set of elements they have in common.

sfr:set:uni:
fig:intersection

of B. This operation is called *intersection*, and can be depicted as in Figure 1.2.

Definition 1.17 (Intersection). The *intersection* of two sets A and B, written $A \cap B$, is the set of all things which are elements of both A and B.

$$A \cap B = \{x : x \in A \wedge x \in B\}$$

Two sets are called *disjoint* if their intersection is empty. This means they have no elements in common.

Example 1.18. If two sets have no elements in common, their intersection is empty: $\{a, b, c\} \cap \{0, 1\} = \emptyset$.

If two sets do have elements in common, their intersection is the set of all those: $\{a, b, c\} \cap \{a, b, d\} = \{a, b\}$.

The intersection of a set with one of its subsets is just the smaller set: $\{a, b, c\} \cap \{a, b\} = \{a, b\}$.

The intersection of any set with the empty set is empty: $\{a, b, c\} \cap \emptyset = \emptyset$.

Problem 1.5. Prove rigorously that if $A \subseteq B$, then $A \cap B = A$.

We can also form the union or intersection of more than two sets. An elegant way of dealing with this in general is the following: suppose you collect all the sets you want to form the union (or intersection) of into a single set. Then we can define the union of all our original sets as the set of all objects which belong to at least one element of the set, and the intersection as the set of all objects which belong to every element of the set.

explanation

Definition 1.19. If A is a set of sets, then $\bigcup A$ is the set of elements of elements of A:

$$\begin{aligned} \bigcup A &= \{x : x \text{ belongs to an element of } A\}, \text{ i.e.,} \\ &= \{x : \text{there is a } B \in A \text{ so that } x \in B\} \end{aligned}$$

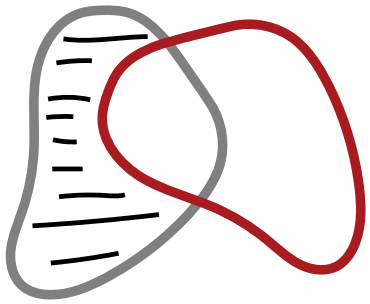


Figure 1.3: The difference $A \setminus B$ of two sets is the set of those elements of A which are not also elements of B.

sfr:set:uni:difference

Definition 1.20. If A is a set of sets, then $\bigcap A$ is the set of objects which all elements of A have in common:

$$\begin{aligned}\bigcap A &= \{x : x \text{ belongs to every element of } A\}, \text{ i.e.,}\\ &= \{x : \text{for all } B \in A, x \in B\}\end{aligned}$$

Example 1.21. Suppose $A = \{\{a, b\}, \{a, d, e\}, \{a, d\}\}$. Then $\bigcup A = \{a, b, d, e\}$ and $\bigcap A = \{a\}$.

Problem 1.6. Show that if A is a set and $A \in B$, then $A \subseteq \bigcup B$.

We could also do the same for a sequence of sets A_1, A_2, ...

$$\begin{aligned}\bigcup_i A_i &= \{x : x \text{ belongs to one of the } A_i\}\\ \bigcap_i A_i &= \{x : x \text{ belongs to every } A_i\}.\end{aligned}$$

When we have an *index* of sets, i.e., some set I such that we are considering A_i for each $i \in I$, we may also use these abbreviations:

$$\begin{aligned}\bigcup_{i \in I} A_i &= \bigcup \{A_i : i \in I\}\\ \bigcap_{i \in I} A_i &= \bigcap \{A_i : i \in I\}\end{aligned}$$

Finally, we may want to think about the set of all elements in A which are not in B. We can depict this as in Figure 1.3.

Definition 1.22 (Difference). The *set difference* $A \setminus B$ is the set of all elements of A which are not also elements of B, i.e.,

$$A \setminus B = \{x : x \in A \text{ and } x \notin B\}.$$

Problem 1.7. Prove that if $A \subsetneq B$, then $B \setminus A \neq \emptyset$.

---

content/sets-functions-relations/sets/pairs-and-products.tex

1.5 Pairs, Tuples, Cartesian Products

sfr:set:pai:sec

It follows from extensionality that sets have no order to their elements. So if we want to represent order, we use *ordered pairs* $\langle x, y\rangle$. In an unordered pair $\{x, y\}$, the order does not matter: $\{x, y\} = \{y, x\}$. In an ordered pair, it does: if $x \neq y$, then $\langle x, y\rangle \neq \langle y, x\rangle$. explanation

How should we think about ordered pairs in set theory? Crucially, we want to preserve the idea that ordered pairs are identical iff they share the same first element and share the same second element, i.e.:

$$\langle a, b\rangle = \langle c, d\rangle \text{ iff both } a = c \text{ and } b = d.$$

We can define ordered pairs in set theory using the Wiener–Kuratowski definition.

sfr:set:pai:wienerkuratowski

Definition 1.23 (Ordered pair). $\langle a, b\rangle = \{\{a\}, \{a, b\}\}$.

Problem 1.8. Using Definition 1.23, prove that $\langle a, b\rangle = \langle c, d\rangle$ iff both $a = c$ and $b = d$.

Having fixed a definition of an ordered pair, we can use it to define further sets. For example, sometimes we also want ordered sequences of more than two objects, e.g., *triples* $\langle x, y, z\rangle$, *quadruples* $\langle x, y, z, u\rangle$, and so on. We can think of triples as special ordered pairs, where the first element is itself an ordered pair: $\langle x, y, z\rangle$ is $\langle\langle x, y\rangle, z\rangle$. The same is true for quadruples: $\langle x, y, z, u\rangle$ is $\langle\langle\langle x, y\rangle, z\rangle, u\rangle$, and so on. In general, we talk of *ordered n-tuples* $\langle x_1, \ldots, x_n\rangle$. explanation

Certain sets of ordered pairs, or other ordered n-tuples, will be useful.

Definition 1.24 (Cartesian product). Given sets A and B, their *Cartesian product* $A \times B$ is defined by

$$A \times B = \{\langle x, y\rangle : x \in A \text{ and } y \in B\}.$$

Example 1.25. If $A = \{0, 1\}$, and $B = \{1, a, b\}$, then their product is

$$A \times B = \{\langle 0, 1\rangle, \langle 0, a\rangle, \langle 0, b\rangle, \langle 1, 1\rangle, \langle 1, a\rangle, \langle 1, b\rangle\}.$$

Example 1.26. If A is a set, the product of A with itself, $A \times A$, is also written A^2. It is the set of *all* pairs $\langle x, y\rangle$ with $x, y \in A$. The set of all triples $\langle x, y, z\rangle$ is A^3, and so on. We can give a recursive definition:

$$A^1 = A$$
$$A^{k+1} = A^k \times A$$

Problem 1.9. List all elements of $\{1, 2, 3\}^3$.

Proposition 1.27. *If A has n elements and B has m elements, then $A \times B$ has $n \cdot m$ elements.* sfr:set:pai:cardnmprod

Proof. For every element x in A, there are m elements of the form $\langle x, y\rangle \in A \times B$. Let $B_x = \{\langle x, y\rangle : y \in B\}$. Since whenever $x_1 \neq x_2$, $\langle x_1, y\rangle \neq \langle x_2, y\rangle$, $B_{x_1} \cap B_{x_2} = \emptyset$. But if $A = \{x_1, \ldots, x_n\}$, then $A \times B = B_{x_1} \cup \cdots \cup B_{x_n}$, and so has $n \cdot m$ elements.

To visualize this, arrange the elements of $A \times B$ in a grid:

$$\begin{array}{rccccc} B_{x_1} = & \{\langle x_1, y_1\rangle & \langle x_1, y_2\rangle & \ldots & \langle x_1, y_m\rangle\} \\ B_{x_2} = & \{\langle x_2, y_1\rangle & \langle x_2, y_2\rangle & \ldots & \langle x_2, y_m\rangle\} \\ \vdots & & \vdots & & \\ B_{x_n} = & \{\langle x_n, y_1\rangle & \langle x_n, y_2\rangle & \ldots & \langle x_n, y_m\rangle\} \end{array}$$

Since the x_i are all different, and the y_j are all different, no two of the pairs in this grid are the same, and there are $n \cdot m$ of them. □

Problem 1.10. Show, by induction on k, that for all $k \geq 1$, if A has n elements, then A^k has n^k elements.

Example 1.28. If A is a set, a *word* over A is any sequence of elements of A. A sequence can be thought of as an n-tuple of elements of A. For instance, if $A = \{a, b, c\}$, then the sequence "bac" can be thought of as the triple $\langle b, a, c\rangle$. Words, i.e., sequences of symbols, are of crucial importance in computer science. By convention, we count elements of A as sequences of length 1, and $\emptyset$ as the sequence of length 0. The set of *all* words over A then is

$$A^* = \{\emptyset\} \cup A \cup A^2 \cup A^3 \cup \ldots$$

---

content/sets-functions-relations/sets/russells-paradox.tex

1.6 Russell's Paradox

sfr:set:rus:sec

Extensionality licenses the notation $\{x : \varphi(x)\}$, for *the* set of x's such that $\varphi(x)$. However, all that extensionality *really* licenses is the following thought. *If* there is a set whose members are all and only the φ's, *then* there is only one such set. Otherwise put: having fixed some φ, the set $\{x : \varphi(x)\}$ is unique, *if it exists*.

But this conditional is important! Crucially, not every property lends itself to *comprehension*. That is, some properties do *not* define sets. If they all did, then we would run into outright contradictions. The most famous example of this is Russell's Paradox.

Sets may be elements of other sets—for instance, the power set of a set A is made up of sets. And so it makes sense to ask or investigate whether a set is

an element of another set. Can a set be a member of itself? Nothing about the idea of a set seems to rule this out. For instance, if *all* sets form a collection of objects, one might think that they can be collected into a single set—the set of all sets. And it, being a set, would be an element of the set of all sets.

Russell's Paradox arises when we consider the property of not having itself as an element, of being *non-self-membered*. What if we suppose that there is a set of all sets that do not have themselves as an element? Does

$$R = \{x : x \notin x\}$$

exist? It turns out that we can prove that it does not.

sfr:set:rus:thm:russells-paradox

Theorem 1.29 (Russell's Paradox). *There is no set $R = \{x : x \notin x\}$.*

Proof. If $R = \{x : x \notin x\}$ exists, then $R \in R$ iff $R \notin R$, which is a contradiction. □

explanation

Let's run through this proof more slowly. If R exists, it makes sense to ask whether $R \in R$ or not. Suppose that indeed $R \in R$. Now, R was defined as the set of all sets that are not elements of themselves. So, if $R \in R$, then R does not itself have R's defining property. But only sets that have this property are in R, hence, R cannot be an element of R, i.e., $R \notin R$. But R can't both be and not be an element of R, so we have a contradiction.

Since the assumption that $R \in R$ leads to a contradiction, we have $R \notin R$. But this also leads to a contradiction! For if $R \notin R$, then R itself does have R's defining property, and so R would be an element of R just like all the other non-self-membered sets. And again, it can't both not be and be an element of R.

digression

How do we set up a set theory which avoids falling into Russell's Paradox, i.e., which avoids making the *inconsistent* claim that $R = \{x : x \notin x\}$ exists? Well, we would need to lay down axioms which give us very precise conditions for stating when sets exist (and when they don't).

The set theory sketched in this chapter doesn't do this. It's *genuinely naïve*. It tells you only that sets obey extensionality and that, if you have some sets, you can form their union, intersection, etc. It is possible to develop set theory

more rigorously than this.

Chapter 2

Relations

content/sets-functions-relations/relations/relations-as-sets.tex

2.1 Relations as Sets

explanation

In section 1.3, we mentioned some important sets: $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$. You will no doubt remember some interesting relations between the elements of some of these sets. For instance, each of these sets has a completely standard *order relation* on it. There is also the relation *is identical with* that every object bears to itself and to no other thing. There are many more interesting relations that we'll encounter, and even more possible relations. Before we review them, though, we will start by pointing out that we can look at relations as a special sort of set.

sfr:rel:set:sec

For this, recall two things from section 1.5. First, recall the notion of an *ordered pair*: given a and b, we can form $\langle a, b\rangle$. Importantly, the order of elements *does* matter here. So if $a \neq b$ then $\langle a, b\rangle \neq \langle b, a\rangle$. (Contrast this with unordered pairs, i.e., 2-element sets, where $\{a, b\} = \{b, a\}$.) Second, recall the notion of a *Cartesian product*: if A and B are sets, then we can form $A \times B$, the set of all pairs $\langle x, y\rangle$ with $x \in A$ and $y \in B$. In particular, $A^2 = A \times A$ is the set of all ordered pairs from A.

Now we will consider a particular relation on a set: the $<$-relation on the set $\mathbb{N}$ of natural numbers. Consider the set of all pairs of numbers $\langle n, m\rangle$ where $n < m$, i.e.,

$$R = \{\langle n, m\rangle : n, m \in \mathbb{N} \text{ and } n < m\}.$$

There is a close connection between n being less than m, and the pair $\langle n, m\rangle$ being a member of R, namely:

$$n < m \text{ iff } \langle n, m\rangle \in R.$$

Indeed, without any loss of information, we can consider the set R to *be* the $<$-relation on $\mathbb{N}$.

In the same way we can construct a subset of $\mathbb{N}^2$ for any relation between numbers. Conversely, given any set of pairs of numbers $S \subseteq \mathbb{N}^2$, there is a corresponding relation between numbers, namely, the relationship n bears to m if and only if $\langle n, m\rangle \in S$. This justifies the following definition:

Definition 2.1 (Binary relation). A *binary relation* on a set A is a subset of A^2. If $R \subseteq A^2$ is a binary relation on A and $x, y \in A$, we sometimes write Rxy (or xRy) for $\langle x, y\rangle \in R$.

sfr:rel:set: relations

Example 2.2. The set $\mathbb{N}^2$ of pairs of natural numbers can be listed in a 2-dimensional matrix like this:

$$\begin{array}{cccccc}
\langle \mathbf{0},\mathbf{0}\rangle & \langle 0,1\rangle & \langle 0,2\rangle & \langle 0,3\rangle & \dots \\
\langle 1,0\rangle & \langle \mathbf{1},\mathbf{1}\rangle & \langle 1,2\rangle & \langle 1,3\rangle & \dots \\
\langle 2,0\rangle & \langle 2,1\rangle & \langle \mathbf{2},\mathbf{2}\rangle & \langle 2,3\rangle & \dots \\
\langle 3,0\rangle & \langle 3,1\rangle & \langle 3,2\rangle & \langle \mathbf{3},\mathbf{3}\rangle & \dots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

We have put the diagonal, here, in bold, since the subset of $\mathbb{N}^2$ consisting of the pairs lying on the diagonal, i.e.,

$$\{\langle 0,0\rangle, \langle 1,1\rangle, \langle 2,2\rangle, \dots\},$$

is the *identity relation on* $\mathbb{N}$. (Since the identity relation is popular, let's define $\mathrm{Id}_A = \{\langle x, x\rangle : x \in A\}$ for any set A.) The subset of all pairs lying above the diagonal, i.e.,

$$L = \{\langle 0,1\rangle, \langle 0,2\rangle, \dots, \langle 1,2\rangle, \langle 1,3\rangle, \dots, \langle 2,3\rangle, \langle 2,4\rangle, \dots\},$$

is the *less than* relation, i.e., Lnm iff $n < m$. The subset of pairs below the diagonal, i.e.,

$$G = \{\langle 1,0\rangle, \langle 2,0\rangle, \langle 2,1\rangle, \langle 3,0\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \dots\},$$

is the *greater than* relation, i.e., Gnm iff $n > m$. The union of L with I, which we might call $K = L \cup I$, is the *less than or equal to* relation: Knm iff $n \leq m$. Similarly, $H = G \cup I$ is the *greater than or equal to relation.* These relations L, G, K, and H are special kinds of relations called *orders.* L and G have the property that no number bears L or G to itself (i.e., for all n, neither Lnn nor Gnn). Relations with this property are called *irreflexive*, and, if they also happen to be orders, they are called *strict orders.*

Although orders and identity are important and natural relations, it should be emphasized that according to our definition *any* subset of A^2 is a relation on A, regardless of how unnatural or contrived it seems. In particular, $\emptyset$ is a relation on any set (the *empty relation*, which no pair of elements bears), and A^2 itself is a relation on A as well (one which every pair bears), called the *universal relation.* But also something like $E = \{\langle n, m\rangle : n > 5 \text{ or } m \times n \geq 34\}$ counts as a relation. explanation

Problem 2.1. List the elements of the relation $\subseteq$ on the set $\wp(\{a, b, c\})$.

---

content/sets-functions-relations/relations/reflections.tex

2.2 Philosophical Reflections

sfr:rel:ref:sec

In section 2.1, we defined relations as certain sets. We should pause and ask a quick philosophical question: what is such a definition *doing*? It is extremely doubtful that we should want to say that we have *discovered* some metaphysical identity facts; that, for example, the order relation on $\mathbb{N}$ *turned out* to be the set $R = \{\langle n, m\rangle : n, m \in \mathbb{N} \text{ and } n < m\}$ that we defined in section 2.1. Here are three reasons why.

First: in Definition 1.23, we defined $\langle a, b\rangle = \{\{a\}, \{a, b\}\}$. Consider instead the definition $\|a, b\| = \{\{b\}, \{a, b\}\} = \langle b, a\rangle$. When $a \neq b$, we have that $\langle a, b\rangle \neq \|a, b\|$. But we could equally have regarded $\|a, b\|$ as our definition of an ordered pair, rather than $\langle a, b\rangle$. Both definitions would have worked equally well. So now we have two equally good candidates to "be" the order relation on the natural numbers, namely:

$$R = \{\langle n, m\rangle : n, m \in \mathbb{N} \text{ and } n < m\}$$
$$S = \{\|n, m\| : n, m \in \mathbb{N} \text{ and } n < m\}.$$

Since $R \neq S$, by extensionality, it is clear that they cannot *both* be identical to the order relation on $\mathbb{N}$. But it would just be arbitrary, and hence a bit embarrassing, to claim that R rather than S (or vice versa) *is* the ordering relation, as a matter of fact. (This is a very simple instance of an argument against set-theoretic reductionism which Benacerraf made famous in 1965. We will revisit it several times.)

Second: if we think that *every* relation should be identified with a set, then the relation of set-membership itself, $\in$, should be a particular set. Indeed, it would have to be the set $\{\langle x, y\rangle : x \in y\}$. But does this set exist? Given Russell's Paradox, it is a non-trivial claim that such a set exists. In fact, it is possible to develop set theory in a rigorous way as an axiomatic theory, and that theory will indeed deny the existence of this set. So, even if some relations can be treated as sets, the relation of set-membership will have to be a special case.

Third: when we "identify" relations with sets, we said that we would allow ourselves to write Rxy for $\langle x, y\rangle \in R$. This is fine, provided that the membership relation, "$\in$", is treated *as* a predicate. But if we think that "$\in$" stands for a certain kind of set, then the expression "$\langle x, y\rangle \in R$" just consists of three singular terms which stand for sets: "$\langle x, y\rangle$", "$\in$", and "R". And such a list of names is no more capable of expressing a proposition than the nonsense string: "the cup penholder the table". Again, even if some relations can be treated as sets, the relation of set-membership must be a special case. (This rolls together

a simple version of Frege's concept *horse* paradox, and a famous objection that Wittgenstein once raised against Russell.)

So where does this leave us? Well, there is nothing *wrong* with our saying that the relations on the numbers are sets. We just have to understand the spirit in which that remark is made. We are not stating a metaphysical identity fact. We are simply noting that, in certain contexts, we can (and will) *treat* (certain) relations as certain sets.

---

content/sets-functions-relations/relations/special-properties.tex

2.3 Special Properties of Relations

sfr:rel:prp:sec

Some kinds of relations turn out to be so common that they have been given special names. For instance, $\leq$ and $\subseteq$ both relate their respective domains (say, $\mathbb{N}$ in the case of $\leq$ and $\wp(A)$ in the case of $\subseteq$) in similar ways. To get at exactly how these relations are similar, and how they differ, we categorize them according to some special properties that relations can have. It turns out that (combinations of) some of these special properties are especially important: orders and equivalence relations. intro

Definition 2.3 (Reflexivity). A relation $R \subseteq A^2$ is *reflexive* iff, for every $x \in A$, Rxx.

Definition 2.4 (Transitivity). A relation $R \subseteq A^2$ is *transitive* iff, whenever Rxy and Ryz, then also Rxz.

Definition 2.5 (Symmetry). A relation $R \subseteq A^2$ is *symmetric* iff, whenever Rxy, then also Ryx.

Definition 2.6 (Anti-symmetry). A relation $R \subseteq A^2$ is *anti-symmetric* iff, whenever both Rxy and Ryx, then $x = y$ (or, in other words: if $x \neq y$ then either $\neg Rxy$ or $\neg Ryx$).

explanation

In a symmetric relation, Rxy and Ryx always hold together, or neither holds. In an anti-symmetric relation, the only way for Rxy and Ryx to hold together is if $x = y$. Note that this does not *require* that Rxy and Ryx holds when $x = y$, only that it isn't ruled out. So an anti-symmetric relation can be reflexive, but it is not the case that every anti-symmetric relation is reflexive. Also note that being anti-symmetric and merely not being symmetric are different conditions. In fact, a relation can be both symmetric and anti-symmetric at the same time (e.g., the identity relation is).

Definition 2.7 (Connectivity). A relation $R \subseteq A^2$ is *connected* if for all $x, y \in A$, if $x \neq y$, then either Rxy or Ryx.

Problem 2.2. Give examples of relations that are (a) reflexive and symmetric but not transitive, (b) reflexive and anti-symmetric, (c) anti-symmetric, transitive, but not reflexive, and (d) reflexive, symmetric, and transitive. Do not use relations on numbers or sets.

Definition 2.8 (Irreflexivity). A relation $R \subseteq A^2$ is called *irreflexive* if, for all $x \in A$, not Rxx.

Definition 2.9 (Asymmetry). A relation $R \subseteq A^2$ is called *asymmetric* if for no pair $x, y \in A$ we have both Rxy and Ryx.

Note that if $A \neq \emptyset$, then no irreflexive relation on A is reflexive and every asymmetric relation on A is also anti-symmetric. However, there are $R \subseteq A^2$ that are not reflexive and also not irreflexive, and there are anti-symmetric relations that are not asymmetric.

---

content/sets-functions-relations/relations/equivalence-relations.tex

2.4 Equivalence Relations

The identity relation on a set is reflexive, symmetric, and transitive. Relations R that have all three of these properties are very common. sfr:rel:eqv:sec

Definition 2.10 (Equivalence relation). A relation $R \subseteq A^2$ that is reflexive, symmetric, and transitive is called an *equivalence relation*. Elements x and y of A are said to be *R-equivalent* if Rxy.

Equivalence relations give rise to the notion of an *equivalence class*. An equivalence relation "chunks up" the domain into different partitions. Within each partition, all the objects are related to one another; and no objects from different partitions relate to one another. Sometimes, it's helpful just to talk about these partitions *directly*. To that end, we introduce a definition:

Definition 2.11. Let $R \subseteq A^2$ be an equivalence relation. For each $x \in A$, the *equivalence class* of x in A is the set $[x]_R = \{y \in A : Rxy\}$. The *quotient* of A under R is $A/_R = \{[x]_R : x \in A\}$, i.e., the set of these equivalence classes. sfr:rel:eqv:def:equivalenceclass

The next result vindicates the definition of an equivalence class, in proving that the equivalence classes are indeed the partitions of A:

Proposition 2.12. *If $R \subseteq A^2$ is an equivalence relation, then Rxy iff $[x]_R = [y]_R$.*

Proof. For the left-to-right direction, suppose Rxy, and let $z \in [x]_R$. By definition, then, Rxz. Since R is an equivalence relation, Ryz. (Spelling this out: as Rxy and R is symmetric we have Ryx, and as Rxz and R is transitive we have Ryz.) So $z \in [y]_R$. Generalising, $[x]_R \subseteq [y]_R$. But exactly similarly, $[y]_R \subseteq [x]_R$. So $[x]_R = [y]_R$, by extensionality.

For the right-to-left direction, suppose $[x]_R = [y]_R$. Since R is reflexive, Ryy, so $y \in [y]_R$. Thus also $y \in [x]_R$ by the assumption that $[x]_R = [y]_R$. So Rxy. □

Example 2.13. A nice example of equivalence relations comes from modular arithmetic. For any a, b, and $n \in \mathbb{Z}^+$, say that $a \equiv_n b$ iff dividing a by n gives the same remainder as dividing b by n. (Somewhat more symbolically: $a \equiv_n b$ iff, for some $k \in \mathbb{Z}$, $a - b = kn$.) Now, $\equiv_n$ is an equivalence relation, for any n. And there are exactly n distinct equivalence classes generated by $\equiv_n$; that is, $\mathbb{N}/_{\equiv_n}$ has n elements. These are: the set of numbers divisible by n without remainder, i.e., $[0]_{\equiv_n}$; the set of numbers divisible by n with remainder 1, i.e., $[1]_{\equiv_n}$; ...; and the set of numbers divisible by n with remainder $n-1$, i.e., $[n-1]_{\equiv_n}$.

Problem 2.3. Show that $\equiv_n$ is an equivalence relation, for any $n \in \mathbb{Z}^+$, and that $\mathbb{N}/_{\equiv_n}$ has exactly n members.

---

content/sets-functions-relations/relations/orders.tex

2.5 Orders

sfr:rel:ord:sec

explanation

Many of our comparisons involve describing some objects as being "less than", "equal to", or "greater than" other objects, in a certain respect. These involve *order* relations. But there are different kinds of order relations. For instance, some require that any two objects be comparable, others don't. Some include identity (like $\leq$) and some exclude it (like $<$). It will help us to have a taxonomy here.

Definition 2.14 (Preorder). A relation which is both reflexive and transitive is called a *preorder*.

Definition 2.15 (Partial order). A preorder which is also anti-symmetric is called a *partial order*.

sfr:rel:ord:def:linearorder

Definition 2.16 (Linear order). A partial order which is also connected is called a *total order* or *linear order*.

Example 2.17. Every linear order is also a partial order, and every partial order is also a preorder, but the converses don't hold. The universal relation on A is a preorder, since it is reflexive and transitive. But, if A has more than one element, the universal relation is not anti-symmetric, and so not a partial order.

Example 2.18. Consider the *no longer than* relation $\preccurlyeq$ on $\mathbb{B}^*$: $x \preccurlyeq y$ iff $\text{len}(x) \leq \text{len}(y)$. This is a preorder (reflexive and transitive), and even connected, but not a partial order, since it is not anti-symmetric. For instance, $01 \preccurlyeq 10$ and $10 \preccurlyeq 01$, but $01 \neq 10$.

Example 2.19. An important partial order is the relation $\subseteq$ on a set of sets. This is not in general a linear order, since if $a \neq b$ and we consider $\wp(\{a, b\}) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$, we see that $\{a\} \not\subseteq \{b\}$ and $\{a\} \neq \{b\}$ and $\{b\} \not\subseteq \{a\}$.

Example 2.20. The relation of *divisibility without remainder* gives us a partial order which isn't a linear order. For integers n and m, we write $n \mid m$ to mean n (evenly) divides m, i.e., iff there is some integer k so that $m = kn$. On $\mathbb{N}$, this is a partial order, but not a linear order: for instance, $2 \nmid 3$ and also $3 \nmid 2$. Considered as a relation on $\mathbb{Z}$, divisibility is only a preorder since it is not anti-symmetric: $1 \mid -1$ and $-1 \mid 1$ but $1 \neq -1$.

Example 2.21. The *extension* relation on a set of sequences A^* is the following: $s \sqsubseteq s'$ iff $s = \Lambda$ (the empty sequence), $s = s'$, or $s = \langle s_1, \ldots, s_n \rangle$ and $s' = \langle s_1, \ldots, s_n, s_{n+1}, \ldots, s_m \rangle$. If $s \sqsubseteq s'$ we also say that s is an *initial segment* of s'. The extension relation on A^* is a partial order but not a linear order, e.g., if $a \neq b$, then $ab \not\sqsubseteq ba$ and $ba \not\sqsubseteq ab$.

Definition 2.22 (Strict order). A *strict order* is a relation which is irreflexive, asymmetric, and transitive.

Definition 2.23 (Strict linear order). A strict order which is also connected is called a *strict total order* or *strict linear order.* sfr:rel:ord: def:strictlinearorder

Example 2.24. $\leq$ is the linear order corresponding to the strict linear order $<$. $\subseteq$ is the partial order corresponding to the strict order $\subsetneq$.

Any strict order R on A can be turned into a partial order by adding the diagonal Id_A, i.e., adding all the pairs $\langle x, x \rangle$. (This is called the *reflexive closure* of R.) Conversely, starting from a partial order, one can get a strict order by removing Id_A. These next two results make this precise.

Proposition 2.25. *If R is a strict order on A, then $R^+ = R \cup \mathrm{Id}_A$ is a partial order. Moreover, if R is a strict linear order, then R^+ is a linear order.* sfr:rel:ord: prop:stricttopartial

Proof. Suppose R is a strict order, i.e., $R \subseteq A^2$ and R is irreflexive, asymmetric, and transitive. Let $R^+ = R \cup \mathrm{Id}_A$. We have to show that R^+ is reflexive, anti-symmetric, and transitive.

R^+ is clearly reflexive, since $\langle x, x \rangle \in \mathrm{Id}_A \subseteq R^+$ for all $x \in A$.

To show R^+ is anti-symmetric, suppose for reductio that R^+xy and R^+yx but $x \neq y$. Since $\langle x, y \rangle \in R \cup \mathrm{Id}_A$, but $\langle x, y \rangle \notin \mathrm{Id}_A$, we must have $\langle x, y \rangle \in R$, i.e., Rxy. Similarly, Ryx. But this contradicts the assumption that R is asymmetric.

To establish transitivity, suppose that R^+xy and R^+yz. If both $\langle x, y \rangle \in R$ and $\langle y, z \rangle \in R$, then $\langle x, z \rangle \in R$ since R is transitive. Otherwise, either $\langle x, y \rangle \in \mathrm{Id}_A$, i.e., $x = y$, or $\langle y, z \rangle \in \mathrm{Id}_A$, i.e., $y = z$. In the first case, we have that R^+yz by assumption, $x = y$, hence R^+xz. Similarly in the second case. In either case, R^+xz, thus, R^+ is also transitive.

Concerning the "moreover" clause, suppose that R is also connected. So for all $x \neq y$, either Rxy or Ryx, i.e., either $\langle x, y\rangle \in R$ or $\langle y, x\rangle \in R$. Since $R \subseteq R^+$, this remains true of R^+, so R^+ is connected as well. □

sfr:rel:ord: prop:partialtostrict

Proposition 2.26. *If R is a partial order on A, then $R^- = R \setminus \mathrm{Id}_A$ is a strict order. Moreover, if R is a linear order, then R^- is a strict linear order.*

Proof. This is left as an exercise. □

Problem 2.4. Give a proof of Proposition 2.26.

The following simple result establishes that strict linear orders satisfy an extensionality-like property:

sfr:rel:ord: p:extensionality-strictlinearorders

Proposition 2.27. *If $<$ is a strict linear order on A, then:*

$$(\forall a, b \in A)((\forall x \in A)(x < a \leftrightarrow x < b) \rightarrow a = b).$$

Proof. Suppose $(\forall x \in A)(x < a \leftrightarrow x < b)$. If $a < b$, then $a < a$, contradicting the fact that $<$ is irreflexive; so $a \not< b$. Exactly similarly, $b \not< a$. So $a = b$, as $<$ is connected. □

---

content/sets-functions-relations/relations/graphs.tex

2.6 Graphs

sfr:rel:grp: sec

A *graph* is a diagram in which points—called "nodes" or "vertices" (plural of "vertex")—are connected by edges. Graphs are a ubiquitous tool in discrete mathematics and in computer science. They are incredibly useful for representing, and visualizing, relationships and structures, from concrete things like networks of various kinds to abstract structures such as the possible outcomes of decisions. There are many different kinds of graphs in the literature which differ, e.g., according to whether the edges are directed or not, have labels or not, whether there can be edges from a node to the same node, multiple edges between the same nodes, etc. *Directed graphs* have a special connection to relations.

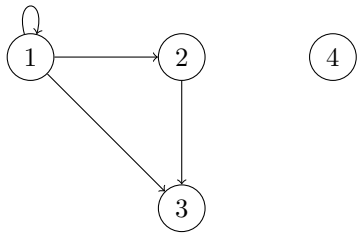
Definition 2.28 (Directed graph). A *directed graph* $G = \langle V, E\rangle$ is a set of *vertices* V and a set of *edges* $E \subseteq V^2$.

According to our definition, a graph just is a set together with a relation on that set. Of course, when talking about graphs, it's only natural to expect that they are graphically represented: we can draw a graph by connecting two vertices v_1 and v_2 by an arrow iff $\langle v_1, v_2\rangle \in E$. The only difference between a relation by itself and a graph is that a graph specifies the set of vertices, i.e., a graph may have isolated vertices. The important point, however, is that every

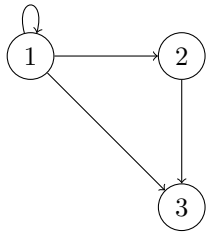
explanation

relation R on a set X can be seen as a directed graph $\langle X, R\rangle$, and conversely, a directed graph $\langle V, E\rangle$ can be seen as a relation $E \subseteq V^2$ with the set V explicitly specified.

Example 2.29. The graph $\langle V, E\rangle$ with $V = \{1, 2, 3, 4\}$ and $E = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$ looks like this:



This is a different graph than $\langle V', E\rangle$ with $V' = \{1, 2, 3\}$, which looks like this:



Problem 2.5. Consider the less-than-or-equal-to relation $\leq$ on the set $\{1, 2, 3, 4\}$ as a graph and draw the corresponding diagram.

---

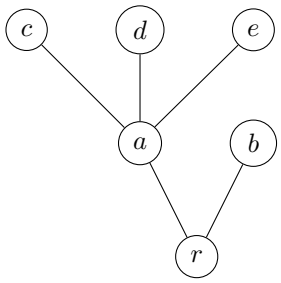
content/sets-functions-relations/relations/trees.tex

2.7 Trees

sfr:rel:tre:sec

A particular kind of partial order which plays an important role in all parts of logic is a *tree*. Finite trees occur in elementary parts of logic: for example, formulas can be understood in terms of their decomposition into a syntax tree, while derivations in many derivation systems also take the form of finite trees. Infinite trees appear already in the proof of the completeness theorems for propositional and first-order logic, and are used throughout mathematical logic.

The set-theoretic concept of a tree is closely related to the notion of a tree in graph theory. Here is a picture of a (finite) tree:



The lowermost node r is the root. Every node other than r has exactly one parent node immediately below it. We can think of the relation a node x stands in to a node y if y can be reached from x by following edges upwards as x being an *ancestor* of y.

The ancestor relation in a tree is a strict partial order. This motivates the set-theoretic definition. To state it we need two concepts. A *least element* in a set A partially ordered by $\leq$ is an element $x \in A$ such that for all $y \in A$ we have that $x \leq y$. A set is *well-ordered* by $\leq$ if every one of its non-empty subsets has a least element.

Definition 2.30 (Tree). A *tree* is a pair $T = \langle A, \leq \rangle$ such that A is a set and $\leq$ is a partial order on A with a unique least element $r \in A$ (called the *root*) such that for all $x \in A$, the set $\{y : y \leq x\}$ is well-ordered by $\leq$.

Definition 2.31 (Successors). Suppose $T = \langle A, \leq \rangle$ is a tree. If $x, y \in A$, $x < y$, and there is no $z \in A$ such that $x < z < y$, then we say that y is a *successor* of x.

The successors of $x \in A$ are also called its *children*. If y is a successor of x, then we call x the *predecessor* or *parent* of y.

Proposition 2.32. *If $\langle A, \leq \rangle$ is a tree, then every $x \in A$ other than the root has at most one predecessor.*

Proof. Suppose $y_1 < x$ and $y_2 < x$ and $y_1 \neq y_2$. Then $\{y_1, y_2\} \subseteq \{z : z < x\}$. Since $\{z : z < x\}$ is well-ordered by $\leq$, its subset $\{y_1, y_2\}$ has a least element, which obviously must be either y_1 or y_2. So either $y_1 \leq y_2$ or $y_2 \leq y_1$. We assumed that $y_1 \neq y_2$, so actually either $y_1 < y_2$ or $y_2 < y_1$. Since we assumed that $y_1 < x$ and $y_2 < x$, we furthermore have that either $y_1 < y_2 < x$ or $y_2 < y_1 < x$. So y_1 and y_2 cannot both be predecessors of x. □

Definition 2.33. A tree $T = \langle A, \leq \rangle$ is said to be *infinite* if A is an infinite set, and *finite* otherwise. If T is such that every $x \in A$ has only finitely many successors, then we say that T is *finitely branching*.

Definition 2.34 (Branches). Given a tree $T = \langle A, \leq \rangle$, a *branch* of T is a maximal chain in T, i.e., a set $B \subseteq A$ such that for any $x, y \in B$ either $x \leq y$ or $y \leq x$, and for any $z \in X \setminus B$ there exists $u \in B$ such that neither $z \leq u$ nor $u \leq z$. We use $[T]$ to denote the set of all branches of T.

Example 2.35. A classic example of a finitely branching tree is the *infinite binary tree* of finite sequences of 0s and 1s, sometimes denoted $\{0,1\}^*$ or $\mathbb{B}^*$, ordered by the extension relation $\sqsubseteq$ (e.g., $101 \sqsubseteq 101101$). Since any binary string can always be extended by adding a 0 or a 1 on the end, this tree contains infinitely many elements: every element s has exactly two successors, $s0$ and $s1$. Its root is the empty sequence Λ.

Example 2.36. Slightly more generally, the set of finite sequences of natural numbers $\mathbb{N}^*$ with the extension relation $\sqsubseteq$ is also a tree. It is obviously not finitely branching: every $s \in \mathbb{N}^*$ has infinitely many successors sn, one for every $n \in \mathbb{N}$. Every $A \subseteq \mathbb{N}^*$ which is closed under $\sqsubseteq$ is a *subtree* of $\mathbb{N}^*$. (That is, A is such that if $s \in A$ and $s' \sqsubseteq s$, then also $s' \in A$.) All finite trees can be represented as finite subtrees of $\mathbb{N}^*$.

Proposition 2.37 (Kőnig's lemma). *If $T = \langle A, \leq \rangle$ is a finitely branching infinite tree, then T has an infinite branch.*

A special case of Kőnig's lemma widely used in computability theory, known as *weak Kőnig's lemma*, is the following: any infinite subtree of $\{0,1\}^*$ has an infinite branch.

---

content/sets-functions-relations/relations/operations.tex

2.8 Operations on Relations

sfr:rel:ops:sec

It is often useful to modify or combine relations. In Proposition 2.25, we considered the *union* of relations, which is just the union of two relations considered as sets of pairs. Similarly, in Proposition 2.26, we considered the relative difference of relations. Here are some other operations we can perform on relations.

sfr:rel:ops:relationoperations

Definition 2.38. Let R, S be relations, and A be any set.

The *inverse* of R is $R^{-1} = \{\langle y, x\rangle : \langle x, y\rangle \in R\}$.

The *relative product* of R and S is $(R \mid S) = \{\langle x, z\rangle : \exists y(Rxy \wedge Syz)\}$.

The *restriction* of R to A is $R\restriction_A = R \cap A^2$.

The *application* of R to A is $R[A] = \{y : (\exists x \in A)Rxy\}$

Example 2.39. Let $S \subseteq \mathbb{Z}^2$ be the successor relation on $\mathbb{Z}$, i.e., $S = \{\langle x, y\rangle \in \mathbb{Z}^2 : x + 1 = y\}$, so that Sxy iff $x + 1 = y$.

S^{-1} is the predecessor relation on $\mathbb{Z}$, i.e., $\{\langle x, y\rangle \in \mathbb{Z}^2 : x - 1 = y\}$.

$S \mid S$ is $\{\langle x, y\rangle \in \mathbb{Z}^2 : x + 2 = y\}$

$S\restriction_{\mathbb{N}}$ is the successor relation on $\mathbb{N}$.

$S[\{1, 2, 3\}]$ is $\{2, 3, 4\}$.

Definition 2.40 (Transitive closure). Let $R \subseteq A^2$ be a binary relation.

The *transitive closure* of R is $R^+ = \bigcup_{0<n\in\mathbb{N}} R^n$, where we recursively define $R^1 = R$ and $R^{n+1} = R^n \mid R$.

The *reflexive transitive closure* of R is $R^* = R^+ \cup \mathrm{Id}_A$.

Example 2.41. Take the successor relation $S \subseteq \mathbb{Z}^2$. S^2xy iff $x+2=y$, S^3xy iff $x+3=y$, etc. So S^+xy iff $x+n=y$ for some $n \geq 1$. In other words, S^+xy iff $x < y$, and S^*xy iff $x \leq y$.

Problem 2.6. Show that the transitive closure of R is in fact transitive.

Chapter 3

Functions

---

content/sets-functions-relations/functions/function-basics.tex

3.1 Basics

sfr:fun:bas:sec

A *function* is a map which sends each element of a given set to a specific element in some (other) given set. For instance, the operation of adding 1 defines a function: each number n is mapped to a unique number $n+1$. explanation

More generally, functions may take pairs, triples, etc., as inputs and return some kind of output. Many functions are familiar to us from basic arithmetic. For instance, addition and multiplication are functions. They take in two numbers and return a third.

In this mathematical, abstract sense, a function is a *black box*: what matters is only what output is paired with what input, not the method for calculating the output.

Definition 3.1 (Function). A *function* $f\colon A \to B$ is a mapping of each element of A to an element of B.

We call A the *domain* of f and B the *codomain* of f. The elements of A are called inputs or *arguments* of f, and the element of B that is paired with an argument x by f is called the *value of* f for argument x, written $f(x)$.

The *range* $\mathrm{ran}(f)$ of f is the subset of the codomain consisting of the values of f for some argument; $\mathrm{ran}(f) = \{f(x) : x \in A\}$.

The diagram in Figure 3.1 may help to think about functions. The ellipse on the left represents the function's *domain*; the ellipse on the right represents the function's *codomain*; and an arrow points from an *argument* in the domain to the corresponding *value* in the codomain.

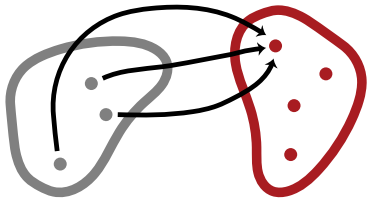


Figure 3.1: A function is a mapping of each element of one set to an element of another. An arrow points from an argument in the domain to the corresponding value in the codomain.

sfr:fun:bas: fig:function

Example 3.2. Multiplication takes pairs of natural numbers as inputs and maps them to natural numbers as outputs, so goes from $\mathbb{N} \times \mathbb{N}$ (the domain) to $\mathbb{N}$ (the codomain). As it turns out, the range is also $\mathbb{N}$, since every $n \in \mathbb{N}$ is $n \times 1$.

Example 3.3. Multiplication is a function because it pairs each input—each pair of natural numbers—with a single output: $\times : \mathbb{N}^2 \to \mathbb{N}$. By contrast, mapping a natural number n to a real x such that $x^2 = n$ is not functional, since each positive integer n has two square roots: $\sqrt{n}$ and $-\sqrt{n}$. We can make it functional by only returning the positive square root: $\sqrt{\ } : \mathbb{N} \to \mathbb{R}$.

Example 3.4. The relation that pairs each student in a class with their final grade is a function—no student can get two different final grades in the same class. The relation that pairs each student in a class with their parents is not a function: students can have zero, or two, or more parents.

explanation

We can define functions by specifying in some precise way what the value of the function is for every possible argument. Different ways of doing this are by giving a formula, describing a method for computing the value, or listing the values for each argument. However functions are defined, we must make sure that for each argument we specify one, and only one, value.

Example 3.5. Let $f : \mathbb{N} \to \mathbb{N}$ be defined such that $f(x) = x + 1$. This is a definition that specifies f as a function which takes in natural numbers and outputs natural numbers. It tells us that, given a natural number x, f will output its successor $x + 1$. In this case, the codomain $\mathbb{N}$ is not the range of f, since the natural number 0 is not the successor of any natural number. The range of f is the set of all positive integers, $\mathbb{Z}^+$.

Example 3.6. Let $g : \mathbb{N} \to \mathbb{N}$ be defined such that $g(x) = x + 2 - 1$. This tells us that g is a function which takes in natural numbers and outputs natural numbers. Given a natural number n, g will output the predecessor of the successor of the successor of x, i.e., $x + 1$.

sfr:fun:bas: examplefunext

explanation

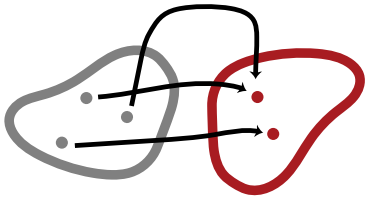


Figure 3.2: A surjective function has every element of the codomain as a value.

sfr:fun:kin:fig:surjective

We just considered two functions, f and g, with different *definitions*. However, these are the *same function*. After all, for any natural number n, we have that $f(n) = n + 1 = n + 2 - 1 = g(n)$. Otherwise put: our definitions for f and g specify the same mapping by means of different equations. Implicitly, then, we are relying upon a principle of extensionality for functions,

$$\text{if } \forall x\, f(x) = g(x), \text{ then } f = g$$

provided that f and g share the same domain and codomain.

Example 3.7. We can also define functions by cases. For instance, we could define $h\colon \mathbb{N} \to \mathbb{N}$ by

$$h(x) = \begin{cases} \frac{x}{2} & \text{if } x \text{ is even} \\ \frac{x+1}{2} & \text{if } x \text{ is odd.} \end{cases}$$

Since every natural number is either even or odd, the output of this function will always be a natural number. Just remember that if you define a function by cases, every possible input must fall into exactly one case. In some cases, this will require a proof that the cases are exhaustive and exclusive.

---

content/sets-functions-relations/functions/function-kinds.tex

3.2 Kinds of Functions

sfr:fun:kin:sec

explanation

It will be useful to introduce a kind of taxonomy for some of the kinds of functions which we encounter most frequently.

To start, we might want to consider functions which have the property that every member of the codomain is a value of the function. Such functions are called surjective, and can be pictured as in Figure 3.2.

Definition 3.8 (Surjective function). A function $f\colon A \to B$ is *surjective* iff B is also the range of f, i.e., for every $y \in B$ there is at least one $x \in A$ such that $f(x) = y$, or in symbols:

$$(\forall y \in B)(\exists x \in A) f(x) = y.$$

We call such a function a surjection from A to B.

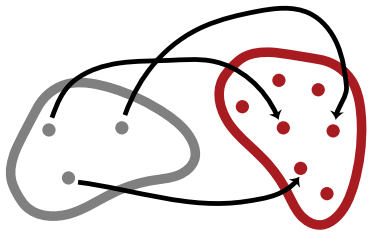


Figure 3.3: An injective function never maps two different arguments to the same value.

sfr:fun:kin:
fig:injective

explanation If you want to show that f is a surjection, then you need to show that every object in f's codomain is the value of $f(x)$ for some input x.

Note that any function *induces* a surjection. After all, given a function $f\colon A \to B$, let $f'\colon A \to \text{ran}(f)$ be defined by $f'(x) = f(x)$. Since $\text{ran}(f)$ is *defined* as $\{f(x) \in B : x \in A\}$, this function f' is guaranteed to be a surjection

explanation Now, any function maps each possible input to a unique output. But there are also functions which never map different inputs to the same outputs. Such functions are called injective, and can be pictured as in Figure 3.3.

Definition 3.9 (Injective function). A function $f\colon A \to B$ is *injective* iff for each $y \in B$ there is at most one $x \in A$ such that $f(x) = y$. We call such a function an injection from A to B.

explanation If you want to show that f is an injection, you need to show that for any elements x and y of f's domain, if $f(x) = f(y)$, then $x = y$.

Example 3.10. The constant function $f\colon \mathbb{N} \to \mathbb{N}$ given by $f(x) = 1$ is neither injective, nor surjective.

The identity function $f\colon \mathbb{N} \to \mathbb{N}$ given by $f(x) = x$ is both injective and surjective.

The successor function $f\colon \mathbb{N} \to \mathbb{N}$ given by $f(x) = x + 1$ is injective but not surjective.

The function $f\colon \mathbb{N} \to \mathbb{N}$ defined by:

$$f(x) = \begin{cases} \frac{x}{2} & \text{if } x \text{ is even} \\ \frac{x+1}{2} & \text{if } x \text{ is odd.} \end{cases}$$

is surjective, but not injective.

explanation Often enough, we want to consider functions which are both injective and surjective. We call such functions bijective. They look like the function pictured in Figure 3.4. Bijections are also sometimes called *one-to-one correspondences*, since they uniquely pair elements of the codomain with elements of the domain.

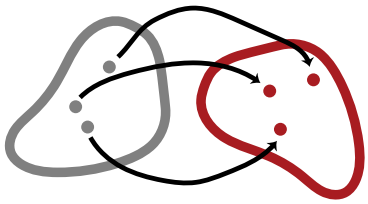


Figure 3.4: A bijective function uniquely pairs the elements of the codomain with those of the domain.

sfr:fun:kin:fig:bijective

Definition 3.11 (Bijection). A function $f\colon A \to B$ is *bijective* iff it is both surjective and injective. We call such a function a bijection from A to B (or between A and B).

---

content/sets-functions-relations/functions/functions-relations.tex

3.3 Functions as Relations

sfr:fun:rel:sec

A function which maps elements of A to elements of B obviously defines a relation between A and B, namely the relation which holds between x and y iff $f(x) = y$. In fact, we might even—if we are interested in reducing the building blocks of mathematics for instance—*identify* the function f with this relation, i.e., with a set of pairs. This then raises the question: which relations define functions in this way? explanation

Definition 3.12 (Graph of a function). Let $f\colon A \to B$ be a function. The *graph* of f is the relation $R_f \subseteq A \times B$ defined by

$$R_f = \{\langle x, y\rangle : f(x) = y\}.$$

The graph of a function is uniquely determined, by extensionality. Moreover, extensionality (on sets) will immediately vindicate the implicit principle of extensionality for functions, whereby if f and g share a domain and codomain then they are identical if they agree on all values. explanation

Similarly, if a relation is "functional", then it is the graph of a function.

sfr:fun:rel:prop:graph-function

Proposition 3.13. *Let $R \subseteq A \times B$ be such that:*

1. *If Rxy and Rxz then $y = z$; and*
2. *for every $x \in A$ there is some $y \in B$ such that $\langle x, y\rangle \in R$.*

Then R is the graph of the function $f\colon A \to B$ defined by $f(x) = y$ iff Rxy.

Proof. Suppose there is a y such that Rxy. If there were another $z \neq y$ such that Rxz, the condition on R would be violated. Hence, if there is a y such that Rxy, this y is unique, and so f is well-defined. Obviously, $R_f = R$. □

explanation Every function $f\colon A \to B$ has a graph, i.e., a relation on $A \times B$ defined by $f(x) = y$. On the other hand, every relation $R \subseteq A \times B$ with the properties given in Proposition 3.13 is the graph of a function $f\colon A \to B$. Because of this close connection between functions and their graphs, we can think of a function simply as its graph. In other words, functions can be identified with certain relations, i.e., with certain sets of tuples. Note, though, that the spirit of this "identification" is as in section 2.2: it is not a claim about the metaphysics of functions, but an observation that it is convenient to *treat* functions as certain sets. One reason that this is so convenient, is that we can now consider performing similar operations on functions as we performed on relations (see section 2.8). In particular:

sfr:fun:rel: defn:funimage **Definition 3.14.** Let $f\colon A \to B$ be a function with $C \subseteq A$.

The *restriction* of f to C is the function $f\restriction_C\colon C \to B$ defined by $(f\restriction_C)(x) = f(x)$ for all $x \in C$. In other words, $f\restriction_C = \{\langle x, y\rangle \in R_f : x \in C\}$.

The *application* of f to C is $f[C] = \{f(x) : x \in C\}$. We also call this the *image* of C under f.

explanation It follows from these definitions that $\mathrm{ran}(f) = f[\mathrm{dom}(f)]$, for any function f. These notions are exactly as one would expect, given the definitions in section 2.8 and our identification of functions with relations. But two other operations—inverses and relative products—require a little more detail. We will provide that in section 3.4 and section 3.5.

---

content/sets-functions-relations/functions/inverses.tex

3.4 Inverses of Functions

explanation We think of functions as maps. An obvious question to ask about functions, then, is whether the mapping can be "reversed." For instance, the successor function $f(x) = x+1$ can be reversed, in the sense that the function $g(y) = y-1$ "undoes" what f does. sfr:fun:inv: sec

But we must be careful. Although the definition of g defines a function $\mathbb{Z} \to \mathbb{Z}$, it does not define a *function* $\mathbb{N} \to \mathbb{N}$, since $g(0) \notin \mathbb{N}$. So even in simple cases, it is not quite obvious whether a function can be reversed; it may depend on the domain and codomain.

This is made more precise by the notion of an inverse of a function.

Definition 3.15. A function $g\colon B \to A$ is an *inverse* of a function $f\colon A \to B$ if $f(g(y)) = y$ and $g(f(x)) = x$ for all $x \in A$ and $y \in B$.

If f has an inverse g, we often write f^{-1} instead of g.

explanation Now we will determine when functions have inverses. A good candidate for an inverse of $f\colon A \to B$ is $g\colon B \to A$ "defined by"

$$g(y) = \text{"the" } x \text{ such that } f(x) = y.$$

But the scare quotes around "defined by" (and "the") suggest that this is not a definition. At least, it will not always work, with complete generality. For, in order for this definition to specify a function, there has to be one and only one x such that $f(x) = y$—the output of g has to be uniquely specified. Moreover, it has to be specified for every $y \in B$. If there are x_1 and $x_2 \in A$ with $x_1 \neq x_2$ but $f(x_1) = f(x_2)$, then $g(y)$ would not be uniquely specified for $y = f(x_1) = f(x_2)$. And if there is no x at all such that $f(x) = y$, then $g(y)$ is not specified at all. In other words, for g to be defined, f must be both injective and surjective.

Let's go slowly. We'll divide the question into two: Given a function $f\colon A \to B$, when is there a function $g\colon B \to A$ so that $g(f(x)) = x$? Such a g "undoes" what f does, and is called a *left inverse* of f. Secondly, when is there a function $h\colon B \to A$ so that $f(h(y)) = y$? Such an h is called a *right inverse* of f—f "undoes" what h does.

Proposition 3.16. *If $f\colon A \to B$ is injective, then there is a* left inverse *$g\colon B \to A$ of f so that $g(f(x)) = x$ for all $x \in A$.*

Proof. Suppose that $f\colon A \to B$ is injective. Consider a $y \in B$. If $y \in \operatorname{ran}(f)$, there is an $x \in A$ so that $f(x) = y$. Because f is injective, there is only one such $x \in A$. Then we can define: $g(y) = x$, i.e., $g(y)$ is "the" $x \in A$ such that $f(x) = y$. If $y \notin \operatorname{ran}(f)$, we can map it to any $a \in A$. So, we can pick an $a \in A$ and define $g\colon B \to A$ by:

$$g(y) = \begin{cases} x & \text{if } f(x) = y \\ a & \text{if } y \notin \operatorname{ran}(f). \end{cases}$$

It is defined for all $y \in B$, since for each such $y \in \operatorname{ran}(f)$ there is exactly one $x \in A$ such that $f(x) = y$. By definition, if $y = f(x)$, then $g(y) = x$, i.e., $g(f(x)) = x$. □

Problem 3.1. Show that if $f\colon A \to B$ has a left inverse g, then f is injective.

Proposition 3.17. *If $f\colon A \to B$ is surjective, then there is a* right inverse *$h\colon B \to A$ of f so that $f(h(y)) = y$ for all $y \in B$.*

Proof. Suppose that $f\colon A \to B$ is surjective. Consider a $y \in B$. Since f is surjective, there is an $x_y \in A$ with $f(x_y) = y$. Then we can define: $h(y) = x_y$, i.e., for each $y \in B$ we choose some $x \in A$ so that $f(x) = y$; since f is surjective there is always at least one to choose from.[1] By definition, if $x = h(y)$, then $f(x) = y$, i.e., for any $y \in B$, $f(h(y)) = y$. □

[1] Since f is surjective, for every $y \in B$ the set $\{x : f(x) = y\}$ is nonempty. Our definition of h requires that we choose a single x from each of these sets. That this is always possible is actually not obvious—the possibility of making these choices is simply assumed as an axiom. In other words, this proposition assumes the so-called Axiom of Choice, an issue we will revisit in chapter 72. However, in many specific cases, e.g., when $A = \mathbb{N}$ or is finite, or when f is bijective, the Axiom of Choice is not required. (In the particular case when f is bijective, for each $y \in B$ the set $\{x : f(x) = y\}$ has exactly one element, so that there is no choice to make.)

Problem 3.2. Show that if $f\colon A \to B$ has a right inverse h, then f is surjective.

explanation

By combining the ideas in the previous proof, we now get that every bijection has an inverse, i.e., there is a single function which is both a left and right inverse of f.

Proposition 3.18. *If $f\colon A \to B$ is bijective, there is a function $f^{-1}\colon B \to A$ so that for all $x \in A$, $f^{-1}(f(x)) = x$ and for all $y \in B$, $f(f^{-1}(y)) = y$.*

sfr:fun:inv: prop:bijection-inverse

Proof. Exercise. □

Problem 3.3. Prove Proposition 3.18. You have to define f^{-1}, show that it is a function, and show that it is an inverse of f, i.e., $f^{-1}(f(x)) = x$ and $f(f^{-1}(y)) = y$ for all $x \in A$ and $y \in B$.

explanation

There is a slightly more general way to extract inverses. We saw in section 3.2 that every function f induces a surjection $f'\colon A \to \operatorname{ran}(f)$ by letting $f'(x) = f(x)$ for all $x \in A$. Clearly, if f is injective, then f' is bijective, so that it has a unique inverse by Proposition 3.18. By a very minor abuse of notation, we sometimes call the inverse of f' simply "the inverse of f."

Proposition 3.19. *Show that if $f\colon A \to B$ has a left inverse g and a right inverse h, then $h = g$.*

sfr:fun:inv: prop:left-right

Proof. Exercise. □

Problem 3.4. Prove Proposition 3.19.

Proposition 3.20. *Every function f has at most one inverse.*

sfr:fun:inv: prop:inverse-unique

Proof. Suppose g and h are both inverses of f. Then in particular g is a left inverse of f and h is a right inverse. By Proposition 3.19, $g = h$. □

---

content/sets-functions-relations/functions/composition.tex

3.5 Composition of Functions

explanation

We saw in section 3.4 that the inverse f^{-1} of a bijection f is itself a function. Another operation on functions is composition: we can define a new function by composing two functions, f and g, i.e., by first applying f and then g. Of course, this is only possible if the ranges and domains match, i.e., the range of f must be a subset of the domain of g. This operation on functions is the analogue of the operation of relative product on relations from section 2.8.

sfr:fun:cmp: sec

A diagram might help to explain the idea of composition. In Figure 3.5, we depict two functions $f\colon A \to B$ and $g\colon B \to C$ and their composition $(g \circ f)$. The function $(g \circ f)\colon A \to C$ pairs each element of A with an element of C.

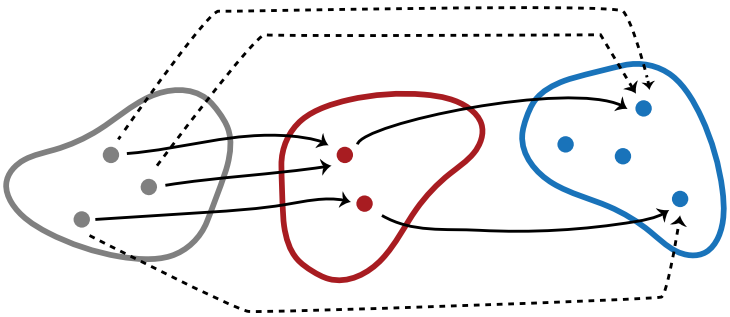


Figure 3.5: The composition $g \circ f$ of two functions f and g.

sfr:fun:cmp:fig:composition

We specify which element of C an element of A is paired with as follows: given an input $x \in A$, first apply the function f to x, which will output some $f(x) = y \in B$, then apply the function g to y, which will output some $g(f(x)) = g(y) = z \in C$.

Definition 3.21 (Composition). Let $f\colon A \to B$ and $g\colon B \to C$ be functions. The *composition* of f with g is $g \circ f\colon A \to C$, where $(g \circ f)(x) = g(f(x))$.

Example 3.22. Consider the functions $f(x) = x + 1$, and $g(x) = 2x$. Since $(g \circ f)(x) = g(f(x))$, for each input x you must first take its successor, then multiply the result by two. So their composition is given by $(g \circ f)(x) = 2(x+1)$.

Problem 3.5. Show that if $f\colon A \to B$ and $g\colon B \to C$ are both injective, then $g \circ f\colon A \to C$ is injective.

Problem 3.6. Show that if $f\colon A \to B$ and $g\colon B \to C$ are both surjective, then $g \circ f\colon A \to C$ is surjective.

Problem 3.7. Suppose $f\colon A \to B$ and $g\colon B \to C$. Show that the graph of $g \circ f$ is $R_f \mid R_g$.

---

content/sets-functions-relations/functions/partial-functions.tex

3.6 Partial Functions

sfr:fun:par:sec

It is sometimes useful to relax the definition of function so that it is not required that the output of the function is defined for all possible inputs. Such mappings are called *partial functions*. explanation

Definition 3.23. A *partial function* $f\colon A \nrightarrow B$ is a mapping which assigns to every element of A at most one element of B. If f assigns an element of B to $x \in A$, we say $f(x)$ is *defined*, and otherwise *undefined*. If $f(x)$ is defined, we write $f(x)\downarrow$, otherwise $f(x)\uparrow$. The *domain* of a partial function f is the subset of A where it is defined, i.e., $\mathrm{dom}(f) = \{x \in A : f(x)\downarrow\}$.

Example 3.24. Every function $f\colon A \to B$ is also a partial function. Partial functions that are defined everywhere on A—i.e., what we so far have simply called a function—are also called *total* functions.

Example 3.25. The partial function $f\colon \mathbb{R} \nrightarrow \mathbb{R}$ given by $f(x) = 1/x$ is undefined for $x = 0$, and defined everywhere else.

Problem 3.8. Given $f\colon A \nrightarrow B$, define the partial function $g\colon B \nrightarrow A$ by: for any $y \in B$, if there is a unique $x \in A$ such that $f(x) = y$, then $g(y) = x$; otherwise $g(y) \uparrow$. Show that if f is injective, then $g(f(x)) = x$ for all $x \in \mathrm{dom}(f)$, and $f(g(y)) = y$ for all $y \in \mathrm{ran}(f)$.

Definition 3.26 (Graph of a partial function). Let $f\colon A \nrightarrow B$ be a partial function. The *graph* of f is the relation $R_f \subseteq A \times B$ defined by

$$R_f = \{\langle x, y\rangle : f(x) = y\}.$$

Proposition 3.27. *Suppose $R \subseteq A \times B$ has the property that whenever Rxy and Rxy' then $y = y'$. Then R is the graph of the partial function $f\colon A \nrightarrow B$ defined by: if there is a y such that Rxy, then $f(x) = y$, otherwise $f(x) \uparrow$. If R is also* serial*, i.e., for each $x \in A$ there is a $y \in B$ such that Rxy, then f is total.*

Proof. Suppose there is a y such that Rxy. If there were another $y' \neq y$ such that Rxy', the condition on R would be violated. Hence, if there is a y such that Rxy, that y is unique, and so f is well-defined. Obviously, $R_f = R$ and f is total if R is serial. □

Chapter 4

The Size of Sets

This chapter discusses enumerations, countability and uncountability. Several sections come in two versions: a more elementary one, that takes enumerations to be lists, or surjections from $\mathbb{Z}^+$; and a more abstract one that defines enumerations as bijections with $\mathbb{N}$.

content/sets-functions-relations/size-of-sets/introduction.tex

4.1 Introduction

sfr:siz:int:sec

When Georg Cantor developed set theory in the 1870s, one of his aims was to make palatable the idea of an infinite collection—an actual infinity, as the medievals would say. A key part of this was his treatment of the *size* of different sets. If a, b and c are all distinct, then the set $\{a, b, c\}$ is intuitively *larger* than $\{a, b\}$. But what about infinite sets? Are they all as large as each other? It turns out that they are not.

The first important idea here is that of an enumeration. We can list every finite set by listing all its elements. For some infinite sets, we can also list all their elements if we allow the list itself to be infinite. Such sets are called enumerable. Cantor's surprising result, which we will fully understand by the end of this chapter, was that some infinite sets are not enumerable.

content/sets-functions-relations/size-of-sets/enumerability.tex

4.2 Enumerations and Enumerable Sets

sfr:siz:enm:sec

This section discusses enumerations of sets, defining them as surjections from $\mathbb{Z}^+$. It does things slowly, for readers with little mathematical background. An alternative, terser version is given in section 4.11, which defines enumerations differently: as bijections with $\mathbb{N}$ (or an initial segment).

We've already given examples of sets by listing their elements. Let's discuss in more general terms how and when we can list the elements of a set, even if that set is infinite. explanation

Definition 4.1 (Enumeration, informally). Informally, an *enumeration* of a set A is a list (possibly infinite) of elements of A such that every element of A appears on the list at some finite position. If A has an enumeration, then A is said to be *enumerable*.

A couple of points about enumerations: explanation

1. We count as enumerations only lists which have a beginning and in which every element other than the first has a single element immediately preceding it. In other words, there are only finitely many elements between the first element of the list and any other element. In particular, this means that every element of an enumeration has a finite position: the first element has position 1, the second position 2, etc.

2. We can have different enumerations of the same set A which differ by the order in which the elements appear: 4, 1, 25, 16, 9 enumerates the (set of the) first five square numbers just as well as 1, 4, 9, 16, 25 does.

3. Redundant enumerations are still enumerations: 1, 1, 2, 2, 3, 3, ... enumerates the same set as 1, 2, 3, ... does.

4. Order and redundancy *do* matter when we specify an enumeration: we can enumerate the positive integers beginning with 1, 2, 3, 1, ..., but the pattern is easier to see when enumerated in the standard way as 1, 2, 3, 4, ...

5. Enumerations must have a beginning: ..., 3, 2, 1 is not an enumeration of the positive integers because it has no first element. To see how this follows from the informal definition, ask yourself, "at what position in the list does the number 76 appear?"

6. The following is not an enumeration of the positive integers: 1, 3, 5, ..., 2, 4, 6, ... The problem is that the even numbers occur at places $\infty + 1$, $\infty + 2$, $\infty + 3$, rather than at finite positions.

7. The empty set is enumerable: it is enumerated by the empty list!

Proposition 4.2. *If A has an enumeration, it has an enumeration without repetitions.*

Proof. Suppose A has an enumeration $x_1, x_2, \ldots$ in which each x_i is an element of A. We can remove repetitions from an enumeration by removing repeated elements. For instance, we can turn the enumeration into a new one in which we list x_i if it is an element of A that is not among $x_1, \ldots, x_{i-1}$ or remove x_i from the list if it already appears among $x_1, \ldots, x_{i-1}$. □

The last argument shows that in order to get a good handle on enumerations and enumerable sets and to prove things about them, we need a more precise definition. The following provides it.

Definition 4.3 (Enumeration, formally). An *enumeration* of a set $A \neq \emptyset$ is any surjective function $f\colon \mathbb{Z}^+ \to A$.

explanation

Let's convince ourselves that the formal definition and the informal definition using a possibly infinite list are equivalent. First, any surjective function from $\mathbb{Z}^+$ to a set A enumerates A. Such a function determines an enumeration as defined informally above: the list $f(1), f(2), f(3), \ldots$. Since f is surjective, every element of A is guaranteed to be the value of $f(n)$ for some $n \in \mathbb{Z}^+$. Hence, every element of A appears at some finite position in the list. Since the function may not be injective, the list may be redundant, but that is acceptable (as noted above).

On the other hand, given a list that enumerates all elements of A, we can define a surjective function $f\colon \mathbb{Z}^+ \to A$ by letting $f(n)$ be the nth element of

the list, or the final element of the list if there is no nth element. The only case where this does not produce a surjective function is when A is empty, and hence the list is empty. So, every non-empty list determines a surjective function $f\colon \mathbb{Z}^+ \to A$.

sfr:siz:enm: defn:enumerable

Definition 4.4. A set A is enumerable iff it is empty or has an enumeration.

Example 4.5. A function enumerating the positive integers ($\mathbb{Z}^+$) is simply the identity function given by $f(n) = n$. A function enumerating the natural numbers $\mathbb{N}$ is the function $g(n) = n - 1$.

Example 4.6. The functions $f\colon \mathbb{Z}^+ \to \mathbb{Z}^+$ and $g\colon \mathbb{Z}^+ \to \mathbb{Z}^+$ given by

$$f(n) = 2n \text{ and}$$
$$g(n) = 2n - 1$$

enumerate the even positive integers and the odd positive integers, respectively. However, neither function is an enumeration of $\mathbb{Z}^+$, since neither is surjective.

Problem 4.1. Define an enumeration of the positive squares 1, 4, 9, 16, ...

Example 4.7. The function $f(n) = (-1)^n \lceil \frac{(n-1)}{2} \rceil$ (where $\lceil x \rceil$ denotes the *ceiling* function, which rounds x up to the nearest integer) enumerates the set of integers $\mathbb{Z}$. Notice how f generates the values of $\mathbb{Z}$ by "hopping" back and forth between positive and negative integers:

$f(1)$	$f(2)$	$f(3)$	$f(4)$	$f(5)$	$f(6)$	$f(7)$	...
$-\lceil \frac{0}{2} \rceil$	$\lceil \frac{1}{2} \rceil$	$-\lceil \frac{2}{2} \rceil$	$\lceil \frac{3}{2} \rceil$	$-\lceil \frac{4}{2} \rceil$	$\lceil \frac{5}{2} \rceil$	$-\lceil \frac{6}{2} \rceil$	...
0	1	-1	2	-2	3	...	

You can also think of f as defined by cases as follows:

$$f(n) = \begin{cases} 0 & \text{if } n = 1 \\ n/2 & \text{if } n \text{ is even} \\ -(n-1)/2 & \text{if } n \text{ is odd and } > 1 \end{cases}$$

Problem 4.2. Show that if A and B are enumerable, so is $A \cup B$. To do this, suppose there are surjective functions $f\colon \mathbb{Z}^+ \to A$ and $g\colon \mathbb{Z}^+ \to B$, and define a surjective function $h\colon \mathbb{Z}^+ \to A \cup B$ and prove that it is surjective. Also consider the cases where A or $B = \emptyset$.

Problem 4.3. Show that if $B \subseteq A$ and A is enumerable, so is B. To do this, suppose there is a surjective function $f\colon \mathbb{Z}^+ \to A$. Define a surjective function $g\colon \mathbb{Z}^+ \to B$ and prove that it is surjective. What happens if $B = \emptyset$?

Problem 4.4. Show by induction on n that if A_1, A_2, ..., A_n are all enumerable, so is $A_1 \cup \cdots \cup A_n$. You may assume the fact that if two sets A and B are enumerable, so is $A \cup B$.

Although it is perhaps more natural when listing the elements of a set to start counting from the 1st element, mathematicians like to use the natural numbers $\mathbb{N}$ for counting things. They talk about the 0th, 1st, 2nd, and so on, elements of a list. Correspondingly, we can define an enumeration as a surjective function from $\mathbb{N}$ to A. Of course, the two definitions are equivalent.

Proposition 4.8. *There is a surjection $f\colon \mathbb{Z}^+ \to A$ iff there is a surjection $g\colon \mathbb{N} \to A$.* *sfr:siz:enm: prop:enum-shift*

Proof. Given a surjection $f\colon \mathbb{Z}^+ \to A$, we can define $g(n) = f(n+1)$ for all $n \in \mathbb{N}$. It is easy to see that $g\colon \mathbb{N} \to A$ is surjective. Conversely, given a surjection $g\colon \mathbb{N} \to A$, define $f(n) = g(n-1)$. □

This gives us the following result:

Corollary 4.9. *A set A is enumerable iff it is empty or there is a surjective function $f\colon \mathbb{N} \to A$.* *sfr:siz:enm: cor:enum-nat*

We discussed above that a list of elements of a set A can be turned into a list without repetitions. This is also true for enumerations, but a bit harder to formulate and prove rigorously. Any function $f\colon \mathbb{Z}^+ \to A$ must be defined for all $n \in \mathbb{Z}^+$. If there are only finitely many elements in A then we clearly cannot have a function defined on the infinitely many elements of $\mathbb{Z}^+$ that takes as values all the elements of A but never takes the same value twice. In that case, i.e., in the case where the list without repetitions is finite, we must choose a different domain for f, one with only finitely many elements. Not having repetitions means that f must be injective. Since it is also surjective, we are looking for a bijection between some finite set $\{1, \ldots, n\}$ or $\mathbb{Z}^+$ and A.

Proposition 4.10. *If $f\colon \mathbb{Z}^+ \to A$ is surjective (i.e., an enumeration of A), there is a bijection $g\colon Z \to A$ where Z is either $\mathbb{Z}^+$ or $\{1, \ldots, n\}$ for some $n \in \mathbb{Z}^+$.* *sfr:siz:enm: prop:enum-bij*

Proof. We define the function g recursively: Let $g(1) = f(1)$. If $g(i)$ has already been defined, let $g(i+1)$ be the first value of $f(1), f(2), \ldots$ not already among $g(1), \ldots, g(i)$, if there is one. If A has just n elements, then $g(1), \ldots, g(n)$ are all defined, and so we have defined a function $g\colon \{1, \ldots, n\} \to A$. If A has infinitely many elements, then for any i there must be an element of A in the enumeration $f(1), f(2), \ldots$, which is not already among $g(1), \ldots, g(i)$. In this case we have defined a function $g\colon \mathbb{Z}^+ \to A$.

The function g is surjective, since any element of A is among $f(1), f(2), \ldots$ (since f is surjective) and so will eventually be a value of $g(i)$ for some i. It is also injective, since if there were $j < i$ such that $g(j) = g(i)$, then $g(i)$ would already be among $g(1), \ldots, g(i-1)$, contrary to how we defined g. □

sfr:siz:enm: cor:enum-nat-bij

Corollary 4.11. *A set A is enumerable iff it is empty or there is a bijection $f\colon N \to A$ where either $N = \mathbb{N}$ or $N = \{0, \dots, n\}$ for some $n \in \mathbb{N}$.*

Proof. A is enumerable iff A is empty or there is a surjective $f\colon \mathbb{Z}^+ \to A$. By Proposition 4.10, the latter holds iff there is a bijective function $f\colon Z \to A$ where $Z = \mathbb{Z}^+$ or $Z = \{1, \dots, n\}$ for some $n \in \mathbb{Z}^+$. By the same argument as in the proof of Proposition 4.8, that in turn is the case iff there is a bijection $g\colon N \to A$ where either $N = \mathbb{N}$ or $N = \{0, \dots, n-1\}$. □

Problem 4.5. According to Definition 4.4, a set A is enumerable iff $A = \emptyset$ or there is a surjective $f\colon \mathbb{Z}^+ \to A$. It is also possible to define "enumerable set" precisely by: a set is enumerable iff there is an injective function $g\colon A \to \mathbb{Z}^+$. Show that the definitions are equivalent, i.e., show that there is an injective function $g\colon A \to \mathbb{Z}^+$ iff either $A = \emptyset$ or there is a surjective $f\colon \mathbb{Z}^+ \to A$.

---

content/sets-functions-relations/size-of-sets/zig-zag.tex

4.3 Cantor's Zig-Zag Method

sfr:siz:zigzag: sec

We've already considered some "easy" enumerations. Now we will consider something a bit harder. Consider the set of pairs of natural numbers, which we defined in section 1.5 thus: explanation

$$\mathbb{N} \times \mathbb{N} = \{\langle n, m\rangle : n, m \in \mathbb{N}\}$$

We can organize these ordered pairs into an *array*, like so:

	0	**1**	**2**	**3**	...
0	$\langle 0,0\rangle$	$\langle 0,1\rangle$	$\langle 0,2\rangle$	$\langle 0,3\rangle$	...
1	$\langle 1,0\rangle$	$\langle 1,1\rangle$	$\langle 1,2\rangle$	$\langle 1,3\rangle$	...
2	$\langle 2,0\rangle$	$\langle 2,1\rangle$	$\langle 2,2\rangle$	$\langle 2,3\rangle$	...
3	$\langle 3,0\rangle$	$\langle 3,1\rangle$	$\langle 3,2\rangle$	$\langle 3,3\rangle$	...
⋮	⋮	⋮	⋮	⋮	⋱

Clearly, every ordered pair in $\mathbb{N} \times \mathbb{N}$ will appear exactly once in the array. In particular, $\langle n, m\rangle$ will appear in the nth row and mth column. But how do we organize the elements of such an array into a "one-dimensional" list? The pattern in the array below demonstrates one way to do this (although of course there are many other options):

	0	**1**	**2**	**3**	**4**	...
0	0	1	3	6	10	...
1	2	4	7	11	...	...
2	5	8	12	...	...	...
3	9	13	...	...	...	...
4	14	...	...	...	...	...
⋮	⋮	⋮	⋮	⋮	...	⋱

This pattern is called *Cantor's zig-zag method*. It enumerates $\mathbb{N} \times \mathbb{N}$ as follows:

$$\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle, \langle 0,2\rangle, \langle 1,1\rangle, \langle 2,0\rangle, \langle 0,3\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \langle 3,0\rangle, \ldots$$

And this establishes the following:

Proposition 4.12. $\mathbb{N} \times \mathbb{N}$ *is enumerable.*

sfr:siz:zigzag: natsquaredenumerable

Proof. Let $f\colon \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ take each $k \in \mathbb{N}$ to the tuple $\langle n, m\rangle \in \mathbb{N} \times \mathbb{N}$ such that k is the value of the nth row and mth column in Cantor's zig-zag array. □

explanation

This technique also generalises rather nicely. For example, we can use it to enumerate the set of ordered triples of natural numbers, i.e.:

$$\mathbb{N} \times \mathbb{N} \times \mathbb{N} = \{\langle n, m, k\rangle : n, m, k \in \mathbb{N}\}$$

We think of $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ as the Cartesian product of $\mathbb{N} \times \mathbb{N}$ with $\mathbb{N}$, that is,

$$\mathbb{N}^3 = (\mathbb{N} \times \mathbb{N}) \times \mathbb{N} = \{\langle\langle n, m\rangle, k\rangle : n, m, k \in \mathbb{N}\}$$

and thus we can enumerate $\mathbb{N}^3$ with an array by labelling one axis with the enumeration of $\mathbb{N}$, and the other axis with the enumeration of $\mathbb{N}^2$:

	0	**1**	**2**	**3**	...
$\langle \mathbf{0},\mathbf{0}\rangle$	$\langle 0,0,0\rangle$	$\langle 0,0,1\rangle$	$\langle 0,0,2\rangle$	$\langle 0,0,3\rangle$	...
$\langle \mathbf{0},\mathbf{1}\rangle$	$\langle 0,1,0\rangle$	$\langle 0,1,1\rangle$	$\langle 0,1,2\rangle$	$\langle 0,1,3\rangle$	...
$\langle \mathbf{1},\mathbf{0}\rangle$	$\langle 1,0,0\rangle$	$\langle 1,0,1\rangle$	$\langle 1,0,2\rangle$	$\langle 1,0,3\rangle$	...
$\langle \mathbf{0},\mathbf{2}\rangle$	$\langle 0,2,0\rangle$	$\langle 0,2,1\rangle$	$\langle 0,2,2\rangle$	$\langle 0,2,3\rangle$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Thus, by using a method like Cantor's zig-zag method, we may similarly obtain an enumeration of $\mathbb{N}^3$. And we can keep going, obtaining enumerations of $\mathbb{N}^n$ for any natural number n. So, we have:

Proposition 4.13. $\mathbb{N}^n$ *is enumerable, for every* $n \in \mathbb{N}$.

Problem 4.6. Show that $(\mathbb{Z}^+)^n$ is enumerable, for every $n \in \mathbb{N}$.

Problem 4.7. Show that $(\mathbb{Z}^+)^*$ is enumerable. You may assume Problem 4.6.

4.4 Pairing Functions and Codes

sfr:siz:pai:sec

Cantor's zig-zag method makes the enumerability of $\mathbb{N}^n$ visually evident. But let us focus on our array depicting $\mathbb{N}^2$. Following the zig-zag line in the array and counting the places, we can check that $\langle 1, 2\rangle$ is associated with the number 7. However, it would be nice if we could compute this more directly. That is, it would be nice to have to hand the *inverse* of the zig-zag enumeration, $g\colon \mathbb{N}^2 \to \mathbb{N}$, such that explanation

$$g(\langle 0, 0\rangle) = 0,\ g(\langle 0, 1\rangle) = 1,\ g(\langle 1, 0\rangle) = 2,\ \ldots, g(\langle 1, 2\rangle) = 7,\ \ldots$$

This would enable us to calculate exactly where $\langle n, m\rangle$ will occur in our enumeration.

In fact, we can define g directly by making two observations. First: if the nth row and mth column contains value v, then the $(n+1)$st row and $(m-1)$st column contains value $v+1$. Second: the first row of our enumeration consists of the triangular numbers, starting with 0, 1, 3, 6, etc. The kth triangular number is the sum of the natural numbers $< k$, which can be computed as $k(k+1)/2$. Putting these two observations together, consider this function:

$$g(n, m) = \frac{(n + m + 1)(n + m)}{2} + n$$

We often just write $g(n, m)$ rather that $g(\langle n, m\rangle)$, since it is easier on the eyes. This tells you first to determine the $(n + m)^{\text{th}}$ triangle number, and then add n to it. And it populates the array in exactly the way we would like. So in particular, the pair $\langle 1, 2\rangle$ is sent to $\frac{4\times 3}{2} + 1 = 7$.

This function g is the *inverse* of an enumeration of a set of pairs. Such functions are called *pairing functions*.

Definition 4.14 (Pairing function). A function $f\colon A \times B \to \mathbb{N}$ is an arithmetical *pairing function* if f is injective. We also say that f *encodes* $A \times B$, and that $f(x, y)$ is the *code* for $\langle x, y\rangle$.

We can use pairing functions to encode, e.g., pairs of natural numbers; or, in other words, we can represent each *pair* of elements using a *single* number. Using the inverse of the pairing function, we can *decode* the number, i.e., find out which pair it represents. explanation

Problem 4.8. Give an enumeration of the set of all non-negative rational numbers.

Problem 4.9. Show that $\mathbb{Q}$ is enumerable. Recall that any rational number can be written as a fraction z/m with $z \in \mathbb{Z}$, $m \in \mathbb{N}^+$.

Problem 4.10. Define an enumeration of $\mathbb{B}^*$.

Problem 4.11. Recall from your introductory logic course that each possible truth table expresses a truth function. In other words, the truth functions are all functions from $\mathbb{B}^k \to \mathbb{B}$ for some k. Prove that the set of all truth functions is enumerable.

Problem 4.12. Show that the set of all finite subsets of an arbitrary infinite enumerable set is enumerable.

Problem 4.13. A subset of $\mathbb{N}$ is said to be *cofinite* iff it is the complement of a finite set $\mathbb{N}$; that is, $A \subseteq \mathbb{N}$ is cofinite iff $\mathbb{N} \setminus A$ is finite. Let I be the set whose elements are exactly the finite and cofinite subsets of $\mathbb{N}$. Show that I is enumerable.

Problem 4.14. Show that the enumerable union of enumerable sets is enumerable. That is, whenever A_1, A_2, ... are sets, and each A_i is enumerable, then the union $\bigcup_{i=1}^{\infty} A_i$ of all of them is also enumerable. [NB: this is hard!]

Problem 4.15. Let $f\colon A \times B \to \mathbb{N}$ be an arbitrary pairing function. Show that the inverse of f is an enumeration of $A \times B$.

Problem 4.16. Specify a function that encodes $\mathbb{N}^3$.

---

content/sets-functions-relations/size-of-sets/pairing-alt.tex

4.5 An Alternative Pairing Function

explanation

There are other enumerations of $\mathbb{N}^2$ that make it easier to figure out what their inverses are. Here is one. Instead of visualizing the enumeration in an array, start with the list of positive integers associated with (initially) empty spaces. Imagine filling these spaces successively with pairs $\langle n, m\rangle$ as follows. Starting with the pairs that have 0 in the first place (i.e., pairs $\langle 0, m\rangle$), put the first (i.e., $\langle 0, 0\rangle$) in the first empty place, then skip an empty space, put the second (i.e., $\langle 0, 2\rangle$) in the next empty place, skip one again, and so forth. The (incomplete) beginning of our enumeration now looks like this

sfr:siz:pai-alt:sec

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	...
$\langle 0,0\rangle$		$\langle 0,1\rangle$		$\langle 0,2\rangle$		$\langle 0,3\rangle$		$\langle 0,4\rangle$		...

Repeat this with pairs $\langle 1, m\rangle$ for the place that still remain empty, again skipping every other empty place:

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	...
$\langle 0,0\rangle$	$\langle 1,0\rangle$	$\langle 0,1\rangle$		$\langle 0,2\rangle$	$\langle 1,1\rangle$	$\langle 0,3\rangle$		$\langle 0,4\rangle$	$\langle 1,2\rangle$	...

Enter pairs $\langle 2, m\rangle$, $\langle 2, m\rangle$, etc., in the same way. Our completed enumeration thus starts like this:

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	...
$\langle 0,0\rangle$	$\langle 1,0\rangle$	$\langle 0,1\rangle$	$\langle 2,0\rangle$	$\langle 0,2\rangle$	$\langle 1,1\rangle$	$\langle 0,3\rangle$	$\langle 3,0\rangle$	$\langle 0,4\rangle$	$\langle 1,2\rangle$	...

If we number the cells in the array above according to this enumeration, we will not find a neat zig-zag line, but this arrangement:

	0	**1**	**2**	**3**	**4**	**5**	...
0	1	3	5	7	9	11	...
1	2	6	10	14	18	...	...
2	4	12	20	28	...	...	...
3	8	24	40	...	...	...	...
4	16	48	...	...	...	...	...
5	32	...	...	...	...	...	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

We can see that the pairs in row 0 are in the odd numbered places of our enumeration, i.e., pair $\langle 0, m\rangle$ is in place $2m + 1$; pairs in the second row, $\langle 1, m\rangle$, are in places whose number is the double of an odd number, specifically, $2 \cdot (2m + 1)$; pairs in the third row, $\langle 2, m\rangle$, are in places whose number is four times an odd number, $4 \cdot (2m + 1)$; and so on. The factors of $(2m + 1)$ for each row, 1, 2, 4, 8, ..., are exactly the powers of 2: $1 = 2^0$, $2 = 2^1$, $4 = 2^2$, $8 = 2^3$, ... In fact, the relevant exponent is always the first member of the pair in question. Thus, for pair $\langle n, m\rangle$ the factor is 2^n. This gives us the general formula: $2^n \cdot (2m+1)$. However, this is a mapping of pairs to *positive* integers, i.e., $\langle 0, 0\rangle$ has position 1. If we want to begin at position 0 we must subtract 1 from the result. This gives us:

Example 4.15. The function $h\colon \mathbb{N}^2 \to \mathbb{N}$ given by

$$h(n, m) = 2^n(2m + 1) - 1$$

is a pairing function for the set of pairs of natural numbers $\mathbb{N}^2$.

Accordingly, in our second enumeration of $\mathbb{N}^2$, the pair $\langle 0, 0\rangle$ has code $h(0, 0) = 2^0(2 \cdot 0 + 1) - 1 = 0$; $\langle 1, 2\rangle$ has code $2^1 \cdot (2 \cdot 2 + 1) - 1 = 2 \cdot 5 - 1 = 9$; $\langle 2, 6\rangle$ has code $2^2 \cdot (2 \cdot 6 + 1) - 1 = 51$. explanation

Sometimes it is enough to encode pairs of natural numbers $\mathbb{N}^2$ without requiring that the encoding is surjective. Such encodings have inverses that are only partial functions.

Example 4.16. The function $j\colon \mathbb{N}^2 \to \mathbb{N}^+$ given by

$$j(n, m) = 2^n 3^m$$

is an injective function $\mathbb{N}^2 \to \mathbb{N}$.

---

content/sets-functions-relations/size-of-sets/non-enumerability.tex

4.6 Non-enumerable Sets

sfr:siz:nen:sec

This section proves the non-enumerability of $\mathbb{B}^\omega$ and $\wp(\mathbb{Z}^+)$ using the definition in section 4.2. It is designed to be a little more elementary and a little more detailed than the version in section 4.11

Some sets, such as the set $\mathbb{Z}^+$ of positive integers, are infinite. So far we've seen examples of infinite sets which were all enumerable. However, there are also infinite sets which do not have this property. Such sets are called *non-enumerable*.

First of all, it is perhaps already surprising that there are non-enumerable sets. For any enumerable set A there is a surjective function $f\colon \mathbb{Z}^+ \to A$. If a set is non-enumerable there is no such function. That is, no function mapping the infinitely many elements of $\mathbb{Z}^+$ to A can exhaust all of A. So there are "more" elements of A than the infinitely many positive integers.

How would one prove that a set is non-enumerable? You have to show that no such surjective function can exist. Equivalently, you have to show that the elements of A cannot be enumerated in a one way infinite list. The best way to do this is to show that every list of elements of A must leave at least one element out; or that no function $f\colon \mathbb{Z}^+ \to A$ can be surjective. We can do this using Cantor's *diagonal method*. Given a list of elements of A, say, x_1, x_2, ..., we construct another element of A which, by its construction, cannot possibly be on that list.

Our first example is the set $\mathbb{B}^\omega$ of all infinite, non-gappy sequences of 0's and 1's.

Theorem 4.17. *$\mathbb{B}^\omega$ is non-enumerable.*

sfr:siz:nen:thm:nonenum-bin-omega

Proof. Suppose, by way of contradiction, that $\mathbb{B}^\omega$ is enumerable, i.e., suppose that there is a list s_1, s_2, s_3, s_4, ... of all elements of $\mathbb{B}^\omega$. Each of these s_i is itself an infinite sequence of 0's and 1's. Let's call the j-th element of the i-th sequence in this list $s_i(j)$. Then the i-th sequence s_i is

$$s_i(1), s_i(2), s_i(3), \ldots$$

We may arrange this list, and the elements of each sequence s_i in it, in an array:

	1	2	3	4	...
1	$\mathbf{s_1(1)}$	$s_1(2)$	$s_1(3)$	$s_1(4)$	...
2	$s_2(1)$	$\mathbf{s_2(2)}$	$s_2(3)$	$s_2(4)$	...
3	$s_3(1)$	$s_3(2)$	$\mathbf{s_3(3)}$	$s_3(4)$	...
4	$s_4(1)$	$s_4(2)$	$s_4(3)$	$\mathbf{s_4(4)}$	...
⋮	⋮	⋮	⋮	⋮	⋱

The labels down the side give the number of the sequence in the list s_1, s_2, ...; the numbers across the top label the elements of the individual sequences. For instance, $s_1(1)$ is a name for whatever number, a 0 or a 1, is the first element in the sequence s_1, and so on.

Now we construct an infinite sequence, $\overline{s}$, of 0's and 1's which cannot possibly be on this list. The definition of $\overline{s}$ will depend on the list s_1, s_2, ... Any infinite list of infinite sequences of 0's and 1's gives rise to an infinite sequence $\overline{s}$ which is guaranteed to not appear on the list.

To define $\overline{s}$, we specify what all its elements are, i.e., we specify $\overline{s}(n)$ for all $n \in \mathbb{Z}^+$. We do this by reading down the diagonal of the array above (hence the name "diagonal method") and then changing every 1 to a 0 and every 0 to a 1. More abstractly, we define $\overline{s}(n)$ to be 0 or 1 according to whether the n-th element of the diagonal, $s_n(n)$, is 1 or 0.

$$\overline{s}(n) = \begin{cases} 1 & \text{if } s_n(n) = 0 \\ 0 & \text{if } s_n(n) = 1. \end{cases}$$

If you like formulas better than definitions by cases, you could also define $\overline{s}(n) = 1 - s_n(n)$.

Clearly $\overline{s}$ is an infinite sequence of 0's and 1's, since it is just the mirror sequence to the sequence of 0's and 1's that appear on the diagonal of our array. So $\overline{s}$ is an element of $\mathbb{B}^\omega$. But it cannot be on the list s_1, s_2, ... Why not?

It can't be the first sequence in the list, s_1, because it differs from s_1 in the first element. Whatever $s_1(1)$ is, we defined $\overline{s}(1)$ to be the opposite. It can't be the second sequence in the list, because $\overline{s}$ differs from s_2 in the second element: if $s_2(2)$ is 0, $\overline{s}(2)$ is 1, and vice versa. And so on.

More precisely: if $\overline{s}$ were on the list, there would be some k so that $\overline{s} = s_k$. Two sequences are identical iff they agree at every place, i.e., for any n, $\overline{s}(n) = s_k(n)$. So in particular, taking $n = k$ as a special case, $\overline{s}(k) = s_k(k)$ would have to hold. $s_k(k)$ is either 0 or 1. If it is 0 then $\overline{s}(k)$ must be 1—that's how we defined $\overline{s}$. But if $s_k(k) = 1$ then, again because of the way we defined $\overline{s}$, $\overline{s}(k) = 0$. In either case $\overline{s}(k) \neq s_k(k)$.

We started by assuming that there is a list of elements of $\mathbb{B}^\omega$, s_1, s_2, ... From this list we constructed a sequence $\overline{s}$ which we proved cannot be on the list. But it definitely is a sequence of 0's and 1's if all the s_i are sequences of 0's and 1's, i.e., $\overline{s} \in \mathbb{B}^\omega$. This shows in particular that there can be no list of *all* elements of $\mathbb{B}^\omega$, since for any such list we could also construct a sequence $\overline{s}$ guaranteed to not be on the list, so the assumption that there is a list of all sequences in $\mathbb{B}^\omega$ leads to a contradiction. □

This proof method is called "diagonalization" because it uses the diagonal of the array to define $\overline{s}$. Diagonalization need not involve the presence of an array: we can show that sets are not enumerable by using a similar idea even when no array and no actual diagonal is involved. explanation

sfr:siz:nen: thm:nonenum-pownat

Theorem 4.18. *$\wp(\mathbb{Z}^+)$ is not enumerable.*

Proof. We proceed in the same way, by showing that for every list of subsets of $\mathbb{Z}^+$ there is a subset of $\mathbb{Z}^+$ which cannot be on the list. Suppose the following is a given list of subsets of $\mathbb{Z}^+$:

$$Z_1, Z_2, Z_3, \ldots$$

We now define a set $\overline{Z}$ such that for any $n \in \mathbb{Z}^+$, $n \in \overline{Z}$ iff $n \notin Z_n$:

$$\overline{Z} = \{n \in \mathbb{Z}^+ : n \notin Z_n\}$$ □

$\overline{Z}$ is clearly a set of positive integers, since by assumption each Z_n is, and thus $\overline{Z} \in \wp(\mathbb{Z}^+)$. But $\overline{Z}$ cannot be on the list. To show this, we'll establish that for each $k \in \mathbb{Z}^+$, $\overline{Z} \neq Z_k$.

So let $k \in \mathbb{Z}^+$ be arbitrary. We've defined $\overline{Z}$ so that for any $n \in \mathbb{Z}^+$, $n \in \overline{Z}$ iff $n \notin Z_n$. In particular, taking $n = k$, $k \in \overline{Z}$ iff $k \notin Z_k$. But this shows that $\overline{Z} \neq Z_k$, since k is an element of one but not the other, and so $\overline{Z}$ and Z_k have different elements. Since k was arbitrary, $\overline{Z}$ is not on the list Z_1, Z_2, ...

explanation

The preceding proof did not mention a diagonal, but you can think of it as involving a diagonal if you picture it this way: Imagine the sets Z_1, Z_2, ..., written in an array, where each element $j \in Z_i$ is listed in the j-th column. Say the first four sets on that list are $\{1, 2, 3, \ldots\}$, $\{2, 4, 6, \ldots\}$, $\{1, 2, 5\}$, and $\{3, 4, 5, \ldots\}$. Then the array would begin with

$$\begin{array}{lccccccc}
Z_1 = \{ & \mathbf{1}, & 2, & 3, & 4, & 5, & 6, & \ldots\} \\
Z_2 = \{ & & \mathbf{2}, & & 4, & & 6, & \ldots\} \\
Z_3 = \{ & 1, & 2, & & & 5 & & \} \\
Z_4 = \{ & & & 3, & \mathbf{4}, & 5, & 6, & \ldots\} \\
\vdots & & & & & \ddots & &
\end{array}$$

Then $\overline{Z}$ is the set obtained by going down the diagonal, leaving out any numbers that appear along the diagonal and include those j where the array has a gap in the j-th row/column. In the above case, we would leave out 1 and 2, include 3, leave out 4, etc.

Problem 4.17. Show that $\wp(\mathbb{N})$ is non-enumerable by a diagonal argument.

Problem 4.18. Show that the set of functions $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ is non-enumerable by an explicit diagonal argument. That is, show that if f_1, f_2, ..., is a list of functions and each $f_i : \mathbb{Z}^+ \to \mathbb{Z}^+$, then there is some $\overline{f} : \mathbb{Z}^+ \to \mathbb{Z}^+$ not on this list.

4.7 Reduction

sfr:siz:red:sec

This section proves non-enumerability by reduction, matching the results in section 4.6. An alternative, slightly more condensed version matching the results in section 4.12 is provided in section 4.13.

We showed $\wp(\mathbb{Z}^+)$ to be non-enumerable by a diagonalization argument. We already had a proof that $\mathbb{B}^\omega$, the set of all infinite sequences of 0s and 1s, is non-enumerable. Here's another way we can prove that $\wp(\mathbb{Z}^+)$ is non-enumerable: Show that *if* $\wp(\mathbb{Z}^+)$ *is enumerable then* $\mathbb{B}^\omega$ *is also enumerable*. Since we know $\mathbb{B}^\omega$ is not enumerable, $\wp(\mathbb{Z}^+)$ can't be either. This is called *reducing* one problem to another—in this case, we reduce the problem of enumerating $\mathbb{B}^\omega$ to the problem of enumerating $\wp(\mathbb{Z}^+)$. A solution to the latter—an enumeration of $\wp(\mathbb{Z}^+)$—would yield a solution to the former—an enumeration of $\mathbb{B}^\omega$.

How do we reduce the problem of enumerating a set B to that of enumerating a set A? We provide a way of turning an enumeration of A into an enumeration of B. The easiest way to do that is to define a surjective function $f\colon A \to B$. If $x_1, x_2, \ldots$ enumerates A, then $f(x_1), f(x_2), \ldots$ would enumerate B. In our case, we are looking for a surjective function $f\colon \wp(\mathbb{Z}^+) \to \mathbb{B}^\omega$.

Problem 4.19. Show that if there is an injective function $g\colon B \to A$, and B is non-enumerable, then so is A. Do this by showing how you can use g to turn an enumeration of A into one of B.

Proof of Theorem 4.18 by reduction. Suppose that $\wp(\mathbb{Z}^+)$ were enumerable, and thus that there is an enumeration of it, $Z_1, Z_2, Z_3, \ldots$

Define the function $f\colon \wp(\mathbb{Z}^+) \to \mathbb{B}^\omega$ by letting $f(Z)$ be the sequence s_k such that $s_k(n) = 1$ iff $n \in Z$, and $s_k(n) = 0$ otherwise. This clearly defines a function, since whenever $Z \subseteq \mathbb{Z}^+$, any $n \in \mathbb{Z}^+$ either is an element of Z or isn't. For instance, the set $2\mathbb{Z}^+ = \{2, 4, 6, \ldots\}$ of positive even numbers gets mapped to the sequence $010101\ldots$, the empty set gets mapped to $0000\ldots$ and the set $\mathbb{Z}^+$ itself to $1111\ldots$.

It also is surjective: Every sequence of 0s and 1s corresponds to some set of positive integers, namely the one which has as its members those integers corresponding to the places where the sequence has 1s. More precisely, suppose $s \in \mathbb{B}^\omega$. Define $Z \subseteq \mathbb{Z}^+$ by:

$$Z = \{n \in \mathbb{Z}^+ : s(n) = 1\}$$

Then $f(Z) = s$, as can be verified by consulting the definition of f.

Now consider the list

$$f(Z_1), f(Z_2), f(Z_3), \ldots$$

Since f is surjective, every member of $\mathbb{B}^\omega$ must appear as a value of f for some argument, and so must appear on the list. This list must therefore enumerate all of $\mathbb{B}^\omega$.

So if $\wp(\mathbb{Z}^+)$ were enumerable, $\mathbb{B}^\omega$ would be enumerable. But $\mathbb{B}^\omega$ is non-enumerable (Theorem 4.17). Hence $\wp(\mathbb{Z}^+)$ is non-enumerable. □

explanation

It is easy to be confused about the direction the reduction goes in. For instance, a surjective function $g\colon \mathbb{B}^\omega \to B$ does *not* establish that B is non-enumerable. (Consider $g\colon \mathbb{B}^\omega \to \mathbb{B}$ defined by $g(s) = s(1)$, the function that maps a sequence of 0's and 1's to its first element. It is surjective, because some sequences start with 0 and some start with 1. But $\mathbb{B}$ is finite.) Note also that the function f must be surjective, or otherwise the argument does not go through: $f(x_1)$, $f(x_2)$, ... would then not be guaranteed to include all the elements of B. For instance,

$$h(n) = \underbrace{000\ldots0}_{n\ 0\text{'s}}$$

defines a function $h\colon \mathbb{Z}^+ \to \mathbb{B}^\omega$, but $\mathbb{Z}^+$ is enumerable.

Problem 4.20. Show that the set of all *sets of* pairs of positive integers is non-enumerable by a reduction argument.

Problem 4.21. Show that the set X of all functions $f\colon \mathbb{N} \to \mathbb{N}$ is non-enumerable by a reduction argument (Hint: give a surjective function from X to $\mathbb{B}^\omega$.)

Problem 4.22. Show that $\mathbb{N}^\omega$, the set of infinite sequences of natural numbers, is non-enumerable by a reduction argument.

Problem 4.23. Let P be the set of functions from the set of positive integers to the set $\{0\}$, and let Q be the set of *partial* functions from the set of positive integers to the set $\{0\}$. Show that P is enumerable and Q is not. (Hint: reduce the problem of enumerating $\mathbb{B}^\omega$ to enumerating Q).

Problem 4.24. Let S be the set of all surjective functions from the set of positive integers to the set $\{0,1\}$, i.e., S consists of all surjective $f\colon \mathbb{Z}^+ \to \mathbb{B}$. Show that S is non-enumerable.

Problem 4.25. Show that the set $\mathbb{R}$ of all real numbers is non-enumerable.

---

content/sets-functions-relations/size-of-sets/equinumerous-sets.tex

4.8 Equinumerosity

sfr:siz:equ: sec

We have an intuitive notion of "size" of sets, which works fine for finite sets. But what about infinite sets? If we want to come up with a formal way of comparing the sizes of two sets of *any* size, it is a good idea to start by defining when sets are the same size. Here is Frege:

> If a waiter wants to be sure that he has laid exactly as many knives as plates on the table, he does not need to count either of them, if he simply lays a knife to the right of each plate, so that every knife on the table lies to the right of some plate. The plates and knives are thus uniquely correlated to each other, and indeed through that same spatial relationship. (Frege, 1884, §70)

The insight of this passage can be brought out through a formal definition:

sfr:siz:equ: comparisondef

Definition 4.19. A is *equinumerous* with B, written $A \approx B$, iff there is a bijection $f\colon A \to B$.

sfr:siz:equ: equinumerosityisequi

Proposition 4.20. *Equinumerosity is an equivalence relation.*

Proof. We must show that equinumerosity is reflexive, symmetric, and transitive. Let A, B, and C be sets.

Reflexivity. The identity map $\mathrm{Id}_A\colon A \to A$, where $\mathrm{Id}_A(x) = x$ for all $x \in A$, is a bijection. So $A \approx A$.

Symmetry. Suppose $A \approx B$, i.e., there is a bijection $f\colon A \to B$. Since f is bijective, its inverse f^{-1} exists and is also bijective. Hence, $f^{-1}\colon B \to A$ is a bijection, so $B \approx A$.

Transitivity. Suppose that $A \approx B$ and $B \approx C$, i.e., there are bijections $f\colon A \to B$ and $g\colon B \to C$. Then the composition $g \circ f\colon A \to C$ is bijective, so that $A \approx C$. □

Proposition 4.21. *If $A \approx B$, then A is enumerable if and only if B is.*

> The following proof uses Definition 4.4 if section 4.2 is included and Definition 4.27 otherwise.

Proof. Suppose $A \approx B$, so there is some bijection $f\colon A \to B$, and suppose that A is enumerable. Then either $A = \emptyset$ or there is a surjective function $g\colon \mathbb{Z}^+ \to A$. If $A = \emptyset$, then $B = \emptyset$ also (otherwise there would be an element $y \in B$ but no $x \in A$ with $g(x) = y$). If, on the other hand, $g\colon \mathbb{Z}^+ \to A$ is surjective, then $f \circ g\colon \mathbb{Z}^+ \to B$ is surjective. To see this, let $y \in B$. Since f is surjective, there is an $x \in A$ such that $f(x) = y$. Since g is surjective, there is an $n \in \mathbb{Z}^+$ such that $g(n) = x$. Hence,

$$(f \circ g)(n) = f(g(n)) = f(x) = y$$

and thus $f \circ g$ is surjective. We have that $f \circ g$ is an enumeration of B, and so B is enumerable.

If B is enumerable, we obtain that A is enumerable by repeating the argument with the bijection $f^{-1} \colon B \to A$ instead of f. □

Problem 4.26. Show that if $A \approx C$ and $B \approx D$, and $A \cap B = C \cap D = \emptyset$, then $A \cup B \approx C \cup D$.

Problem 4.27. Show that if A is infinite and enumerable, then $A \approx \mathbb{N}$.

---

content/sets-functions-relations/size-of-sets/comparing-size.tex

4.9 Sets of Different Sizes, and Cantor's Theorem

explanation

We have offered a precise statement of the idea that two sets have the same size. We can also offer a precise statement of the idea that one set is smaller than another. Our definition of "is smaller than (or equinumerous)" will require, instead of a bijection between the sets, an injection from the first set to the second. If such a function exists, the size of the first set is less than or equal to the size of the second. Intuitively, an injection from one set to another guarantees that the range of the function has at least as many elements as the domain, since no two elements of the domain map to the same element of the range.

sfr:siz:car:sec

Definition 4.22. A is *no larger than* B, written $A \preceq B$, iff there is an injection $f \colon A \to B$.

It is clear that this is a reflexive and transitive relation, but that it is not symmetric (this is left as an exercise). We can also introduce a notion, which states that one set is (strictly) smaller than another.

Definition 4.23. A is *smaller than* B, written $A \prec B$, iff there is an injection $f \colon A \to B$ but no bijection $g \colon A \to B$, i.e., $A \preceq B$ and $A \not\approx B$.

It is clear that this relation is irreflexive and transitive. (This is left as an exercise.) Using this notation, we can say that a set A is enumerable iff $A \preceq \mathbb{N}$, and that A is non-enumerable iff $\mathbb{N} \prec A$. This allows us to restate Theorem 4.32 as the observation that $\mathbb{N} \prec \wp(\mathbb{N})$. In fact, Cantor (1892) proved that this last point is *perfectly general*:

Theorem 4.24 (Cantor). *$A \prec \wp(A)$, for any set A.*

sfr:siz:car:thm:cantor

Proof. The map $f(x) = \{x\}$ is an injection $f \colon A \to \wp(A)$, since if $x \neq y$, then also $\{x\} \neq \{y\}$ by extensionality, and so $f(x) \neq f(y)$. So we have that $A \preceq \wp(A)$.

We present the slow proof if section 4.6 is present, otherwise a faster proof matching section 4.12.

We will now show that there cannot be a surjective function $g\colon A \to \wp(A)$, let alone a bijective one, and hence that $A \not\approx \wp(A)$. For suppose that $g\colon A \to \wp(A)$. Since g is total, every $x \in A$ is mapped to a subset $g(x) \subseteq A$. We can show that g cannot be surjective. To do this, we define a subset $\overline{A} \subseteq A$ which by definition cannot be in the range of g. Let

$$\overline{A} = \{x \in A : x \notin g(x)\}.$$

Since $g(x)$ is defined for all $x \in A$, $\overline{A}$ is clearly a well-defined subset of A. But, it cannot be in the range of g. Let $x \in A$ be arbitrary, we will show that $\overline{A} \neq g(x)$. If $x \in g(x)$, then it does not satisfy $x \notin g(x)$, and so by the definition of $\overline{A}$, we have $x \notin \overline{A}$. If $x \in \overline{A}$, it must satisfy the defining property of $\overline{A}$, i.e., $x \in A$ and $x \notin g(x)$. Since x was arbitrary, this shows that for each $x \in \overline{A}$, $x \in g(x)$ iff $x \notin \overline{A}$, and so $g(x) \neq \overline{A}$. In other words, $\overline{A}$ cannot be in the range of g, contradicting the assumption that g is surjective. □

explanation

It's instructive to compare the proof of Theorem 4.24 to that of Theorem 4.18. There we showed that for any list Z_1, Z_2, ..., of subsets of $\mathbb{Z}^+$ one can construct a set $\overline{Z}$ of numbers guaranteed not to be on the list. It was guaranteed not to be on the list because, for every $n \in \mathbb{Z}^+$, $n \in Z_n$ iff $n \notin \overline{Z}$. This way, there is always some number that is an element of one of Z_n or $\overline{Z}$ but not the other. We follow the same idea here, except the indices n are now elements of A instead of $\mathbb{Z}^+$. The set $\overline{A}$ is defined so that it is different from $g(x)$ for each $x \in A$, because $x \in g(x)$ iff $x \notin \overline{A}$. Again, there is always an element of A which is an element of one of $g(x)$ and $\overline{A}$ but not the other. And just as $\overline{Z}$ therefore cannot be on the list Z_1, Z_2, ..., $\overline{A}$ cannot be in the range of g.

It's instructive to compare the proof of Theorem 4.24 to that of Theorem 4.32. There we showed that for any list N_0, N_1, N_2, ..., of subsets of $\mathbb{N}$ we can construct a set D of numbers guaranteed not to be on the list. It was guaranteed not to be on the list because $n \in N_n$ iff $n \notin D$, for every $n \in \mathbb{N}$. We follow the same idea here, except the indices n are now elements of A rather than of $\mathbb{N}$. The set D is defined so that it is different from $g(x)$ for each $x \in A$, because $x \in g(x)$ iff $x \notin D$.

The proof is also worth comparing with the proof of Russell's Paradox, Theorem 1.29. Indeed, Cantor's Theorem was the inspiration for Russell's own paradox.

Problem 4.28. Show that there cannot be an injection $g\colon \wp(A) \to A$, for any set A. Hint: Suppose $g\colon \wp(A) \to A$ is injective. Consider $D = \{g(B) : B \subseteq A \text{ and } g(B) \notin B\}$. Let $x = g(D)$. Use the fact that g is injective to derive a contradiction.

content/sets-functions-relations/size-of-sets/schroder-bernstein.tex

4.10 The Notion of Size, and Schröder-Bernstein

explanation

Here is an intuitive thought: if A is no larger than B and B is no larger than A, then A and B are equinumerous. To be honest, if this thought were *wrong*, then we could scarcely justify the thought that our defined notion of equinumerosity has anything to do with comparisons of "sizes" between sets! Fortunately, though, the intuitive thought is correct. This is justified by the Schröder-Bernstein Theorem.

sfr:siz:sb:sec

Theorem 4.25 (Schröder-Bernstein). *If $A \preceq B$ and $B \preceq A$, then $A \approx B$.*

sfr:siz:sb:thm:schroder-bernstein

explanation

In other words, if there is an injection from A to B, and an injection from B to A, then there is a bijection from A to B.

This result, however, is really rather *difficult* to prove. Indeed, although Cantor stated the result, others proved it.[1] For now, you can (and must) take it on trust.

Fortunately, Schröder-Bernstein is *correct*, and it vindicates our thinking of the relations we defined, i.e., $A \approx B$ and $A \preceq B$, as having something to do with "size". Moreover, Schröder-Bernstein is very *useful*. It can be difficult to think of a bijection between two equinumerous sets. The Schröder-Bernstein Theorem allows us to break the comparison down into cases so we only have to think of an injection from the first to the second, and vice-versa.

> The following section 4.11, section 4.12, section 4.13 are alternative versions of section 4.2, section 4.6, section 4.7 due to Tim Button for use in his Open Set Theory text. They are slightly more advanced and use a difference definition of enumerability more suitable in a set theory context (i.e., bijection with $\mathbb{N}$ or an initial segment, rather than being listable or being the range of a surjective function from $\mathbb{Z}^+$).

---

content/sets-functions-relations/size-of-sets/enumerability-alt.tex

4.11 Enumerations and Enumerable Sets

sfr:siz:enm-alt:sec

> This section defines enumerations as bijections with (initial segments) of $\mathbb{N}$, the way it's done in set theory. So it conflicts slightly with the definitions in section 4.2, and repeats all the examples there. It is also a bit more terse than that section.

[1] For more on the history, see e.g., Potter (2004, pp. 165–6).

We can specify finite set is by simply enumerating its elements. We do this when we define a set like so:

$$A = \{a_1, a_2, \ldots, a_n\}.$$

Assuming that the elements $a_1, \ldots, a_n$ are all distinct, this gives us a bijection between A and the first n natural numbers $0, \ldots, n-1$. Conversely, since every finite set has only finitely many elements, every finite set can be put into such a correspondence. In other words, if A is finite, there is a bijection between A and $\{0, \ldots, n-1\}$, where n is the number of elements of A.

If we allow for certain kinds of infinite sets, then we will also allow some infinite sets to be enumerated. We can make this precise by saying that an infinite set is enumerated by a bijection between it and all of $\mathbb{N}$.

Definition 4.26 (Enumeration, set-theoretic). An *enumeration* of a set A is a bijection whose range is A and whose domain is either an initial set of natural numbers $\{0, 1, \ldots, n\}$ or the entire set of natural numbers $\mathbb{N}$.

There is an intuitive underpinning to this use of the word *enumeration*. For to say that we have enumerated a set A is to say that there is a bijection f which allows us to count out the elements of the set A. The 0th element is $f(0)$, the 1st is $f(1)$, ...the nth is $f(n)$....[2] The rationale for this may be made even clearer by adding the following: explanation

sfr:siz:enm-alt: defn:enumerable

Definition 4.27. A set A is enumerable iff either $A = \emptyset$ or there is an enumeration of A. We say that A is non-enumerable iff A is not enumerable.

So a set is enumerable iff it is empty or you can use an enumeration to count out its elements. explanation

Example 4.28. A function enumerating the natural numbers is simply the identity function $\mathrm{Id}_{\mathbb{N}} \colon \mathbb{N} \to \mathbb{N}$ given by $\mathrm{Id}_{\mathbb{N}}(n) = n$. A function enumerating the *positive* natural numbers, $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$, is the function $g(n) = n + 1$, i.e., the successor function.

Problem 4.29. Show that a set A is enumerable iff either $A = \emptyset$ or there is a surjection $f \colon \mathbb{N} \to A$. Show that A is enumerable iff there is an injection $g \colon A \to \mathbb{N}$.

Example 4.29. The functions $f \colon \mathbb{N} \to \mathbb{N}$ and $g \colon \mathbb{N} \to \mathbb{N}$ given by

$$f(n) = 2n \text{ and}$$
$$g(n) = 2n + 1$$

respectively enumerate the even natural numbers and the odd natural numbers. But neither is surjective, so neither is an enumeration of $\mathbb{N}$.

[2] Yes, we count from 0. Of course we could also start with 1. This would make no big difference. We would just have to replace $\mathbb{N}$ by $\mathbb{Z}^+$.

Problem 4.30. Define an enumeration of the square numbers 1, 4, 9, 16, ...

Example 4.30. Let $\lceil x \rceil$ be the *ceiling* function, which rounds x up to the nearest integer. Then the function $f\colon \mathbb{N} \to \mathbb{Z}$ given by:

$$f(n) = (-1)^n \left\lceil \tfrac{n}{2} \right\rceil$$

enumerates the set of integers $\mathbb{Z}$ as follows:

$f(0)$	$f(1)$	$f(2)$	$f(3)$	$f(4)$	$f(5)$	$f(6)$	...
$\lceil \frac{0}{2} \rceil$	$-\lceil \frac{1}{2} \rceil$	$\lceil \frac{2}{2} \rceil$	$-\lceil \frac{3}{2} \rceil$	$\lceil \frac{4}{2} \rceil$	$-\lceil \frac{5}{2} \rceil$	$\lceil \frac{6}{2} \rceil$	...
0	-1	1	-2	2	-3	3	...

Notice how f generates the values of $\mathbb{Z}$ by "hopping" back and forth between positive and negative integers. You can also think of f as defined by cases as follows:

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ -\frac{n+1}{2} & \text{if } n \text{ is odd} \end{cases}$$

Problem 4.31. Show that if A and B are enumerable, so is $A \cup B$.

Problem 4.32. Show by induction on n that if A_1, A_2, ..., A_n are all enumerable, so is $A_1 \cup \cdots \cup A_n$.

---

content/sets-functions-relations/size-of-sets/non-enumerability-alt.tex

4.12 Non-enumerable Sets

sfr:siz:nen-alt:sec

This section proves the non-enumerability of $\mathbb{B}^\omega$ and $\wp(\mathbb{N})$ using the definitions in section 4.11, i.e., requiring a bijection with $\mathbb{N}$ instead of a surjection from $\mathbb{Z}^+$.

explanation

The set $\mathbb{N}$ of natural numbers is infinite. It is also trivially enumerable. But the remarkable fact is that there are *non-enumerable* sets, i.e., sets which are not enumerable (see Definition 4.27).

This might be surprising. After all, to say that A is non-enumerable is to say that there is *no* bijection $f\colon \mathbb{N} \to A$; that is, no function mapping the infinitely many elements of $\mathbb{N}$ to A exhausts all of A. So if A is non-enumerable, there are "more" elements of A than there are natural numbers.

To prove that a set is non-enumerable, you have to show that no appropriate bijection can exist. The best way to do this is to show that every attempt to

enumerate elements of A must leave at least one element out; this shows that no function $f\colon \mathbb{N} \to A$ is surjective. And a general strategy for establishing this is to use Cantor's *diagonal method*. Given a list of elements of A, say, x_1, x_2, ..., we construct another element of A which, by its construction, cannot possibly be on that list.

But all of this is best understood by example. So, our first example is the set $\mathbb{B}^\omega$ of all infinite strings of 0's and 1's. (The '$\mathbb{B}$' stands for binary, and we can just think of it as the two-element set $\{0, 1\}$.)

sfr:siz:nen-alt: thm:nonenum-bin-omega

Theorem 4.31. *$\mathbb{B}^\omega$ is non-enumerable.*

Proof. Consider any enumeration of a subset of $\mathbb{B}^\omega$. So we have some list s_0, s_1, s_2, ... where every s_n is an infinite string of 0's and 1's. Let $s_n(m)$ be the nth digit of the mth string in this list. So we can now think of our list as an array, where $s_n(m)$ is placed at the nth row and mth column:

	0	1	2	3	...
0	$\mathbf{s_0(0)}$	$s_0(1)$	$s_0(2)$	$s_0(3)$	...
1	$s_1(0)$	$\mathbf{s_1(1)}$	$s_1(2)$	$s_1(3)$	...
2	$s_2(0)$	$s_2(1)$	$\mathbf{s_2(2)}$	$s_2(3)$	...
3	$s_3(0)$	$s_3(1)$	$s_3(2)$	$\mathbf{s_3(3)}$	...
⋮	⋮	⋮	⋮	⋮	⋱

We will now construct an infinite string, d, of 0's and 1's which is not on this list. We will do this by specifying each of its entries, i.e., we specify $d(n)$ for all $n \in \mathbb{N}$. Intuitively, we do this by reading down the diagonal of the array above (hence the name "diagonal method") and then changing every 1 to a 0 and every 1 to a 0. More abstractly, we define $d(n)$ to be 0 or 1 according to whether the n-th element of the diagonal, $s_n(n)$, is 1 or 0, that is:

$$d(n) = \begin{cases} 1 & \text{if } s_n(n) = 0 \\ 0 & \text{if } s_n(n) = 1 \end{cases}$$

Clearly $d \in \mathbb{B}^\omega$, since it is an infinite string of 0's and 1's. But we have constructed d so that $d(n) \neq s_n(n)$ for any $n \in \mathbb{N}$. That is, d differs from s_n in its nth entry. So $d \neq s_n$ for any $n \in \mathbb{N}$. So d cannot be on the list s_0, s_1, s_2, ...

We have shown, given an arbitrary enumeration of some subset of $\mathbb{B}^\omega$, that it will omit some element of $\mathbb{B}^\omega$. So there is no enumeration of the set $\mathbb{B}^\omega$, i.e., $\mathbb{B}^\omega$ is non-enumerable. □

explanation

This proof method is called "diagonalization" because it uses the diagonal of the array to define d. However, diagonalization need not involve the presence of an array. Indeed, we can show that some set is non-enumerable by using a similar idea, even when no array and no actual diagonal is involved. The following result illustrates how.

Theorem 4.32. $\wp(\mathbb{N})$ *is not enumerable.*

sfr:siz:nen-alt: thm:nonenum-pownat

Proof. We proceed in the same way, by showing that every list of subsets of $\mathbb{N}$ omits some subset of $\mathbb{N}$. So, suppose that we have some list $N_0, N_1, N_2, \ldots$ of subsets of $\mathbb{N}$. We define a set D as follows: $n \in D$ iff $n \notin N_n$:

$$D = \{n \in \mathbb{N} : n \notin N_n\}$$

Clearly $D \subseteq \mathbb{N}$. But D cannot be on the list. After all, by construction $n \in D$ iff $n \notin N_n$, so that $D \neq N_n$ for any $n \in \mathbb{N}$. □

explanation

The preceding proof did not mention a diagonal. Still, you can think of it as involving a diagonal if you picture it this way: Imagine the sets N_0, N_1, ..., written in an array, where we write N_n on the nth row by writing m in the mth column iff if $m \in N_n$. For example, say the first four sets on that list are $\{0, 1, 2, \ldots\}$, $\{1, 3, 5, \ldots\}$, $\{0, 1, 4\}$, and $\{2, 3, 4, \ldots\}$; then our array would begin with

$$\begin{array}{lccccccc}
N_0 = \{\mathbf{0}, & 1, & 2, & & & & \ldots\} \\
N_1 = \{ & \mathbf{1}, & & 3, & & 5, & \ldots\} \\
N_2 = \{0, & 1, & & & 4 & & \} \\
N_3 = \{ & & 2, & \mathbf{3}, & 4, & & \ldots\} \\
\vdots & & & & & & \ddots
\end{array}$$

Then D is the set obtained by going down the diagonal, placing $n \in D$ iff n is *not* on the diagonal. So in the above case, we would leave out 0 and 1, we would include 2, we would leave out 3, etc.

Problem 4.33. Show that the set of all functions $f : \mathbb{N} \to \mathbb{N}$ is non-enumerable by an explicit diagonal argument. That is, show that if f_1, f_2, ..., is a list of functions and each $f_i : \mathbb{N} \to \mathbb{N}$, then there is some $g : \mathbb{N} \to \mathbb{N}$ not on this list.

---

content/sets-functions-relations/size-of-sets/reduction-alt.tex

4.13 Reduction

sfr:siz:red-alt: sec

This section proves non-enumerability by reduction, matching the results in section 4.12. An alternative, slightly more elaborate version matching the results in section 4.6 is provided in section 4.7.

We proved that $\mathbb{B}^\omega$ is non-enumerable by a diagonalization argument. We used a similar diagonalization argument to show that $\wp(\mathbb{N})$ is non-enumerable. But here's another way we can prove that $\wp(\mathbb{N})$ is non-enumerable: show that *if*

$\wp(\mathbb{N})$ *is enumerable then* $\mathbb{B}^\omega$ *is also enumerable*. Since we know $\mathbb{B}^\omega$ is non-enumerable, it will follow that $\wp(\mathbb{N})$ is too.

This is called *reducing* one problem to another. In this case, we reduce the problem of enumerating $\mathbb{B}^\omega$ to the problem of enumerating $\wp(\mathbb{N})$. A solution to the latter—an enumeration of $\wp(\mathbb{N})$—would yield a solution to the former—an enumeration of $\mathbb{B}^\omega$.

To reduce the problem of enumerating a set B to that of enumerating a set A, we provide a way of turning an enumeration of A into an enumeration of B. The easiest way to do that is to define a surjection $f\colon A \to B$. If x_1, x_2, ... enumerates A, then $f(x_1)$, $f(x_2)$, ... would enumerate B. In our case, we are looking for a surjection $f\colon \wp(\mathbb{N}) \to \mathbb{B}^\omega$.

Problem 4.34. Show that if there is an injective function $g\colon B \to A$, and B is non-enumerable, then so is A. Do this by showing how you can use g to turn an enumeration of A into one of B.

Proof of Theorem 4.32 by reduction. For a reduction, suppose that $\wp(\mathbb{N})$ is enumerable, and thus that there is an enumeration of it, N_1, N_2, N_3, ...

Define the function $f\colon \wp(\mathbb{N}) \to \mathbb{B}^\omega$ by letting $f(N)$ be the string s_k such that $s_k(n) = 1$ iff $n \in N$, and $s_k(n) = 0$ otherwise.

This clearly defines a function, since whenever $N \subseteq \mathbb{N}$, any $n \in \mathbb{N}$ either is an element of N or isn't. For instance, the set $2\mathbb{N} = \{2n : n \in \mathbb{N}\} = \{0, 2, 4, 6, \dots\}$ of even naturals gets mapped to the string $1010101\dots$; $\emptyset$ gets mapped to $0000\dots$; $\mathbb{N}$ gets mapped to $1111\dots$.

It is also surjective: every string of 0s and 1s corresponds to some set of natural numbers, namely the one which has as its members those natural numbers corresponding to the places where the string contains a 1s. More precisely, if $s \in \mathbb{B}^\omega$, then define $N \subseteq \mathbb{N}$ by:

$$N = \{n \in \mathbb{N} : s(n) = 1\}$$

Then $f(N) = s$, as can be verified by consulting the definition of f.

Now consider the list

$$f(N_1), f(N_2), f(N_3), \dots$$

Since f is surjective, every member of $\mathbb{B}^\omega$ must appear as a value of f for some argument, and so must appear on the list. This list must therefore enumerate all of $\mathbb{B}^\omega$.

So if $\wp(\mathbb{N})$ were enumerable, $\mathbb{B}^\omega$ would be enumerable. But $\mathbb{B}^\omega$ is non-enumerable (Theorem 4.31). Hence $\wp(\mathbb{N})$ is non-enumerable. □

Problem 4.35. Show that the set X of all functions $f\colon \mathbb{N} \to \mathbb{N}$ is non-enumerable by a reduction argument (Hint: give a surjective function from X to $\mathbb{B}^\omega$.)

Problem 4.36. Show that the set of all *sets of* pairs of natural numbers, i.e., $\wp(\mathbb{N} \times \mathbb{N})$, is non-enumerable by a reduction argument.

Problem 4.37. Show that $\mathbb{N}^\omega$, the set of infinite sequences of natural numbers, is non-enumerable by a reduction argument.

Problem 4.38. Let S be the set of all surjections from $\mathbb{N}$ to the set $\{0, 1\}$, i.e., S consists of all surjections $f : \mathbb{N} \to \mathbb{B}$. Show that S is non-enumerable.

Problem 4.39. Show that the set $\mathbb{R}$ of all real numbers is non-enumerable.

Chapter 5

Arithmetization

The material in this chapter presents the construction of the number systems in naïve set theory. It is taken from Tim Button's Open Set Theory text.

---

content/sets-functions-relations/arithmetization/integers.tex

5.1 From $\mathbb{N}$ to $\mathbb{Z}$

sfr:arith:int:sec

Here are two basic realisations:

1. Every integer can be written in the form $n - m$, with $n, m \in \mathbb{N}$.
2. The information encoded in an expression $n - m$ can equally be encoded by an ordered pair $\langle n, m \rangle$.

We already know that the ordered pairs of natural numbers are the elements of $\mathbb{N}^2$. And we are assuming that we understand $\mathbb{N}$. So here is a naïve suggestion, based on the two realisations we have had: *let's treat integers as ordered pairs of natural numbers.*

In fact, this suggestion is too naïve. Obviously we want it to be the case that $0 - 2 = 4 - 6$. But evidently $\langle 0, 2 \rangle \neq \langle 4, 6 \rangle$. So we cannot simply say that $\mathbb{N}^2$ is the set of integers.

Generalising from the preceding problem, what we want is the following:

$$a - b = c - d \text{ iff } a + d = c + b$$

(It should be obvious that this is how integers are *meant* to behave: just add b and d to both sides.) And the easy way to guarantee this behaviour is just to define an equivalence relation between ordered pairs, $\sim$, as follows:

$$\langle a, b\rangle \sim \langle c, d\rangle \text{ iff } a + d = c + b$$

We now have to show that this is an equivalence relation.

Proposition 5.1. *$\sim$ is an equivalence relation.*

Proof. We must show that $\sim$ is reflexive, symmetric, and transitive.

Reflexivity: Evidently $\langle a, b\rangle \sim \langle a, b\rangle$, since $a + b = b + a$.

Symmetry: Suppose $\langle a, b\rangle \sim \langle c, d\rangle$, so $a + d = c + b$. Then $c + b = a + d$, so that $\langle c, d\rangle \sim \langle a, b\rangle$.

Transitivity: Suppose $\langle a, b\rangle \sim \langle c, d\rangle \sim \langle m, n\rangle$. So $a + d = c + b$ and $c + n = m + d$. So $a + d + c + n = c + b + m + d$, and so $a + n = m + b$. Hence $\langle a, b\rangle \sim \langle m, n\rangle$. □

Now we can use this equivalence relation to take equivalence classes:

Definition 5.2. The integers are the equivalence classes, under $\sim$, of ordered pairs of natural numbers; that is, $\mathbb{Z} = \mathbb{N}^2/_\sim$.

Now, one might have plenty of different *philosophical* reactions to this stipulative definition. Before we consider those reactions, though, it is worth continuing with some of the technicalities.

Having said what the integers are, we shall need to define basic functions and relations on them. Let's write $[m, n]_\sim$ for the equivalence class under $\sim$ with $\langle m, n\rangle$ as an element.[1] That is:

$$[m, n]_\sim = \{\langle a, b\rangle \in \mathbb{N}^2 : \langle a, b\rangle \sim \langle m, n\rangle\}$$

So now we offer some definitions:

$$\begin{aligned}
[a, b]_\sim + [c, d]_\sim &= [a + c, b + d]_\sim \\
[a, b]_\sim \times [c, d]_\sim &= [ac + bd, ad + bc]_\sim \\
[a, b]_\sim \leq [c, d]_\sim &\text{ iff } a + d \leq b + c
\end{aligned}$$

(As is common, I'm using 'ab' stand for '$(a \times b)$', just to make the axioms easier to read.) Now, we need to make sure that these definitions behave as they *ought* to. Spelling out what this means, and checking it through, is rather laborious; we relegate the details to section 5.6. But the short point is: everything works!

[1] Note: using the notation introduced in Definition 2.11, we would have written $[\langle m, n\rangle]_\sim$ for the same thing. But that's just a bit harder to read.

One final thing remains. We have constructed the integers using natural numbers. But this will mean that the natural numbers *are not themselves integers*. We will return to the philosophical significance of this in section 5.5. On a purely technical front, though, we will need some way to be able to treat natural numbers *as* integers. The idea is quite easy: for each $n \in \mathbb{N}$, we just stipulate that $n_{\mathbb{Z}} = [n, 0]_{\sim}$. We need to confirm that this definition is well-behaved, i.e., that for any $m, n \in \mathbb{N}$

$$\begin{aligned}(m + n)_{\mathbb{Z}} &= m_{\mathbb{Z}} + n_{\mathbb{Z}} \\ (m \times n)_{\mathbb{Z}} &= m_{\mathbb{Z}} \times n_{\mathbb{Z}} \\ m \leq n &\leftrightarrow m_{\mathbb{Z}} \leq n_{\mathbb{Z}}\end{aligned}$$

But this is all pretty straightforward. For example, to show that the second of these obtains, we can simply help ourselves to the behaviour of the natural numbers and reason as follows:

$$\begin{aligned}(m \times n)_{\mathbb{Z}} &= [m \times n, 0]_{\sim} \\ &= [m \times n + 0 \times 0, m \times 0 + 0 \times n]_{\sim} \\ &= [m, 0]_{\sim} \times [n, 0]_{\sim} \\ &= m_{\mathbb{Z}} \times n_{\mathbb{Z}}\end{aligned}$$

We leave it as an exercise to confirm that the other two conditions hold.

Problem 5.1. Show that $(m + n)_{\mathbb{Z}} = m_{\mathbb{Z}} + n_{\mathbb{Z}}$ and $m \leq n \leftrightarrow m_{\mathbb{Z}} \leq n_{\mathbb{Z}}$, for any $m, n \in \mathbb{N}$.

---

content/sets-functions-relations/arithmetization/rationals.tex

5.2 From $\mathbb{Z}$ to $\mathbb{Q}$

sfr:arith:rat:sec

We just saw how to construct the integers from the natural numbers, using some naïve set theory. We shall now see how to construct the rationals from the integers in a very similar way. Our initial realisations are:

1. Every rational can be written in the form i/j, where both i and j are integers but j is non-zero.

2. The information encoded in an expression i/j can equally be encoded in an ordered pair $\langle i, j\rangle$.

The obvious approach would be to think of the rationals *as* ordered pairs drawn from $\mathbb{Z} \times (\mathbb{Z} \setminus \{0_{\mathbb{Z}}\})$. As before, though, that would be a bit too naïve, since we want $3/2 = 6/4$, but $\langle 3, 2\rangle \neq \langle 6, 4\rangle$. More generally, we will want the following:

$$a/b = c/d \text{ iff } a \times d = b \times c$$

To get this, we define an equivalence relation on $\mathbb{Z} \times (\mathbb{Z} \setminus \{0_\mathbb{Z}\})$ thus:

$$\langle a, b\rangle \backsim \langle c, d\rangle \text{ iff } a \times d = b \times c$$

We must check that this is an equivalence relation. This is very much like the case of $\sim$, and we will leave it as an exercise.

Problem 5.2. Show that $\backsim$ is an equivalence relation.

But it allows us to say:

Definition 5.3. The rationals are the equivalence classes, under $\backsim$, of pairs of integers (whose second element is non-zero). That is, $\mathbb{Q} = (\mathbb{Z} \times (\mathbb{Z} \setminus \{0_\mathbb{Z}\}))/\backsim$.

As with the integers, we also want to define some basic operations. Where $[i, j]_\backsim$ is the equivalence class under $\backsim$ with $\langle i, j\rangle$ as an element, we say:

$$[a, b]_\backsim + [c, d]_\backsim = [ad + bc, bd]_\backsim$$
$$[a, b]_\backsim \times [c, d]_\backsim = [ac, bd]_\backsim.$$

To define $r \leq s$ on these rationals, we use the fact that $r \leq s$ iff $s - r$ is not negative, i.e., $r - s$ can be written as i/j with i non-negative and j positive:

$$[a, b]_\backsim \leq [c, d]_\backsim \text{ iff } [c, d]_\backsim - [a, b]_\backsim = [i_\mathbb{Z}, j_\mathbb{Z}]_\backsim$$

for some $i \in \mathbb{N}$ and $0 \neq j \in \mathbb{N}$.

We then need to check that these definitions behave as they *ought* to; and we relegate this to section 5.6. But they indeed do! Finally, we want some way to treat integers *as* rationals; so for each $i \in \mathbb{Z}$, we stipulate that $i_\mathbb{Q} = [i, 1_\mathbb{Z}]_\backsim$. Again, we check that all of this behaves correctly in section 5.6.

Problem 5.3. Show that $(i + j)_\mathbb{Q} = i_\mathbb{Q} + j_\mathbb{Q}$ and $(i \times j)_\mathbb{Q} = i_\mathbb{Q} \times j_\mathbb{Q}$ and $i \leq j \leftrightarrow i_\mathbb{Q} \leq j_\mathbb{Q}$, for any $i, j \in \mathbb{Z}$.

---

content/sets-functions-relations/arithmetization/reals.tex

5.3 The Real Line

sfr:arith:real:sec

The next step is to show how to construct the reals from the rationals. Before that, we need to understand what is *distinctive* about the reals.

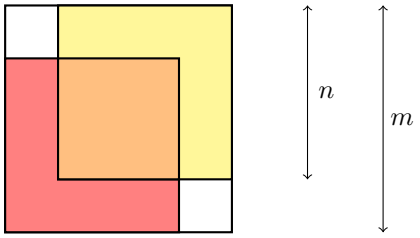
The reals behave very much like the rationals. (Technically, both are examples of *ordered fields*; for the definition of this, see Definition 5.9.) Now, if you worked through the exercises to chapter 4, you will know that there are strictly more reals than rationals, i.e., that $\mathbb{Q} \prec \mathbb{R}$. This was first proved by Cantor. But it's been known for about two and a half millennia that there are irrational numbers, i.e., reals which are not rational. Indeed:

sfr:arith:real:root2irrational

Theorem 5.4. *$\sqrt{2}$ is not rational, i.e., $\sqrt{2} \notin \mathbb{Q}$*

Proof. Suppose, for reductio, that $\sqrt{2}$ is rational. So $\sqrt{2} = {}^{m}/_{n}$ for some natural numbers m and n. Indeed, we can choose m and n so that the fraction cannot be reduced any further. Re-organising, $m^2 = 2n^2$. From here, we can complete the proof in two ways:

First, geometrically (following Tennenbaum).[2] Consider these squares:



□

Since $m^2 = 2n^2$, the region where the two squares of side n overlap has the same area as the region which neither of the two squares cover; i.e., the area of the orange square equals the sum of the area of the two unshaded squares. So where the orange square has side p, and each unshaded square has side q, $p^2 = 2q^2$. But now $\sqrt{2} = {}^{p}/_{q}$, with $p < m$ and $q < n$ and $p, q \in \mathbb{N}$. This contradicts the fact that m and n were chosen to be as small as possible.

Second, formally. Since $m^2 = 2n^2$, it follows that m is even. (It is easy to show that, if x is odd, then x^2 is odd.) So $m = 2r$, for some $r \in \mathbb{N}$. Rearranging, $2r^2 = n^2$, so n is also even. So both m and n are even, and hence the fraction ${}^{m}/_{n}$ *can* be reduced further. Contradiction!

In passing, this diagrammatic proof allows us to revisit the material from section 76.4. Tennenbaum (1927–2006) was a thoroughly modern mathematician; but the proof is undeniably lovely, completely rigorous, and appeals to geometric intuition!

In any case: the reals are "more expansive" than the rationals. In some sense, there are "gaps" in the rationals, and these are filled by the reals. Weierstrass realised that this describes a single property of the real numbers, which distinguishes them from the rationals, namely the Completeness Property: *Every non-empty set of real numbers with an upper bound has a least upper bound.*

It is easy to see that the rationals do not have the Completeness Property. For example, consider the set of rationals less than $\sqrt{2}$, i.e.:

$$\{p \in \mathbb{Q} : p^2 < 2 \text{ or } p < 0\}$$

This has an upper bound in the rationals; its elements are all smaller than 3, for example. But what is its least upper bound? We want to say '$\sqrt{2}$'; but we have just seen that $\sqrt{2}$ is *not* rational. And there is no *least* rational number greater than $\sqrt{2}$. So the set has an upper bound but no least upper bound. Hence the rationals lack the Completeness Property.

[2] This proof is reported by Conway (2006).

By contrast, the continuum "morally ought" to have the Completeness Property. We do not just want $\sqrt{2}$ to be a real number; we want to fill all the "gaps" in the rational line. Indeed, we want the continuum itself to have no "gaps" in it. That is just what we will get via Completeness.

---

content/sets-functions-relations/arithmetization/cuts.tex

5.4 From $\mathbb{Q}$ to $\mathbb{R}$

sfr:arith:cuts:sec

In essence, the Completeness Property shows that any point α of the real line divides that line into two halves perfectly: those for which α is the least upper bound, and those for which α is the greatest lower bound. To *construct* the real numbers from the rational numbers, Dedekind suggested that we simply think of the reals as the *cuts* that partition the rationals. That is, we identify $\sqrt{2}$ with the *cut* which separates the rationals $< \sqrt{2}$ from the rationals $> \sqrt{2}$.

Let's tidy this up. If we cut the rational numbers into two halves, we can uniquely identify the partition we made just by considering its *bottom* half. So, getting precise, we offer the following definition:

Definition 5.5 (Cut). A *cut* α is any non-empty proper initial segment of the rationals with no greatest element. That is, α is a cut iff:

1. *non-empty, proper*: $\emptyset \neq \alpha \subsetneq \mathbb{Q}$
2. *initial*: for all $p, q \in \mathbb{Q}$: if $p < q \in \alpha$ then $p \in \alpha$
3. *no maximum*: for all $p \in \alpha$ there is a $q \in \alpha$ such that $p < q$

Then $\mathbb{R}$ is the set of cuts.

So now we can say that $\sqrt{2} = \{p \in \mathbb{Q} : p^2 < 2 \text{ or } p < 0\}$. Of course, we need to check that this *is* a cut, but we relegate that to section 5.6.

As before, having defined some entities, we next need to define basic functions and relations upon them. We begin with an easy one:

$$\alpha \leq \beta \text{ iff } \alpha \subseteq \beta$$

This definition of an order allows to *state* the central result, that the set of cuts has the Completeness Property. Spelled out fully, the statement has this shape. If S is a non-empty set of cuts with an upper bound, then S has a least upper bound. In more detail: there is a cut, λ, which is an upper bound for S, i.e. $(\forall \alpha \in S)\alpha \subseteq \lambda$, and λ is the least such cut, i.e. $(\forall \beta \in \mathbb{R})((\forall \alpha \in S)\alpha \subseteq \beta \to \lambda \subseteq \beta)$. Now here is the proof of the result:

sfr:arith:cuts:realcompleteness

Theorem 5.6. *The set of cuts has the Completeness Property.*

Proof. Let S be any non-empty set of cuts with an upper bound. Let $\lambda = \bigcup S$. We first claim that λ is a cut:

1. Since S has an upper bound, at least one cut is in S, so $\emptyset \neq \lambda$. Since S is a set of cuts, $\lambda \subseteq \mathbb{Q}$. Since S has an upper bound, some $p \in \mathbb{Q}$ is absent from every cut $\alpha \in S$. So $p \notin \lambda$, and hence $\lambda \subsetneq \mathbb{Q}$.

2. Suppose $p < q \in \lambda$. So there is some $\alpha \in S$ such that $q \in \alpha$. Since α is a cut, $p \in \alpha$. So $p \in \lambda$.

3. Suppose $p \in \lambda$. So there is some $\alpha \in S$ such that $p \in \alpha$. Since α is a cut, there is some $q \in \alpha$ such that $p < q$. So $q \in \lambda$.

This proves the claim. Moreover, clearly $(\forall \alpha \in S)\alpha \subseteq \bigcup S = \lambda$, i.e. λ is an upper bound on S. So now suppose $\beta \in \mathbb{R}$ is also an upper bound, i.e. $(\forall \alpha \in S)\alpha \subseteq \beta$. For any $p \in \mathbb{Q}$, if $p \in \lambda$, then there is $\alpha \in S$ such that $p \in \alpha$, so that $p \in \beta$. Generalizing, $\lambda \subseteq \beta$. So λ is the *least* upper bound on S. □

So we have a bunch of entities which satisfy the Completeness Property. And one way to put this is: there are no "gaps" in our cuts. (So: taking further "cuts" of reals, rather than rationals, would yield no interesting new objects.)

Next, we must define some operations on the reals. We start by embedding the rationals into the reals by stipulating that $p_{\mathbb{R}} = \{q \in \mathbb{Q} : q < p\}$ for each $p \in \mathbb{Q}$. We then define:

$$\begin{aligned} \alpha + \beta &= \{p + q : p \in \alpha \wedge q \in \beta\} \\ \alpha \times \beta &= \{p \times q : 0 \leq p \in \alpha \wedge 0 \leq q \in \beta\} \cup 0^{\mathbb{R}} \qquad \text{if } \alpha, \beta \geq 0_{\mathbb{R}} \end{aligned}$$

To handle the other multiplication cases, first let:

$$-\alpha = \{p - q : p < 0 \wedge q \notin \alpha\}$$

and then stipulate:

$$\alpha \times \beta = \begin{cases} -\alpha \times -\beta & \text{if } \alpha < 0_{\mathbb{R}} \text{ and } \beta < 0_{\mathbb{R}} \\ -(-\alpha \times \beta) & \text{if } \alpha < 0_{\mathbb{R}} \text{ and } \beta > 0_{\mathbb{R}} \\ -(\alpha \times -\beta) & \text{if } \alpha > 0_{\mathbb{R}} \text{ and } \beta < 0_{\mathbb{R}} \end{cases}$$

We then need to check that each of these definitions always yields a cut. And finally, we need to go through an easy (but long-winded) demonstration that the cuts, so defined, behave exactly as they should. But we relegate all of this to section 5.6.

---

content/sets-functions-relations/arithmetization/reflections.tex

5.5 Some Philosophical Reflections

So much for the technicalities. But what did they achieve? sfr:arith:ref:sec

Well, pretty uncontestably, they gave us some lovely pure mathematics. Moreover, there were some deep conceptual achievements. It was a profound insight, to see that the Completeness Property expresses the crucial difference between the reals and the rationals. Moreover, the explicit construction of reals, as Dedekind cuts, puts the subject matter of analysis on a firm footing. We know that the notion of a *complete ordered field* is coherent, for the cuts form just such a field.

For all that, we should air a few reservations about these achievements.

First, it is not clear that thinking of reals in terms of cuts is any *more* rigorous than thinking of reals in terms of their familiar (possibly infinite) decimal expansions. This latter "construction" of the reals has some resemblance to the construction of the reals via Cauchy sequence; but in fact, it was essentially known to mathematicians from the early 17th century onwards (see section 5.7). The real increase in rigour came from the realisation that the reals have the Completeness Property; the ability to construct real numbers as particular sets is perhaps not, by itself, so very interesting.

It is even less clear that the (much easier) arithmetization of the integers, or of the rationals, increases rigour in those areas. Here, it is worth making a simple observation. Having *constructed* the integers as equivalence classes of ordered pairs of naturals, and then constructed the rationals as equivalence classes of ordered pairs of integers, and then constructed the reals as sets of rationals, we immediately *forget about* the constructions. In particular: no one would ever want to *invoke* these constructions during a mathematical proof (excepting, of course, a proof that the constructions behaved as they were supposed to). It's much easier to speak about a real, directly, than to speak about some set of sets of sets of sets of sets of sets of sets of naturals.

It is most doubtful of all that these definitions tell us what the integers, rationals, or reals *are*, *metaphysically speaking*. That is, it is doubtful that the reals (say) *are* certain sets (of sets of sets...). The main barrier to such a view is that the construction could have been done in many different ways. In the case of the reals, there are some genuinely interestingly different constructions (see section 5.7). But here is a really trivial way to obtain some different constructions: as in section 2.2, we could have defined ordered pairs slightly differently; if we had used this alternative notion of an ordered pair, then our constructions would have worked precisely as well as they did, but we would have ended up with different objects. As such, there are many rival set-theoretic constructions of the integers, the rationals, and the reals. And now it would just be arbitrary (and embarrassing) to claim that the integers (say) are *these* sets, rather than *those*. (As in section 2.2, this is an instance of an argument made famous by Benacerraf 1965.)

A further point is worth raising: there is something quite *odd* about our constructions. We started with the natural numbers. We then construct the integers, and construct "the 0 of the integers", i.e., $[0,0]_\sim$. But $0 \neq [0,0]_\sim$. Indeed, given our constructions, *no* natural number is an integer. But that seems extremely counter-intuitive. Indeed, in section 1.3, we claimed without much argument that $\mathbb{N} \subseteq \mathbb{Q}$. If the constructions tell us exactly *what* the

numbers are, this claim was trivially false.

Standing back, then, where do we get to? Working in a naïve set theory, and helping ourselves to the naturals, we are able to *treat* integers, rationals, and reals as certain sets. In that sense, we can *embed* the theories of these entities within a set theory. But the philosophical import of this embedding is just not that straightforward.

Of course, none of this is the last word! The point is only this. Showing that the arithmetization of the reals *is* of deep philosophical significance would require some additional *philosophical* argument.

---

content/sets-functions-relations/arithmetization/checking-details.tex

5.6 Ordered Rings and Fields

sfr:arith:check:sec

Throughout this chapter, we claimed that certain definitions behave "as they ought". In this technical appendix, we will spell out what we mean, and (sketch how to) show that the definitions do behave "correctly".

In section 5.1, we defined addition and multiplication on $\mathbb{Z}$. We want to show that, as defined, they endow $\mathbb{Z}$ with the structure we "would want" it to have. In particular, the structure in question is that of a commutative ring.

Definition 5.7. A *commutative ring* is a set S, equipped with specific elements 0 and 1 and operations $+$ and $\times$, satisfying these eight formulas:

Associativity	$a + (b + c) = (a + b) + c$
	$(a \times b) \times c = a \times (b \times c)$
Commutativity	$a + b = b + a$
	$a \times b = b \times a$
Identities	$a + 0 = a$
	$a \times 1 = a$
Additive Inverse	$(\exists b \in S) 0 = a + b$
Distributivity	$a \times (b + c) = (a \times b) + (a \times c)$

Implicitly, these are all bound with universal quantifiers restricted to S. And note that the elements 0 and 1 here need not be the natural numbers with the same name.

So, to check that the integers form a commutative ring, we just need to check that we meet these eight conditions. None of the conditions is difficult to establish, but this is a bit laborious. For example, here is how to prove *Associativity*, in the case of addition:

Proof. Fix $i, j, k \in \mathbb{Z}$. So there are $a_1, b_1, a_2, b_2, a_3, b_3 \in \mathbb{N}$ such that $i = [a_1, b_1]$ and $j = [a_2, b_2]$ and $k = [a_3, b_3]$. (For legibility, we write "$[x, y]$" rather than

"$[x, y]_\sim$"; we'll do this throughout this section.) Now:

$$\begin{aligned} i + (j + k) &= [a_1, b_1] + ([a_2, b_2] + [a_3, b_3]) \\ &= [a_1, b_1] + [a_2 + a_3, b_2 + b_3] \\ &= [a_1 + (a_2 + a_3), b_1 + (b_2 + b_3)] \\ &= [(a_1 + a_2) + a_3, (b_1 + b_2) + b_3] \\ &= [a_1 + a_2, b_1 + b_2] + [a_3, b_3] \\ &= ([a_1, b_1] + [a_2, b_2]) + [a_3, b_3] \\ &= (i + j) + k \end{aligned}$$

helping ourselves freely to the behavior of addition on $\mathbb{N}$. □

Equally, here is how to prove *Additive Inverse*:

Proof. Fix $i \in \mathbb{Z}$, so that $i = [a, b]$ for some $a, b \in \mathbb{N}$. Let $j = [b, a] \in \mathbb{Z}$. Helping ourselves to the behaviour of the naturals, $(a + b) + 0 = 0 + (a + b)$, so that $\langle a + b, b + a\rangle \sim_{\mathbb{Z}} \langle 0, 0\rangle$ by definition, and hence $[a + b, b + a] = [0, 0] = 0_{\mathbb{Z}}$. So now $i + j = [a, b] + [b, a] = [a + b, b + a] = [0, 0] = 0_{\mathbb{Z}}$. □

And here is a proof of *Distributivity*:

Proof. As above, fix $i = [a_1, b_1]$ and $j = [a_2, b_2]$ and $k = [a_3, b_3]$. Now:

$$\begin{aligned} i \times (j + k) &= [a_1, b_1] \times ([a_2, b_2] + [a_3, b_3]) \\ &= [a_1, b_1] \times [a_2 + a_3, b_2 + b_3] \\ &= [a_1(a_2 + a_3) + b_1(b_2 + b_3), a_1(b_2 + b_3) + b_1(a_2 + a_3)] \\ &= [a_1a_2 + a_1a_3 + b_1b_2 + b_1b_3, a_1b_2 + a_1b_3 + a_2b_1 + a_3b_1] \\ &= [a_1a_2 + b_1b_2, a_1b_2 + a_2b_1] + [a_1a_3 + b_1b_3, a_1b_3 + a_3b_1] \\ &= ([a_1, b_1] \times [a_2, b_2]) + ([a_1, b_1] \times [a_3, b_3]) \\ &= (i \times j) + (i \times k) \end{aligned}$$

□

We leave it as an exercise to prove the remaining five conditions. Having done that, we have shown that $\mathbb{Z}$ constitutes a commutative ring, i.e., that addition and multiplication (as defined) behave as they should.

Problem 5.4. Prove that $\mathbb{Z}$ is a commutative ring.

But our task is not over. As well as defining addition and multiplication over $\mathbb{Z}$, we defined an ordering relation, $\leq$, and we must check that this behaves as it should. In more detail, we must show that $\mathbb{Z}$ constitutes an *ordered* ring.[3]

[3]Recall from Definition 2.16 that a total order is a relation which is reflexive, transitive, anti-symmetric, and connected. In the context of order relations, connectedness is sometimes called *trichotomy*, since for any a and b we have $a \leq b \vee a = b \vee a \geq b$.

Definition 5.8. An *ordered ring* is a commutative ring which is also equipped with a total order relation, $\leq$, such that:

$$a \leq b \to a + c \leq b + c$$
$$(a \leq b \wedge 0 \leq c) \to a \times c \leq b \times c$$

Problem 5.5. Prove that $\mathbb{Z}$ is an ordered ring.

As before, it is laborious but routine to show that $\mathbb{Z}$, as constructed, is an ordered ring. We will leave that to you.

This takes care of the integers. But now we need to show very similar things of the rationals. In particular, we now need to show that the rationals form an ordered *field*, under our given definitions of $+$, $\times$, and $\leq$:

Definition 5.9. An *ordered field* is an ordered ring which also satisfies: sfr:arith:check: orderedfield

$$\textit{Multiplicative Inverse} \qquad (\forall a \in S \setminus \{0\})(\exists b \in S) a \times b = 1$$

Once you have shown that $\mathbb{Z}$ constitutes an ordered ring, it is easy but laborious to show that $\mathbb{Q}$ constitutes an ordered field.

Problem 5.6. Prove that $\mathbb{Q}$ is an ordered field.

Having dealt with the integers and the rationals, it only remains to deal with the reals. In particular, we need to show that $\mathbb{R}$ constitutes a *complete* ordered field, i.e., an ordered field with the Completeness Property. Now, Theorem 5.6 established that $\mathbb{R}$ has the Completeness Property. However, it remains to run through the (tedious) of checking that $\mathbb{R}$ is an ordered field.

Before tearing off into *that* laborious exercise, we need to check some more "immediate" things. For example, we need a guarantee that $\alpha + \beta$, as defined, is indeed a *cut*, for any cuts α and β. Here is a proof of that fact:

Proof. Since α and β are both cuts, $\alpha + \beta = \{p + q : p \in \alpha \wedge q \in \beta\}$ is a non-empty proper subset of $\mathbb{Q}$. Now suppose $x < p + q$ for some $p \in \alpha$ and $q \in \beta$. Then $x - p < q$, so $x - p \in \beta$, and $x = p + (x - p) \in \alpha + \beta$. So $\alpha + \beta$ is an initial segment of $\mathbb{Q}$. Finally, for any $p + q \in \alpha + \beta$, since α and β are both cuts, there are $p_1 \in \alpha$ and $q_1 \in \beta$ such that $p < p_1$ and $q < q_1$; so $p + q < p_1 + q_1 \in \alpha + \beta$; so $\alpha + \beta$ has no maximum. □

Similar efforts will allow you to check that $\alpha - \beta$ and $\alpha \times \beta$ and $\alpha \div \beta$ are cuts (in the last case, ignoring the case where β is the zero-cut). Again, though, we will simply leave this to you.

Problem 5.7. Prove that $\mathbb{R}$ is an ordered field.

But here is a small loose end to tidy up. In section 5.4, we suggest that we can take $\sqrt{2} = \{p \in \mathbb{Q} : p < 0 \text{ or } p^2 < 2\}$. But we do need to show that this set is a *cut*. Here is a proof of that fact:

Proof. Clearly this is a nonempty proper initial segment of the rationals; so it suffices to show that it has no maximum. In particular, it suffices to show that, where p is a positive rational with $p^2 < 2$ and $q = \frac{2p+2}{p+2}$, both $p < q$ and $q^2 < 2$. To see that $p < q$, just note:

$$\begin{aligned} p^2 &< 2 \\ p^2 + 2p &< 2 + 2p \\ p(p+2) &< 2 + 2p \\ p &< \tfrac{2+2p}{p+2} = q \end{aligned}$$

To see that $q^2 < 2$, just note:

$$\begin{aligned} p^2 &< 2 \\ 2p^2 + 4p + 2 &< p^2 + 4p + 4 \\ 4p^2 + 8p + 4 &< 2(p^2 + 4p + 4) \\ (2p+2)^2 &< 2(p+2)^2 \\ \tfrac{(2p+2)^2}{(p+2)^2} &< 2 \\ q^2 &< 2 \end{aligned}$$

□

---

content/sets-functions-relations/arithmetization/cauchy.tex

5.7 Appendix: the Reals as Cauchy Sequences

sfr:arith:cauchy:sec

In section 5.4, we constructed the reals as Dedekind cuts. In this section, we explain an alternative construction. It builds on Cauchy's definition of (what we now call) a Cauchy sequence; but the use of this definition to *construct* the reals is due to other nineteenth-century authors, notably Weierstrass, Heine, Méray and Cantor. (For a nice history, see O'Connor and Robertson 2005.)

Before we get to the nineteenth century, it's worth considering Simon Stevin (1548–1620). In brief, Stevin realised that we can think of each real in terms of its decimal expansion. Thus even an irrational number, like $\sqrt{2}$, has a nice decimal expansion, beginning:

$$1.41421356237\ldots$$

It is very easy to model decimal expansions in set theory: simply consider them as functions $d\colon \mathbb{N} \to \mathbb{N}$, where $d(n)$ is the nth decimal place that we are interested in. We will then need a bit of tweak, to handle the bit of the real number that comes before the decimal point (here, just 1). We will also need a further tweak (an equivalence relation) to guarantee that, for example, $0.999\ldots = 1$. But it is not difficult to offer a perfectly rigorous construction of the real numbers, in the manner of Stevin, within set theory.

Stevin is not our focus. (For more on Stevin, see Katz and Katz 2012.) But here is a closely related thought. Instead of treating $\sqrt{2}$'s decimal expansion directly, we can instead consider a *sequence* of increasingly accurate rational approximations to $\sqrt{2}$, by considering the increasingly precise expansions:

$$1, 1.4, 1.414, 1.4142, 1.41421, \ldots$$

The idea that reals can be considered via "increasingly good approximations" provides us with the basis for another sequence of insights (akin to the realisations that we used when constructing $\mathbb{Q}$ from $\mathbb{Z}$, or $\mathbb{Z}$ from $\mathbb{N}$). The basic insights are these:

1. Every real can be written as a (perhaps infinite) decimal expansion.
2. The information encoded by a (perhaps infinite) decimal expansion can be equally be encoded by a sequence of rational numbers.
3. A sequence of rational numbers can be thought of as a function from $\mathbb{N}$ to $\mathbb{Q}$; just let $f(n)$ be the nth rational in the sequence.

Of course, not just *any* function from $\mathbb{N}$ to $\mathbb{Q}$ will give us a real number. For instance, consider this function:

$$f(n) = \begin{cases} 1 & \text{if } n \text{ is odd} \\ 0 & \text{if } n \text{ is even} \end{cases}$$

Essentially the worry here is that the sequence $0, 1, 0, 1, 0, 1, 0, \ldots$ doesn't seem to "hone in" on any real. So: to ensure that we consider sequences which do hone in on some real, we need to restrict our attention to sequences which have some *limit*.

We have already encountered the idea of a limit, in section 76.2. But we cannot use *quite* the same definition as we used there. The expression "$(\forall \varepsilon > 0)$" there tacitly involved quantification over the real numbers; and we were considering the limits of functions on the real numbers; so invoking that definition would be to help ourselves to the real numbers; and they are exactly what we were aiming to *construct*. Fortunately, we can work with a closely related idea of a limit.

sfr:arith:cauchy:def:CauchySequence

Definition 5.10. A function $f : \mathbb{N} \to \mathbb{Q}$ is a *Cauchy sequence* iff for any positive $\varepsilon \in \mathbb{Q}$ we have that $(\exists \ell \in \mathbb{N})(\forall m, n > \ell)|f(m) - f(n)| < \varepsilon$.

The general idea of a limit is the same as before: if you want a certain level of precision (measured by ε), there is a "region" to look in (any input greater than ℓ). And it is easy to see that our sequence 1, 1.4, 1.414, 1.4142, 1.41421... has a limit: if you want to approximate $\sqrt{2}$ to within an error of $1/10^n$, then just look to any entry after the nth.

The obvious thought, then, would be to say that a real number just *is* any Cauchy sequence. But, as in the constructions of $\mathbb{Z}$ and $\mathbb{Q}$, this would

be too naïve: for any given real number, multiple different Cauchy sequences indicate that real number. A simple way to see this as follows. Given a Cauchy sequence f, define g to be exactly the same function as f, except that $g(0) \neq f(0)$. Since the two sequences agree everywhere after the first number, we will (ultimately) want to say that they have the same limit, in the sense employed in Definition 5.10, and so should be thought of "defining" the same real. So, we should really think of these Cauchy sequences as the same real number.

Consequently, we again need to define an equivalence relation on the Cauchy sequences, and identify real numbers with equivalence relations. First we need the idea of a function which tends to 0 in the limit. For any function $h : \mathbb{N} \to \mathbb{Q}$, say that h *tends to* 0 iff for any positive $\varepsilon \in \mathbb{Q}$ we have that $(\exists \ell \in \mathbb{N})(\forall n > \ell)|f(n)| < \varepsilon$.[4] Further, where f and g are functions $\mathbb{N} \to \mathbb{Q}$, let $(f - g)(n) = f(n) - g(n)$. Now define:

$$f \eqsim g \text{ iff } (f - g) \text{ tends to } 0.$$

We need to check that $\eqsim$ is an equivalence relation; and it is. We can then, if we like, define the reals as the equivalence classes, under $\eqsim$, of all Cauchy sequences from $\mathbb{N} \to \mathbb{Q}$.

Problem 5.8. Let $f(n) = 0$ for every n. Let $g(n) = \frac{1}{(n+1)^2}$. Show that both are Cauchy sequences, and indeed that the limit of both functions is 0, so that also $f \sim_{\mathbb{R}} g$.

Having done this, we shall as usual write $[f]_{\eqsim}$ for the equivalence class with f as an element. However, to keep things readable, in what follows we will drop the subscript and write just $[f]$. We also stipulate that, for each $q \in \mathbb{Q}$, we have $q_{\mathbb{R}} = [c_q]$, where c_q is the constant function $c_q(n) = q$ for all $n \in \mathbb{N}$. We then define basic relations and operations on the reals, e.g.:

$$[f] + [g] = [(f + g)]$$
$$[f] \times [g] = [(f \times g)]$$

where $(f+g)(n) = f(n)+g(n)$ and $(f \times g)(n) = f(n) \times g(n)$. Of course, we also need to check that each of $(f + g)$, $(f - g)$ and $(f \times g)$ are Cauchy sequences when f and g are; but they are, and we leave this to you.

Finally, we define we a notion of order. Say $[f]$ is *positive* iff both $[f] \neq 0_{\mathbb{Q}}$ and $(\exists \ell \in \mathbb{N})(\forall n > \ell) 0 < f(n)$. Then say $[f] < [g]$ iff $[(g - f)]$ is positive. We have to check that this is well-defined (i.e., that it does not depend upon choice of "representative" function from the equivalence class). But having done this, it is quite easy to show that these yield the right algebraic properties; that is:

sfr:arith:cauchy: thm:cauchyorderedfield

Theorem 5.11. *The Cauchy sequences constitute an ordered field.*

Proof. Exercise. □

[4] Compare this with the definition of $\lim_{x\to\infty} f(x) = 0$ in section 76.2.

Problem 5.9. Prove that the Cauchy sequences constitute an ordered field.

It is harder to prove that the reals, so constructed, have the Completeness Property, so we will give the proof.

Theorem 5.12. *Every non-empty set of Cauchy sequences with an upper bound has a least upper bound.*

Proof sketch. Let S be any non-empty set of Cauchy sequences with an upper bound. So there is some $p \in \mathbb{Q}$ such that $p_{\mathbb{R}}$ is an upper bound for S. Let $r \in S$; then there is some $q \in \mathbb{Q}$ such that $q_{\mathbb{R}} < r$. So if a least upper bound on S exists, it is between $q_{\mathbb{R}}$ and $p_{\mathbb{R}}$ (inclusive).

We will hone in on the l.u.b., by approaching it simultaneously from below and above. In particular, we define two functions, $f, g\colon \mathbb{N} \to \mathbb{Q}$, with the aim that f will hone in on the l.u.b. from above, and g will hone on in it from below. We start by defining:

$$f(0) = p$$
$$g(0) = q$$

Then, where $a_n = \frac{f(n)+g(n)}{2}$, let:[5]

$$f(n+1) = \begin{cases} a_n & \text{if } (\forall h \in S)[h] \leq (a_n)_{\mathbb{R}} \\ f(n) & \text{otherwise} \end{cases}$$

$$g(n+1) = \begin{cases} a_n & \text{if } (\exists h \in S)[h] \geq (a_n)_{\mathbb{R}} \\ g(n) & \text{otherwise} \end{cases}$$

Both f and g are Cauchy sequences. (This can be checked fairly easily, but we leave it as an exercise.) Note that the function $(f - g)$ tends to 0, since the difference between f and g halves at each step. Hence $[f] = [g]$.

We first show that $[f]$ is an upper bound on S, i.e. that $(\forall h \in S)[h] \leq [f]$. (We will invoke Theorem 5.11 as we go.) Let $h \in S$ and suppose, for reductio, that $[f] < [h]$, so that $0_{\mathbb{R}} < [(h - f)]$. Since f is a monotonically decreasing Cauchy sequence, there is some $n \in \mathbb{N}$ such that $[(c_{f(n)} - f)] < [(h - f)]$. So:

$$(f(n))_{\mathbb{R}} = [c_{f(n)}] < [f] + [(h - f)] = [h],$$

contradicting the fact that, by construction, $[h] \leq (f(n))_{\mathbb{R}}$.

We next show that $[f] = [g]$ is the *least* upper bound on S. So let j be any Cauchy sequence and suppose $[j] < [g]$. Reasoning as above (using the fact that g is *increasing*), there is $n \in \mathbb{N}$ such that $[j] < (g(n))_{\mathbb{R}}$. But by construction there is $h \in S$ such that $(g(n))_{\mathbb{R}} \leq [h]$, so $[j] < [h]$ and therefore $[j]$ is not an upper bound on S. □

[5] This is a recursive definition. But we have not *yet* given any reason to think that recursive definitions are ok.

Chapter 6

Infinite Sets

This chapter on infinite sets is taken from Tim Button's *Open Set Theory*.

---

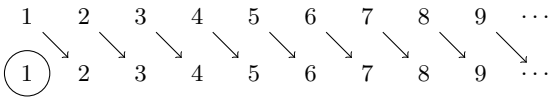
content/sets-functions-relations/infinite/hilberts-hotel.tex

6.1 Hilbert's Hotel

sfr:infinite:hilbert:sec

The set of the natural numbers is obviously infinite. So, if we do not want to *help ourselves* to the natural numbers, our first step must be characterize an infinite set in terms that do not require mentioning the natural numbers themselves. Here is a nice approach, presented by Hilbert in a lecture from 1924. He asks us to imagine

> [...] a hotel with a finite number of rooms. All of these rooms should be occupied by exactly one guest. If the guests now swap their rooms somehow, [but] so that each room still contains no more than one person, then no rooms will become free, and the hotel-owner cannot in this way create a new place for a newly arriving guest [...¶...]
>
> Now we stipulate that the hotel shall have infinitely many numbered rooms 1, 2, 3, 4, 5, ..., each of which is occupied by exactly one guest. As soon as a new guest comes along, the owner only needs to move each of the old guests into the room associated with the number one higher, and room 1 will be free for the newly-arriving guest.



(published in Hilbert 2013, 730; our translation)

The crucial point is that Hilbert's Hotel has infinitely many rooms; and we can take his explanation to define what it means to say this. Indeed, this was Dedekind's approach (presented here, of course, with massive anachronism; Dedekind's definition is from 1888):

Definition 6.1. A set A is *Dedekind infinite* iff there is an injection from A to a proper subset of A. That is, there is some $o \in A$ and an injection $f : A \to A$ such that $o \notin \mathrm{ran}(f)$. sfr:infinite:hilbert:defn:DedekindInfinite

---

content/sets-functions-relations/infinite/dedekind-algebra.tex

6.2 Dedekind Algebras

We not only want natural numbers to be infinite; we want them to have certain (algebraic) properties: they need to behave well under addition, multiplication, and so forth. sfr:infinite:dedekind:sec

Dedekind's idea was to take the idea of the *successor function* as basic, and then characterise the numbers as those with the following properties:

1. There is a number, 0, which is not the successor of any number
   i.e., $0 \notin \mathrm{ran}(s)$
   i.e., $\forall x\ s(x) \neq 0$

2. Distinct numbers have distinct successors
   i.e., s is an injection
   i.e., $\forall x \forall y (s(x) = s(y) \to x = y)$

3. Every number is obtained from 0 by repeated applications of the successor function. sfr:infinite:dedekind:repeatedapplication

The first two conditions are easy to deal with using first-order logic (see above). But we cannot deal with (3) just using first-order logic. Dedekind's breakthrough was to reformulate condition (3), set-theoretically, as follows:

3′. The natural numbers are the smallest set that is *closed under the successor function*: that is, if we apply s to any element of the set, we obtain another element of the set.

But we shall need to spell this out slowly.

Definition 6.2. For any function f, the set X is *f-closed* iff $(\forall x \in X) f(x) \in X$. Now define, for any o: sfr:infinite:dedekind:Closure

$$\mathrm{clo}_f(o) = \bigcap \{X : o \in X \text{ and } X \text{ is } f\text{-closed}\}$$

So $\mathrm{clo}_f(o)$ is the intersection of all the f-closed sets with o as an element. Intuitively, then, $\mathrm{clo}_f(o)$ is the *smallest* f-closed set with o as an element. This next result makes that intuitive thought precise;

sfr:infinite:dedekind: closureproperties

Lemma 6.3. *For any function f and any $o \in A$:*

sfr:infinite:dedekind: closurehaselem

1. $o \in \mathrm{clo}_f(o)$*; and*

sfr:infinite:dedekind: closureclosed

2. $\mathrm{clo}_f(o)$ *is f-closed; and*

sfr:infinite:dedekind: closuresmallest

3. *if X is f-closed and $o \in X$, then* $\mathrm{clo}_f(o) \subseteq X$

Proof. Note that there is at least one f-closed set with o as an element, namely $\mathrm{ran}(f) \cup \{o\}$. So $\mathrm{clo}_f(o)$, the intersection of *all* such sets, exists. We must now check (1)–(3).

Concerning (1): $o \in \mathrm{clo}_f(o)$ as it is an intersection of sets which all have o as an element.

Concerning (2): suppose $x \in \mathrm{clo}_f(o)$. So if $o \in X$ and X is f-closed, then $x \in X$, and now $f(x) \in X$ as X is f-closed. So $f(x) \in \mathrm{clo}_f(o)$.

Concerning (3): quite generally, if $X \in C$ then $\bigcap C \subseteq X$. □

Using this, we can say:

Definition 6.4. A *Dedekind algebra* is a set A together with a function $f \colon A \to A$ and some $o \in A$ such that:

sfr:infinite:dedekind: ded:proper

1. $o \notin \mathrm{ran}(f)$

sfr:infinite:dedekind: ded:injection

2. f is an injection

sfr:infinite:dedekind: ded:closure

3. $A = \mathrm{clo}_f(o)$

Since $A = \mathrm{clo}_f(o)$, our earlier result tells us that A is the smallest f-closed set with o as an element. Clearly a Dedekind algebra is Dedekind infinite; just look at clauses (1) and (2) of the definition. But the more exciting fact is that any Dedekind infinite set can be turned into a Dedekind algebra.

sfr:infinite:dedekind: thm:DedekindInfiniteAlgebra

Theorem 6.5. *If there is a Dedekind infinite set, then there is a Dedekind algebra.*

Proof. Let D be Dedekind infinite. So there is an injection $g \colon D \to D$ and an element $o \in D \setminus \mathrm{ran}(g)$. Now let $A = \mathrm{clo}_g(o)$; by Lemma 6.3, A exists and $o \in A$. Let $f = g{\restriction_A}$. We will show that A, f, o comprise a Dedekind algebra.

Concerning (1): $o \notin \mathrm{ran}(g)$ and $\mathrm{ran}(f) \subseteq \mathrm{ran}(g)$ so $o \notin \mathrm{ran}(f)$.

Concerning (2): g is an injection on D; so $f \subseteq g$ must be an injection.

Concerning (3): by Lemma 6.3, A is g-closed; a fortiori, A is f-closed. So $\mathrm{clo}_f(o) \subseteq A$ by Lemma 6.3. Since also $\mathrm{clo}_f(o)$ is f-closed and $f = g{\restriction_A}$, it follows that $\mathrm{clo}_f(o)$ is g-closed. So $A \subseteq \mathrm{clo}_f(o)$ by Lemma 6.3. □

---

content/sets-functions-relations/infinite/dedekind-induction.tex

6.3 Dedekind Algebras and Arithmetical Induction

Crucially, now, a Dedekind algebra—indeed, *any* Dedekind algebra—will serve as a surrogate for the natural numbers. This is thanks to the following trivial consequence: sfr:infinite:induction:sec

Theorem 6.6 (Arithmetical induction). *Let N, s, o comprise a Dedekind algebra. Then for any set X:* sfr:infinite:induction:thm:dedinfiniteinduction

$$\text{if } o \in X \text{ and } (\forall n \in N \cap X)s(n) \in X, \text{ then } N \subseteq X.$$

Proof. By the definition of a Dedekind algebra, $N = \mathrm{clo}_s(o)$. Now if both $o \in X$ and $(\forall n \in N)(n \in X \to s(n) \in X)$, then $N = \mathrm{clo}_s(o) \subseteq X$. □

Since induction is characteristic of the natural numbers, the point is this. Given any Dedekind infinite set, we can form a Dedekind algebra, and use that algebra as our surrogate for the natural numbers.

Admittedly, Theorem 6.6 formulates induction in *set-theoretic* terms. But we can easily put the principle in terms which might be more familiar:

Corollary 6.7. *Let N, s, o comprise a Dedekind algebra. Then for any formula $\varphi(x)$, which may have parameters:* sfr:infinite:induction:natinductionschema

$$\text{if } \varphi(o) \text{ and } (\forall n \in N)(\varphi(n) \to \varphi(s(n))), \text{ then } (\forall n \in N)\varphi(n)$$

Proof. Let $X = \{n \in N : \varphi(n)\}$, and now use Theorem 6.6 □

In this result, we spoke of a formula "having parameters". What this means, roughly, is that for any objects $c_1, \ldots, c_k$, we can work with $\varphi(x, c_1, \ldots, c_k)$. More precisely, we can state the result without mentioning "parameters" as follows. For any formula $\varphi(x, v_1, \ldots, v_k)$, whose free variables are all displayed, we have:

$$\begin{aligned}\forall v_1 \ldots \forall v_k((&\varphi(o, v_1, \ldots, v_k) \wedge \\ &(\forall x \in N)(\varphi(x, v_1, \ldots, v_k) \to \varphi(s(x), v_1, \ldots, v_k))) \to \\ &(\forall x \in N)\varphi(x, v_1, \ldots, v_k))\end{aligned}$$

Evidently, speaking of "having parameters" can make things much easier to read. (In part XV, we will use this device rather frequently.)

Returning to Dedekind algebras: given any Dedekind algebra, we can also define the usual arithmetical functions of addition, multiplication and exponentiation. This is non-trivial, however, and it involves the technique of *recursive definition*. That is a technique which we shall introduce and justify much later, and in a much more general context. (Enthusiasts might want to revisit this after chapter 69, or perhaps read an alternative treatment, such as Potter 2004,

pp. 95–8.) But, where N, s, o comprise a Dedekind algebra, we will ultimately be able to stipulate the following:

$$\begin{array}{lll} a + o = a & a \times o = o & a^o = s(o) \\ a + s(b) = s(a + b) & a \times s(b) = (a \times b) + a & a^{s(b)} = a^b \times a \end{array}$$

and show that these behave as one would hope.

---

content/sets-functions-relations/infinite/dedekinds-proof.tex

6.4 Dedekind's "Proof" of the Existence of an Infinite Set

sfr:infinite:dedekindsproof:sec

In this chapter, we have offered a set-theoretic treatment of the natural numbers, in terms of Dedekind algebras. In section 5.5, we reflected on the philosophical significance of the arithmetisation of analysis (among other things). Now we should reflect on the significance of what we have achieved here.

Throughout chapter 5, we took the natural numbers as given, and used them to construct the integers, rationals, and reals, explicitly. In this chapter, we have not given an explicit construction of the natural numbers. We have just shown that, *given any Dedekind infinite set*, we can define a set which will behave just like we want $\mathbb{N}$ to behave.

Obviously, then, we cannot claim to have answered a metaphysical question, such as *which objects are the natural numbers*. But that's a good thing. After all, in section 5.5, we emphasized that we would be wrong to think of the definition of $\mathbb{R}$ as the set of Dedekind cuts as a *discovery*, rather than a convenient stipulation. The crucial observation is that the Dedekind cuts exemplify the key mathematical properties of the real numbers. So too here: the crucial observation is that *any* Dedekind algebra exemplifies the key mathematical properties of the natural numbers. (Indeed, Dedekind pushed this point home by proving that all Dedekind algebras are *isomorphic* (1888, Theorems 132–3). It is no surprise, then, that many contemporary "structuralists" cite Dedekind as a forerunner.)

Moreover, we have shown how to embed the theory of the natural numbers into a naïve simple set theory, which itself still remains rather informal, but which doesn't (apparently) assume the natural numbers as given. So, we may be on the way to realising Dedekind's own ambitious project, which he explained thus:

> In science nothing capable of proof ought to be believed without proof. Though this demand seems reasonable, I cannot regard it as having been met even in the most recent methods of laying the foundations of the simplest science; viz., that part of logic which deals with the theory of numbers. In speaking of arithmetic (algebra, analysis) as merely a part of logic I mean to imply that I consider the number-concept entirely independent of the notions or

> intuitions of space and time—that I rather consider it an immediate product of the pure laws of thought. (Dedekind, 1888, preface)

Dedekind's bold idea is this. We have just shown how to build the natural numbers using (naïve) set theory alone. In chapter 5, we saw how to construct the reals given the natural numbers and some set theory. So, perhaps, "arithmetic (algebra, analysis)" turn out to be "merely a part of logic" (in Dedekind's extended sense of the word "logic").

That's the idea. But hold on for a moment. Our construction of a Dedekind algebra (our surrogate for the natural numbers) is conditional on the existence of a Dedekind infinite set. (Just look back to Theorem 6.5.) Unless the existence of a Dedekind infinite set can be established via "logic" or "the pure laws of thought", the project stalls.

So, *can* the existence of a Dedekind infinite set be established by "the pure laws of thought"? Here was Dedekind's effort:

> My own realm of thoughts, i.e., the totality S of all things which can be objects of my thought, is infinite. For if s signifies an element of S, then the thought s' that s can be an object of my thought, is itself an element of S. If we regard this as an image $\varphi(s)$ of the element s, then ... S is [Dedekind] infinite, which was to be proved. (Dedekind, 1888, §66)

This is quite an astonishing thing to find in the middle of a book which largely consists of highly rigorous mathematical proofs. Two remarks are worth making.

First: this "proof" scarcely has what we would now recognize as a "mathematical" character. It speaks of psychological objects (thoughts), and merely *possible* ones at that.

Second: at least as we have presented Dedekind algebras, this "proof" has a straightforward technical shortcoming. If Dedekind's argument is successful, it establishes only that there are infinitely many things (specifically, infinitely many thoughts). But Dedekind also needs to give us a reason to regard S as a single *set*, with infinitely many elements, rather than thinking of S as *some things* (in the plural).

The fact that Dedekind did not see a gap here might suggest that his use of the word "totality" does not precisely track *our* use of the word "set".[1] But this would not be too surprising. The project we have pursued in the last two chapters—a "construction" of the naturals, and from them a "construction" of the integers, reals and rationals—has all been carried out naïvely. We have helped ourselves to this set, or that set, as and when we have needed them, without laying down many general principles concerning exactly which sets exist, and when. But we know that we need *some* general principles, for otherwise we will fall into Russell's Paradox.

The time has come for us to outgrow our naïvety.

[1] Indeed, we have other reasons to think it did not; see Potter (2004, p. 23).

content/sets-functions-relations/infinite/card-sb.tex

6.5 Appendix: Proving Schröder-Bernstein

sfr:infinite:card-sb:sec

Before we depart from naïve set theory, we have one last naïve (but sophisticated!) proof to consider. This is a proof of Schröder-Bernstein (Theorem 4.25): if $A \preceq B$ and $B \preceq A$ then $A \approx B$; i.e., given injections $f\colon A \to B$ and $g\colon B \to A$ there is a bijection $h\colon A \to B$.

In this chapter, we followed Dedekind's notion of *closures*. In fact, Dedekind provided a lovely proof of Schröder-Bernstein using this notion, and we will present it here. The proof closely follows Potter (2004, pp. 157–8), if you want a slightly different but essentially similar treatment. A little googling will also convince you that this is a theorem—rather like the irrationality of $\sqrt{2}$—for which *many* interesting and different proofs exist.

Using similar notation as Definition 6.2, let

$$\mathrm{Clo}_f(B) = \bigcap\{X : B \subseteq X \text{ and } X \text{ is } f\text{-closed}\}$$

for each set B and function f. Defined thus, $\mathrm{Clo}_f(B)$ is the smallest f-closed set containing B, in that:

sfr:infinite:card-sb:Closureprops

Lemma 6.8. *For any function f, and any B:*

sfr:infinite:card-sb:Closurehaselem

1. $B \subseteq \mathrm{Clo}_f(B)$*; and*

sfr:infinite:card-sb:Closureclosed

2. $\mathrm{Clo}_f(B)$ *is f-closed; and*

sfr:infinite:card-sb:Closuresmallest

3. *if X is f-closed and $B \subseteq X$, then $\mathrm{Clo}_f(B) \subseteq X$.*

Proof. Exactly as in Lemma 6.3. □

We need one last fact to get to Schröder-Bernstein:

sfr:infinite:card-sb:sbhelper

Proposition 6.9. *If $A \subseteq B \subseteq C$ and $A \approx C$, then $A \approx B \approx C$.*

Proof. Given a bijection $f\colon C \to A$, let $F = \mathrm{Clo}_f(C \setminus B)$ and define a function g with domain C as follows:

$$g(x) = \begin{cases} f(x) & \text{if } x \in F \\ x & \text{otherwise} \end{cases}$$

We'll show that g is a bijection from $C \to B$, from which it will follow that $g \circ f^{-1}\colon A \to B$ is a bijection, completing the proof.

First we claim that if $x \in F$ but $y \notin F$ then $g(x) \neq g(y)$. For reductio suppose otherwise, so that $y = g(y) = g(x) = f(x)$. Since $x \in F$ and F is f-closed by Lemma 6.8, we have $y = f(x) \in F$, a contradiction.

Now suppose $g(x) = g(y)$. So, by the above, $x \in F$ iff $y \in F$. If $x, y \in F$, then $f(x) = g(x) = g(y) = f(y)$ so that $x = y$ since f is a bijection. If $x, y \notin F$, then $x = g(x) = g(y) = y$. So g is an injection.

It remains to show that $\mathrm{ran}(g) = B$. So fix $x \in B \subseteq C$. If $x \notin F$, then $g(x) = x$. If $x \in F$, then $x = f(y)$ for some $y \in F$, since otherwise $F \setminus \{x\}$ would be f-closed and extend $C \setminus B$, which is impossible by Lemma 6.8; now $g(y) = f(y) = x$. □

Finally, here is the proof of the main result. Recall that given a function h and set D, we define $h[D] = \{h(x) : x \in D\}$.

Proof of Schröder-Bernstein. Let $f\colon A \to B$ and $g\colon B \to A$ be injections. Since $f[A] \subseteq B$ we have that $g[f[A]] \subseteq g[B] \subseteq A$. Also, $g \circ f\colon A \to g[f[A]]$ is an injection since both g and f are; and indeed $g \circ f$ is a bijection, just by the way we defined its codomain. So $g[f[A]] \approx A$, and hence by Proposition 6.9 there is a bijection $h\colon A \to g[B]$. Moreover, g^{-1} is a bijection $g[B] \to B$. So $g^{-1} \circ h\colon A \to B$ is a bijection. □

Part II

Propositional Logic

This part contains material on classical propositional logic. The first chapter is relatively rudimentary and just lists definitions and results, many proofs are not carried out but are left as exercises. The material on proof systems and the completeness theorem is included from the part on first-order logic, with the "FOL" tag set to false. This leaves out everything related to predicates, terms, and quantifiers, and replaces talk of structures $\mathfrak{M}$ with talk about valuations $\mathfrak{v}$.

It is planned to expand this part to include more detail, and to add further topics and results, such as truth-functional completeness.

Chapter 7

Syntax and Semantics

This is a very quick summary of definitions only. It should be expanded to provide a gentle intro to proofs by induction on formulas, with lots more examples.

---

content/propositional-logic/syntax-and-semantics/introduction.tex

7.1 Introduction

pl:syn:int:sec

Propositional logic deals with formulas that are built from propositional variables using the propositional connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. Intuitively, a propositional variable p stands for a sentence or proposition that is true or false. Whenever the "truth value" of the propositional variable in a formula

is determined, so is the truth value of any formulas formed from them using propositional connectives. We say that propositional logic is *truth functional*, because its semantics is given by functions of truth values. In particular, in propositional logic we leave out of consideration any further determination of truth and falsity, e.g., whether something is necessarily true rather than just contingently true, or whether something is known to be true, or whether something is true now rather than was true or will be true. We only consider two truth values true ($\mathbb{T}$) and false ($\mathbb{F}$), and so exclude from discussion the possibility that a statement may be neither true nor false, or only half true. We also concentrate only on connectives where the truth value of a formula built from them is completely determined by the truth values of its parts (and not, say, on its meaning). In particular, whether the truth value of conditionals in English is truth functional in this sense is contentious. The material conditional $\rightarrow$ is; other logics deal with conditionals that are not truth functional.

In order to develop the theory and metatheory of truth-functional propositional logic, we must first define the syntax and semantics of its expressions. We will describe one way of constructing formulas from propositional variables using the connectives. Alternative definitions are possible. Other systems will choose different symbols, will select different sets of connectives as primitive, and will use parentheses differently (or even not at all, as in the case of so-called Polish notation). What all approaches have in common, though, is that the formation rules define the set of formulas *inductively*. If done properly, every expression can result essentially in only one way according to the formation rules. The inductive definition resulting in expressions that are *uniquely readable* means we can give meanings to these expressions using the same method—inductive definition.

Giving the meaning of expressions is the domain of semantics. The central concept in semantics for propositional logic is that of satisfaction in a valuation. A valuation $\mathfrak{v}$ assigns truth values $\mathbb{T}$, $\mathbb{F}$ to the propositional variables. Any valuation determines a truth value $\overline{\mathfrak{v}}(\varphi)$ for any formula φ. A formula is satisfied in a valuation $\mathfrak{v}$ iff $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$—we write this as $\mathfrak{v} \vDash \varphi$. This relation can also be defined by induction on the structure of φ, using the truth functions for the logical connectives to define, say, satisfaction of $\varphi \wedge \psi$ in terms of satisfaction (or not) of φ and ψ.

On the basis of the satisfaction relation $\mathfrak{v} \vDash \varphi$ for sentences we can then define the basic semantic notions of tautology, entailment, and satisfiability. A formula is a tautology, $\vDash \varphi$, if every valuation satisfies it, i.e., $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$ for any $\mathfrak{v}$. It is entailed by a set of formulas, $\Gamma \vDash \varphi$, if every valuation that satisfies all the formulas in Γ also satisfies φ. And a set of formulas is satisfiable if some valuation satisfies all formulas in it at the same time. Because formulas are inductively defined, and satisfaction is in turn defined by induction on the structure of formulas, we can use induction to prove properties of our semantics and to relate the semantic notions defined.

---

content/propositional-logic/syntax-and-semantics/formulas.tex

7.2 Propositional Formulas

pl:syn:fml:sec

Formulas of propositional logic are built up from *propositional variables*, the propositional constant $\bot$ and the propositional constant $\top$ using *logical connectives*.

1. A denumerable set At_0 of propositional variables p_0, p_1, ...
2. The propositional constant for falsity $\bot$.
3. The propositional constant for truth $\top$.
4. The logical connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\to$ (conditional), $\leftrightarrow$ (biconditional)
5. Punctuation marks: (,), and the comma.

We denote this language of propositional logic by $\mathcal{L}_0$.

.

intro

You may be familiar with different terminology and symbols than the ones we use above. Logic texts (and teachers) commonly use either $\sim$, $\neg$, and ! for "negation", $\wedge$, $\cdot$, and & for "conjunction". Commonly used symbols for the "conditional" or "implication" are $\to$, $\Rightarrow$, and $\supset$. Symbols for "biconditional," "bi-implication," or "(material) equivalence" are $\leftrightarrow$, $\Leftrightarrow$, and $\equiv$. The $\bot$ symbol is variously called "falsity," "falsum," "absurdity," or "bottom." The $\top$ symbol is variously called "truth," "verum," or "top."

pl:syn:fml:defn:formulas

Definition 7.1 (Formula). The set $\mathrm{Frm}(\mathcal{L}_0)$ of *formulas* of propositional logic is defined inductively as follows:

1. $\bot$ is an atomic formula.
2. $\top$ is an atomic formula.
3. Every propositional variable p_i is an atomic formula.
4. If φ is a formula, then $\neg\varphi$ is a formula.
5. If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.
6. If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.
7. If φ and ψ are formulas, then $(\varphi \to \psi)$ is a formula.
8. If φ and ψ are formulas, then $(\varphi \leftrightarrow \psi)$ is a formula.
9. Nothing else is a formula.

explanation

The definition of formulas is an *inductive definition*. Essentially, we construct the set of formulas in infinitely many stages. In the initial stage, we pronounce all atomic formulas to be formulas; this corresponds to the first few cases of the definition, i.e., the cases for $\top$, $\bot$, p_i. "Atomic formula" thus means any formula of this form.

The other cases of the definition give rules for constructing new formulas out of formulas already constructed. At the second stage, we can use them to construct formulas out of atomic formulas. At the third stage, we construct new formulas from the atomic formulas and those obtained in the second stage, and so on. A formula is anything that is eventually constructed at such a stage, and nothing else.

When writing a formula $(\psi * \chi)$ constructed from ψ, χ using a two-place connective $*$, we will often leave out the outermost pair of parentheses and write simply $\psi * \chi$.

Definition 7.2 (Syntactic identity). The symbol $\equiv$ expresses syntactic identity between strings of symbols, i.e., $\varphi \equiv \psi$ iff φ and ψ are strings of symbols of the same length and which contain the same symbol in each place.

The $\equiv$ symbol may be flanked by strings obtained by concatenation, e.g., $\varphi \equiv (\psi \vee \chi)$ means: the string of symbols φ is the same string as the one obtained by concatenating an opening parenthesis, the string ψ, the $\vee$ symbol, the string χ, and a closing parenthesis, in this order. If this is the case, then we know that the first symbol of φ is an opening parenthesis, φ contains ψ as a substring (starting at the second symbol), that substring is followed by $\vee$, etc.

---

content/propositional-logic/syntax-and-semantics/preliminaries.tex

7.3 Preliminaries

pl:syn:pre:sec
pl:syn:pre:thm:induction

Theorem 7.3 (*Principle of induction on formulas*). *If some property P holds for all the atomic formulas and is such that*

1. *it holds for $\neg\varphi$ whenever it holds for φ;*
2. *it holds for $(\varphi \wedge \psi)$ whenever it holds for φ and ψ;*
3. *it holds for $(\varphi \vee \psi)$ whenever it holds for φ and ψ;*
4. *it holds for $(\varphi \to \psi)$ whenever it holds for φ and ψ;*
5. *it holds for $(\varphi \leftrightarrow \psi)$ whenever it holds for φ and ψ;*

then P holds for all formulas.

Proof. Let S be the collection of all formulas with property P. Clearly $S \subseteq \mathrm{Frm}(\mathcal{L}_0)$. S satisfies all the conditions of Definition 7.1: it contains all atomic formulas and is closed under the logical operators. $\mathrm{Frm}(\mathcal{L}_0)$ is the smallest such class, so $\mathrm{Frm}(\mathcal{L}_0) \subseteq S$. So $\mathrm{Frm}(\mathcal{L}_0) = S$, and every formula has property P. □

pl:syn:pre:prop:balanced

Proposition 7.4. *Any formula in* $\mathrm{Frm}(\mathcal{L}_0)$ *is* balanced, *in that it has as many left parentheses as right ones.*

Problem 7.1. Prove Proposition 7.4

pl:syn:pre:prop:noinit

Proposition 7.5. *No proper initial segment of a formula is a formula.*

Problem 7.2. Prove Proposition 7.5

Proposition 7.6 (Unique Readability). *Any formula* φ *in* $\mathrm{Frm}(\mathcal{L}_0)$ *has exactly one parsing as one of the following*

1. $\bot$.
2. $\top$.
3. p_n *for some* $p_n \in \mathrm{At}_0$.
4. $\neg\psi$ *for some formula* ψ.
5. $(\psi \wedge \chi)$ *for some formulas* ψ *and* χ.
6. $(\psi \vee \chi)$ *for some formulas* ψ *and* χ.
7. $(\psi \to \chi)$ *for some formulas* ψ *and* χ.
8. $(\psi \leftrightarrow \chi)$ *for some formulas* ψ *and* χ.

Moreover, this parsing is unique.

Proof. By induction on φ. For instance, suppose that φ has two distinct readings as $(\psi \to \chi)$ and $(\psi' \to \chi')$. Then ψ and ψ' must be the same (or else one would be a proper initial segment of the other); so if the two readings of φ are distinct it must be because χ and χ' are distinct readings of the same sequence of symbols, which is impossible by the inductive hypothesis. □

Definition 7.7 (Uniform Substitution). If φ and ψ are formulas, and p_i is a propositional variable, then $\varphi[\psi/p_i]$ denotes the result of replacing each occurrence of p_i by an occurrence of ψ in φ; similarly, the simultaneous substitution of $p_1, \ldots, p_n$ by formulas $\psi_1, \ldots, \psi_n$ is denoted by $\varphi[\psi_1/p_1, \ldots, \psi_n/p_n]$.

Problem 7.3. For each of the five formulas below determine whether the formula can be expressed as a substitution $\varphi[\psi/p_i]$ where φ is (i) p_0; (ii) $(\neg p_0 \wedge p_1)$; and (iii) $((\neg p_0 \to p_1) \wedge p_2)$. In each case specify the relevant substitution.

1. p_1

2. $(\neg p_0 \land p_0)$
3. $((p_0 \lor p_1) \land p_2)$
4. $\neg((p_0 \to p_1) \land p_2)$
5. $((\neg(p_0 \to p_1) \to (p_0 \lor p_1)) \land \neg(p_0 \land p_1))$

Problem 7.4. Give a mathematically rigorous definition of $\varphi[\psi/p]$ by induction.

---

content/propositional-logic/syntax-and-semantics/formation-sequences.tex

7.4 Formation Sequences

Defining formulas via an inductive definition, and the complementary technique of proving properties of formulas via induction, is an elegant and efficient approach. However, it can also be useful to consider a more bottom-up, step-by-step approach to the construction of formulas, which we do here using the notion of a *formation sequence*. pl:syn:fseq:sec

Definition 7.8 (Formation sequences for formulas). A finite sequence $\langle \varphi_0, \ldots, \varphi_n \rangle$ of strings of symbols from the language $\mathcal{L}_0$ is a *formation sequence* for φ if $\varphi \equiv \varphi_n$ and for all $i \leq n$, either φ_i is an atomic formula or there exist $j, k < i$ such that one of the following holds: pl:syn:fseq:defn:fseq-frm

1. $\varphi_i \equiv \neg\varphi_j$.
2. $\varphi_i \equiv (\varphi_j \land \varphi_k)$.
3. $\varphi_i \equiv (\varphi_j \lor \varphi_k)$.
4. $\varphi_i \equiv (\varphi_j \to \varphi_k)$.
5. $\varphi_i \equiv (\varphi_j \leftrightarrow \varphi_k)$.

Example 7.9.

$$\langle p_0, p_1, (p_1 \land p_0), \neg(p_1 \land p_0) \rangle$$

is a formation sequence of $\neg(p_1 \land p_0)$, as is

$$\langle p_0, p_1, p_0, (p_1 \land p_0), (p_0 \to p_1), \neg(p_1 \land p_0) \rangle.$$

As can be seen from the second example, formation sequences may contain 'junk': formulas which are redundant or do not contribute to the construction.

Proposition 7.10. *Every formula φ in $\mathrm{Frm}(\mathcal{L}_0)$ has a formation sequence.* pl:syn:fseq:prop:formed

Proof. Suppose φ is atomic. Then the sequence $\langle\varphi\rangle$ is a formation sequence for φ. Now suppose that ψ and χ have formation sequences $\langle\psi_0, \ldots, \psi_n\rangle$ and $\langle\chi_0, \ldots, \chi_m\rangle$ respectively.

1. If $\varphi \equiv \neg\psi$, then $\langle\psi_0, \ldots, \psi_n, \neg\psi_n\rangle$ is a formation sequence for φ.
2. If $\varphi \equiv (\psi \wedge \chi)$, then $\langle\psi_0, \ldots, \psi_n, \chi_0, \ldots, \chi_m, (\psi_n \wedge \chi_m)\rangle$ is a formation sequence for φ.
3. If $\varphi \equiv (\psi \vee \chi)$, then $\langle\psi_0, \ldots, \psi_n, \chi_0, \ldots, \chi_m, (\psi_n \vee \chi_m)\rangle$ is a formation sequence for φ.
4. If $\varphi \equiv (\psi \to \chi)$, then $\langle\psi_0, \ldots, \psi_n, \chi_0, \ldots, \chi_m, (\psi_n \to \chi_m)\rangle$ is a formation sequence for φ.
5. If $\varphi \equiv (\psi \leftrightarrow \chi)$, then $\langle\psi_0, \ldots, \psi_n, \chi_0, \ldots, \chi_m, (\psi_n \leftrightarrow \chi_m)\rangle$ is a formation sequence for φ.

By the principle of induction on formulas, every formula has a formation sequence. □

We can also prove the converse. This is important because it shows that our two ways of defining formulas are equivalent: they give the same results. It also means that we can prove theorems about formulas by using ordinary induction on the length of formation sequences.

pl:syn:fseq: lem:fseq-init

Lemma 7.11. *Suppose that $\langle\varphi_0, \ldots, \varphi_n\rangle$ is a formation sequence for φ_n, and that $k \leq n$. Then $\langle\varphi_0, \ldots, \varphi_k\rangle$ is a formation sequence for φ_k.*

Proof. Exercise. □

pl:syn:fseq: thm:fseq-frm-equiv

Theorem 7.12. $\mathrm{Frm}(\mathcal{L}_0)$ *is the set of all strings of symbols in the language $\mathcal{L}_0$ with a formation sequence.*

Proof. Let F be the set of all strings of symbols in the language $\mathcal{L}_0$ that have a formation sequence. We have seen in Proposition 7.10 that $\mathrm{Frm}(\mathcal{L}_0) \subseteq F$, so now we prove the converse.

Suppose φ has a formation sequence $\langle\varphi_0, \ldots, \varphi_n\rangle$. We prove that $\varphi \in \mathrm{Frm}(\mathcal{L}_0)$ by strong induction on n. Our induction hypothesis is that every string of symbols with a formation sequence of length $m < n$ is in $\mathrm{Frm}(\mathcal{L}_0)$. By the definition of a formation sequence, either φ_n is atomic or there must exist $j, k < n$ such that one of the following is the case:

1. $\varphi_n \equiv \neg\varphi_j$.
2. $\varphi_n \equiv (\varphi_j \wedge \varphi_k)$.
3. $\varphi_n \equiv (\varphi_j \vee \varphi_k)$.
4. $\varphi_n \equiv (\varphi_j \to \varphi_k)$.

5. $\varphi_n \equiv (\varphi_j \leftrightarrow \varphi_k)$.

Now we reason by cases. If φ_n is atomic then $\varphi_n \in \mathrm{Frm}(\mathcal{L}_0)$. Suppose instead that $\varphi \equiv (\varphi_j \wedge \varphi_k)$. By Lemma 7.11, $\langle \varphi_0, \ldots, \varphi_j \rangle$ and $\langle \varphi_0, \ldots, \varphi_k \rangle$ are formation sequences for φ_j and φ_k respectively. Since these are proper initial subsequences of the formation sequence for φ, they both have length less than n. Therefore by the induction hypothesis, φ_j and φ_k are in $\mathrm{Frm}(\mathcal{L}_0)$, and so by the definition of a formula, so is $(\varphi_j \wedge \varphi_k)$. The other cases follow by parallel reasoning. □

---

content/propositional-logic/syntax-and-semantics/valuations-sat.tex

7.5 Valuations and Satisfaction

pl:syn:val:sec

Definition 7.13 (Valuations). Let $\{\mathbb{T}, \mathbb{F}\}$ be the set of the two truth values, "true" and "false." A *valuation* for $\mathcal{L}_0$ is a function $\mathfrak{v}$ assigning either $\mathbb{T}$ or $\mathbb{F}$ to the propositional variables of the language, i.e., $\mathfrak{v} \colon \mathrm{At}_0 \to \{\mathbb{T}, \mathbb{F}\}$.

Definition 7.14. Given a valuation $\mathfrak{v}$, define the evaluation function $\overline{\mathfrak{v}} \colon \mathrm{Frm}(\mathcal{L}_0) \to \{\mathbb{T}, \mathbb{F}\}$ inductively by:

$$
\begin{aligned}
\overline{\mathfrak{v}}(\bot) &= \mathbb{F};\\
\overline{\mathfrak{v}}(\top) &= \mathbb{T};\\
\overline{\mathfrak{v}}(p_n) &= \mathfrak{v}(p_n);\\
\overline{\mathfrak{v}}(\neg\varphi) &= \begin{cases} \mathbb{T} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{F};\\ \mathbb{F} & \text{otherwise.}\end{cases}\\
\overline{\mathfrak{v}}(\varphi \wedge \psi) &= \begin{cases} \mathbb{T} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{T} \text{ and } \overline{\mathfrak{v}}(\psi) = \mathbb{T};\\ \mathbb{F} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{F} \text{ or } \overline{\mathfrak{v}}(\psi) = \mathbb{F}.\end{cases}\\
\overline{\mathfrak{v}}(\varphi \vee \psi) &= \begin{cases} \mathbb{T} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{T} \text{ or } \overline{\mathfrak{v}}(\psi) = \mathbb{T};\\ \mathbb{F} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{F} \text{ and } \overline{\mathfrak{v}}(\psi) = \mathbb{F}.\end{cases}\\
\overline{\mathfrak{v}}(\varphi \to \psi) &= \begin{cases} \mathbb{T} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{F} \text{ or } \overline{\mathfrak{v}}(\psi) = \mathbb{T};\\ \mathbb{F} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{T} \text{ and } \overline{\mathfrak{v}}(\psi) = \mathbb{F}.\end{cases}\\
\overline{\mathfrak{v}}(\varphi \leftrightarrow \psi) &= \begin{cases} \mathbb{T} & \text{if } \overline{\mathfrak{v}}(\varphi) = \overline{\mathfrak{v}}(\psi);\\ \mathbb{F} & \text{if } \overline{\mathfrak{v}}(\varphi) \neq \overline{\mathfrak{v}}(\psi).\end{cases}
\end{aligned}
$$

explanation

The clauses correspond to the following truth tables:

φ	$\neg\varphi$
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$

φ	ψ	$\varphi \wedge \psi$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

φ	ψ	$\varphi \vee \psi$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

φ	ψ	$\varphi \to \psi$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$

φ	ψ	$\varphi \leftrightarrow \psi$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$

Problem 7.5. Consider adding to $\mathcal{L}_0$ a ternary connective $\diamondsuit$ with evaluation given by

$$\overline{\mathfrak{v}}(\diamondsuit(\varphi, \psi, \chi)) = \begin{cases} \overline{\mathfrak{v}}(\psi) & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{T}; \\ \overline{\mathfrak{v}}(\chi) & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{F}. \end{cases}$$

Write down the truth table for this connective.

pl:syn:val: thm:LocalDetermination

Theorem 7.15 (Local Determination). *Suppose that $\mathfrak{v}_1$ and $\mathfrak{v}_2$ are valuations that agree on the propositional variables occurring in φ, i.e., $\mathfrak{v}_1(p_n) = \mathfrak{v}_2(p_n)$ whenever p_n occurs in some formula φ. Then $\overline{\mathfrak{v}_1}$ and $\overline{\mathfrak{v}_2}$ also agree on φ, i.e., $\overline{\mathfrak{v}_1}(\varphi) = \overline{\mathfrak{v}_2}(\varphi)$.*

Proof. By induction on φ. □

pl:syn:val: defn:satisfaction

Definition 7.16 (Satisfaction). We can inductively define the notion of *satisfaction of a formula φ by a valuation $\mathfrak{v}$*, $\mathfrak{v} \vDash \varphi$, as follows. (We write $\mathfrak{v} \nvDash \varphi$ to mean "not $\mathfrak{v} \vDash \varphi$.")

1. $\varphi \equiv \bot$: $\mathfrak{v} \nvDash \varphi$.
2. $\varphi \equiv \top$: $\mathfrak{v} \vDash \varphi$.
3. $\varphi \equiv p_i$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v}(p_i) = \mathbb{T}$.
4. $\varphi \equiv \neg\psi$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \nvDash \psi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \vDash \psi$ and $\mathfrak{v} \vDash \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \vDash \psi$ or $\mathfrak{v} \vDash \chi$ (or both).
7. $\varphi \equiv (\psi \to \chi)$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \nvDash \psi$ or $\mathfrak{v} \vDash \chi$ (or both).
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{v} \vDash \varphi$ iff either both $\mathfrak{v} \vDash \psi$ and $\mathfrak{v} \vDash \chi$, or neither $\mathfrak{v} \vDash \psi$ nor $\mathfrak{v} \vDash \chi$.

If Γ is a set of formulas, $\mathfrak{v} \vDash \Gamma$ iff $\mathfrak{v} \vDash \varphi$ for every $\varphi \in \Gamma$.

Proposition 7.17. $\mathfrak{v} \vDash \varphi$ *iff* $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$. pl:syn:val: prop:sat-value

Proof. By induction on φ. □

Problem 7.6. Prove Proposition 7.17

---

content/propositional-logic/syntax-and-semantics/semantic-notions.tex

7.6 Semantic Notions

We define the following semantic notions: pl:syn:sem: sec

Definition 7.18.

1. A formula φ is *satisfiable* if for some $\mathfrak{v}$, $\mathfrak{v} \vDash \varphi$; it is *unsatisfiable* if for no $\mathfrak{v}$, $\mathfrak{v} \vDash \varphi$;
2. A formula φ is a *tautology* if $\mathfrak{v} \vDash \varphi$ for all valuations $\mathfrak{v}$;
3. A formula φ is *contingent* if it is satisfiable but not a tautology;
4. If Γ is a set of formulas, $\Gamma \vDash \varphi$ ("Γ entails φ") if and only if $\mathfrak{v} \vDash \varphi$ for every valuation $\mathfrak{v}$ for which $\mathfrak{v} \vDash \Gamma$.
5. If Γ is a set of formulas, Γ is *satisfiable* if there is a valuation $\mathfrak{v}$ for which $\mathfrak{v} \vDash \Gamma$, and Γ is *unsatisfiable* otherwise.

Problem 7.7. For each of the following four formulas determine whether it is (a) satisfiable, (b) tautology, and (c) contingent.

1. $(p_0 \to (\neg p_1 \to \neg p_0))$.
2. $((p_0 \land \neg p_1) \to (\neg p_0 \land p_2)) \leftrightarrow ((p_2 \to p_0) \to (p_0 \to p_1))$.
3. $(p_0 \leftrightarrow p_1) \to (p_2 \leftrightarrow \neg p_1)$.
4. $((p_0 \leftrightarrow (\neg p_1 \land p_2)) \lor (p_2 \to (p_0 \leftrightarrow p_1)))$.

Proposition 7.19. pl:syn:sem: prop:semanticalfacts

1. *φ is a tautology if and only if $\emptyset \vDash \varphi$;*
2. *If $\Gamma \vDash \varphi$ and $\Gamma \vDash \varphi \to \psi$ then $\Gamma \vDash \psi$;*
3. *If Γ is satisfiable then every finite subset of Γ is also satisfiable;*
4. *Monotonicity: if $\Gamma \subseteq \Delta$ and $\Gamma \vDash \varphi$ then also $\Delta \vDash \varphi$;* pl:syn:sem: def:monotonicity
5. *Transitivity: if $\Gamma \vDash \varphi$ and $\Delta \cup \{\varphi\} \vDash \psi$ then $\Gamma \cup \Delta \vDash \psi$.* pl:syn:sem: def:Cut

Proof. Exercise. □

Problem 7.8. Prove Proposition 7.19

pl:syn:sem: prop:entails-unsat

Proposition 7.20. *$\Gamma \vDash \varphi$ if and only if $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable.*

Proof. Exercise. □

Problem 7.9. Prove Proposition 7.20

pl:syn:sem: thm:sem-deduction

Theorem 7.21 (Semantic Deduction Theorem). *$\Gamma \vDash \varphi \to \psi$ if and only if $\Gamma \cup \{\varphi\} \vDash \psi$.*

Proof. Exercise. □

Problem 7.10. Prove Theorem 7.21

Chapter 8

Derivation Systems

This chapter collects general material on derivation systems. A textbook using a specific system can insert the introduction section plus the relevant survey section at the beginning of the chapter introducing that system.

---

content/propositional-logic/../first-order-logic/proof-systems/introduction.tex

8.1 Introduction

pl:prf:int: sec

Logics commonly have both a semantics and a derivation system. The semantics concerns concepts such as truth, satisfiability, validity, and entailment. The purpose of derivation systems is to provide a purely syntactic method of establishing entailment and validity. They are purely syntactic in the sense that a derivation in such a system is a finite syntactic object, usually a sequence (or other finite arrangement) of sentences or formulas. Good derivation systems have the property that any given sequence or arrangement of sentences or formulas can be verified mechanically to be "correct."

The simplest (and historically first) derivation systems for first-order logic were *axiomatic*. A sequence of formulas counts as a derivation in such a system if each individual formula in it is either among a fixed set of "axioms" or follows from formulas coming before it in the sequence by one of a fixed number of "inference rules"—and it can be mechanically verified if a formula is an axiom and whether it follows correctly from other formulas by one of the inference rules. Axiomatic derivation systems are easy to describe—and also easy to handle meta-theoretically—but derivations in them are hard to read and understand, and are also hard to produce.

Other derivation systems have been developed with the aim of making it easier to construct derivations or easier to understand derivations once they are complete. Examples are natural deduction, truth trees, also known as tableaux proofs, and the sequent calculus. Some derivation systems are designed especially with mechanization in mind, e.g., the resolution method is easy to implement in software (but its derivations are essentially impossible to understand). Most of these other derivation systems represent derivations as trees of formulas rather than sequences. This makes it easier to see which parts of a derivation depend on which other parts.

So for a given logic, such as first-order logic, the different derivation systems will give different explications of what it is for a sentence to be a *theorem* and what it means for a sentence to be derivable from some others. However that is done (via axiomatic derivations, natural deductions, sequent derivations, truth trees, resolution refutations), we want these relations to match the semantic notions of validity and entailment. Let's write $\vdash \varphi$ for "φ is a theorem" and "$\Gamma \vdash \varphi$" for "φ is derivable from Γ." However $\vdash$ is defined, we want it to match up with $\models$, that is:

1. $\vdash \varphi$ if and only if $\models \varphi$
2. $\Gamma \vdash \varphi$ if and only if $\Gamma \models \varphi$

The "only if" direction of the above is called *soundness*. A derivation system is sound if derivability guarantees entailment (or validity). Every decent derivation system has to be sound; unsound derivation systems are not useful at all. After all, the entire purpose of a derivation is to provide a syntactic guarantee of validity or entailment. We'll prove soundness for the derivation systems we present.

The converse "if" direction is also important: it is called *completeness*. A complete derivation system is strong enough to show that φ is a theorem whenever φ is valid, and that $\Gamma \vdash \varphi$ whenever $\Gamma \models \varphi$. Completeness is harder to establish, and some logics have no complete derivation systems. First-order logic does. Kurt Gödel was the first one to prove completeness for a derivation system of first-order logic in his 1929 dissertation.

Another concept that is connected to derivation systems is that of *consistency*. A set of sentences is called inconsistent if anything whatsoever can be derived from it, and consistent otherwise. Inconsistency is the syntactic counterpart to unsatisfiablity: like unsatisfiable sets, inconsistent sets of sentences

do not make good theories, they are defective in a fundamental way. Consistent sets of sentences may not be true or useful, but at least they pass that minimal threshold of logical usefulness. For different derivation systems the specific definition of consistency of sets of sentences might differ, but like $\vdash$, we want consistency to coincide with its semantic counterpart, satisfiability. We want it to always be the case that Γ is consistent if and only if it is satisfiable. Here, the "only if" direction amounts to completeness (consistency guarantees satisfiability), and the "if" direction amounts to soundness (satisfiability guarantees consistency). In fact, for classical first-order logic, the two versions of soundness and completeness are equivalent.

---

content/propositional-logic/../first-order-logic/proof-systems/sequent-calculus.tex

8.2 The Sequent Calculus

pl:prf:seq:sec

While many derivation systems operate with arrangements of sentences, the sequent calculus operates with *sequents*. A sequent is an expression of the form

$$\varphi_1, \ldots, \varphi_m \Rightarrow \psi_1, \ldots, \psi_m,$$

that is a pair of sequences of sentences, separated by the sequent symbol $\Rightarrow$. Either sequence may be empty. A derivation in the sequent calculus is a tree of sequents, where the topmost sequents are of a special form (they are called "initial sequents" or "axioms") and every other sequent follows from the sequents immediately above it by one of the rules of inference. The rules of inference either manipulate the sentences in the sequents (adding, removing, or rearranging them on either the left or the right), or they introduce a complex formula in the conclusion of the rule. For instance, the $\wedge$L rule allows the inference from $\varphi, \Gamma \Rightarrow \Delta$ to $\varphi \wedge \psi, \Gamma \Rightarrow \Delta$, and the $\rightarrow$R allows the inference from $\varphi, \Gamma \Rightarrow \Delta, \psi$ to $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$, for any Γ, Δ, φ, and ψ. (In particular, Γ and Δ may be empty.)

The $\vdash$ relation based on the sequent calculus is defined as follows: $\Gamma \vdash \varphi$ iff there is some sequence Γ_0 such that every φ in Γ_0 is in Γ and there is a derivation with the sequent $\Gamma_0 \Rightarrow \varphi$ at its root. φ is a theorem in the sequent calculus if the sequent $\Rightarrow \varphi$ has a derivation. For instance, here is a derivation that shows that $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

$$\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi}\,\wedge\text{L}}{\Rightarrow (\varphi \wedge \psi) \rightarrow \varphi}\,\rightarrow\text{R}$$

A set Γ is inconsistent in the sequent calculus if there is a derivation of $\Gamma_0 \Rightarrow$ (where every $\varphi \in \Gamma_0$ is in Γ and the right side of the sequent is empty). Using the rule WR, any sentence can be derived from an inconsistent set.

The sequent calculus was invented in the 1930s by Gerhard Gentzen. Because of its systematic and symmetric design, it is a very useful formalism for developing a theory of derivations. It is relatively easy to find derivations in

the sequent calculus, but these derivations are often hard to read and their connection to proofs are sometimes not easy to see. It has proved to be a very elegant approach to derivation systems, however, and many logics have sequent calculus systems.

---

content/propositional-logic/../first-order-logic/proof-systems/natural-deduction.te

8.3 Natural Deduction

pl:prf:ntd:sec

Natural deduction is a derivation system intended to mirror actual reasoning (especially the kind of regimented reasoning employed by mathematicians). Actual reasoning proceeds by a number of "natural" patterns. For instance, proof by cases allows us to establish a conclusion on the basis of a disjunctive premise, by establishing that the conclusion follows from either of the disjuncts. Indirect proof allows us to establish a conclusion by showing that its negation leads to a contradiction. Conditional proof establishes a conditional claim "if ... then ..." by showing that the consequent follows from the antecedent. Natural deduction is a formalization of some of these natural inferences. Each of the logical connectives and quantifiers comes with two rules, an introduction and an elimination rule, and they each correspond to one such natural inference pattern. For instance, $\rightarrow$Intro corresponds to conditional proof, and $\vee$Elim to proof by cases. A particularly simple rule is $\wedge$Elim which allows the inference from $\varphi \wedge \psi$ to φ (or ψ).

One feature that distinguishes natural deduction from other derivation systems is its use of assumptions. A derivation in natural deduction is a tree of formulas. A single formula stands at the root of the tree of formulas, and the "leaves" of the tree are formulas from which the conclusion is derived. In natural deduction, some leaf formulas play a role inside the derivation but are "used up" by the time the derivation reaches the conclusion. This corresponds to the practice, in actual reasoning, of introducing hypotheses which only remain in effect for a short while. For instance, in a proof by cases, we assume the truth of each of the disjuncts; in conditional proof, we assume the truth of the antecedent; in indirect proof, we assume the truth of the negation of the conclusion. This way of introducing hypothetical assumptions and then doing away with them in the service of establishing an intermediate step is a hallmark of natural deduction. The formulas at the leaves of a natural deduction derivation are called assumptions, and some of the rules of inference may "discharge" them. For instance, if we have a derivation of ψ from some assumptions which include φ, then the $\rightarrow$Intro rule allows us to infer $\varphi \rightarrow \psi$ and discharge any assumption of the form φ. (To keep track of which assumptions are discharged at which inferences, we label the inference and the assumptions it discharges with a number.) The assumptions that remain undischarged at the end of the derivation are together sufficient for the truth of the conclusion, and so a derivation establishes that its undischarged assumptions entail its conclusion.

The relation $\Gamma \vdash \varphi$ based on natural deduction holds iff there is a derivation in which φ is the last sentence in the tree, and every leaf which is undischarged is in Γ. φ is a theorem in natural deduction iff there is a derivation in which φ is the last sentence and all assumptions are discharged. For instance, here is a derivation that shows that $\vdash (\varphi \wedge \psi) \to \varphi$:

$$\dfrac{\dfrac{[\varphi \wedge \psi]^1}{\varphi}\,\wedge\text{Elim}}{(\varphi \wedge \psi) \to \varphi}\,{}^{1}\,\to\text{Intro}$$

The label 1 indicates that the assumption $\varphi \wedge \psi$ is discharged at the $\to$Intro inference.

A set Γ is inconsistent iff $\Gamma \vdash \bot$ in natural deduction. The rule $\bot_I$ makes it so that from an inconsistent set, any sentence can be derived.

Natural deduction systems were developed by Gerhard Gentzen and Stanisław Jaśkowski in the 1930s, and later developed by Dag Prawitz and Frederic Fitch. Because its inferences mirror natural methods of proof, it is favored by philosophers. The versions developed by Fitch are often used in introductory logic textbooks. In the philosophy of logic, the rules of natural deduction have sometimes been taken to give the meanings of the logical operators ("proof-theoretic semantics").

---

content/propositional-logic/../first-order-logic/proof-systems/tableaux.tex

8.4 Tableaux

pl:prf:tab:sec

While many derivation systems operate with arrangements of sentences, tableaux operate with signed formulas. A signed formula is a pair consisting of a truth value sign ($\mathbb{T}$ or $\mathbb{F}$) and a sentence

$$\mathbb{T}\,\varphi \text{ or } \mathbb{F}\,\varphi.$$

A tableau consists of signed formulas arranged in a downward-branching tree. It begins with a number of *assumptions* and continues with signed formulas which result from one of the signed formulas above it by applying one of the rules of inference. Each rule allows us to add one or more signed formulas to the end of a branch, or two signed formulas side by side—in this case a branch splits into two, with the two added signed formulas forming the ends of the two branches.

A rule applied to a complex signed formula results in the addition of signed formulas which are immediate sub-formulas. They come in pairs, one rule for each of the two signs. For instance, the $\wedge\mathbb{T}$ rule applies to $\mathbb{T}\,\varphi \wedge \psi$, and allows the addition of both the two signed formulas $\mathbb{T}\,\varphi$ and $\mathbb{T}\,\psi$ to the end of any branch containing $\mathbb{T}\,\varphi \wedge \psi$, and the rule $\varphi \wedge \psi\mathbb{F}$ allows a branch to be split by adding $\mathbb{F}\,\varphi$ and $\mathbb{F}\,\psi$ side-by-side. A tableau is closed if every one of its branches contains a matching pair of signed formulas $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$.

The $\vdash$ relation based on tableaux is defined as follows: $\Gamma \vdash \varphi$ iff there is some finite set $\Gamma_0 = \{\psi_1, \ldots, \psi_n\} \subseteq \Gamma$ such that there is a closed tableau for the assumptions

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$

For instance, here is a closed tableau that shows that $\vdash (\varphi \wedge \psi) \to \varphi$:

1.	$\mathbb{F}\,(\varphi \wedge \psi) \to \varphi$	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\varphi$	$\to\mathbb{T}\,2$
5.	$\mathbb{T}\,\psi$	$\to\mathbb{T}\,2$
	$\otimes$	

A set Γ is inconsistent in the tableau calculus if there is a closed tableau for assumptions

$$\{\mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$

for some $\psi_i \in \Gamma$.

Tableaux were invented in the 1950s independently by Evert Beth and Jaakko Hintikka, and simplified and popularized by Raymond Smullyan. They are very easy to use, since constructing a tableau is a very systematic procedure. Because of the systematic nature of tableaux, they also lend themselves to implementation by computer. However, a tableau is often hard to read and their connection to proofs are sometimes not easy to see. The approach is also quite general, and many different logics have tableau systems. Tableaux also help us to find structures that satisfy given (sets of) sentences: if the set is satisfiable, it won't have a closed tableau, i.e., any tableau will have an open branch. The satisfying structure can be "read off" an open branch, provided every rule it is possible to apply has been applied on that branch. There is also a very close connection to the sequent calculus: essentially, a closed tableau is a condensed derivation in the sequent calculus, written upside-down.

---

content/propositional-logic/../first-order-logic/proof-systems/axiomatic-deduction.

8.5 Axiomatic Derivations

pl:prf:axd:sec

Axiomatic derivations are the oldest and simplest logical derivation systems. Its derivations are simply sequences of sentences. A sequence of sentences counts as a correct derivation if every sentence φ in it satisfies one of the following conditions:

1. φ is an axiom, or
2. φ is an element of a given set Γ of sentences, or
3. φ is justified by a rule of inference.

To be an axiom, φ has to have the form of one of a number of fixed sentence schemas. There are many sets of axiom schemas that provide a satisfactory (sound and complete) derivation system for first-order logic. Some are organized according to the connectives they govern, e.g., the schemas

$$\varphi \to (\psi \to \varphi) \qquad \psi \to (\psi \vee \chi) \qquad (\psi \wedge \chi) \to \psi$$

are common axioms that govern $\to$, $\vee$ and $\wedge$. Some axiom systems aim at a minimal number of axioms. Depending on the connectives that are taken as primitives, it is even possible to find axiom systems that consist of a single axiom.

A rule of inference is a conditional statement that gives a sufficient condition for a sentence in a derivation to be justified. Modus ponens is one very common such rule: it says that if φ and $\varphi \to \psi$ are already justified, then ψ is justified. This means that a line in a derivation containing the sentence ψ is justified, provided that both φ and $\varphi \to \psi$ (for some sentence φ) appear in the derivation before ψ.

The $\vdash$ relation based on axiomatic derivations is defined as follows: $\Gamma \vdash \varphi$ iff there is a derivation with the sentence φ as its last formula (and Γ is taken as the set of sentences in that derivation which are justified by (2) above). φ is a theorem if φ has a derivation where Γ is empty, i.e., every sentence in the derivation is justified either by (1) or (3). For instance, here is a derivation that shows that $\vdash \varphi \to (\psi \to (\psi \vee \varphi))$:

1. $\psi \to (\psi \vee \varphi)$
2. $(\psi \to (\psi \vee \varphi)) \to (\varphi \to (\psi \to (\psi \vee \varphi)))$
3. $\varphi \to (\psi \to (\psi \vee \varphi))$

The sentence on line 1 is of the form of the axiom $\varphi \to (\varphi \vee \psi)$ (with the roles of φ and ψ reversed). The sentence on line 2 is of the form of the axiom $\varphi \to (\psi \to \varphi)$. Thus, both lines are justified. Line 3 is justified by modus ponens: if we abbreviate it as θ, then line 2 has the form $\chi \to \theta$, where χ is $\psi \to (\psi \vee \varphi)$, i.e., line 1.

A set Γ is inconsistent if $\Gamma \vdash \bot$. A complete axiom system will also prove that $\bot \to \varphi$ for any φ, and so if Γ is inconsistent, then $\Gamma \vdash \varphi$ for any φ.

Systems of axiomatic derivations for logic were first given by Gottlob Frege in his 1879 *Begriffsschrift*, which for this reason is often considered the first work of modern logic. They were perfected in Alfred North Whitehead and Bertrand Russell's *Principia Mathematica* and by David Hilbert and his students in the 1920s. They are thus often called "Frege systems" or "Hilbert systems." They are very versatile in that it is often easy to find an axiomatic system for a logic. Because derivations have a very simple structure and only one or two inference rules, it is also relatively easy to prove things *about* them.

However, they are very hard to use in practice, i.e., it is difficult to find and write proofs.

Chapter 9

The Sequent Calculus

This chapter presents Gentzen's standard sequent calculus LK for classical first-order logic. It could use more examples and exercises. To include or exclude material relevant to the sequent calculus as a proof system, use the "prfLK" tag.

---

content/propositional-logic/../first-order-logic/sequent-calculus/rules-and-proofs.

9.1 Rules and Derivations

pl:seq:rul:sec

For the following, let $\Gamma, \Delta, \Pi, \Lambda$ represent finite sequences of sentences.

Definition 9.1 (Sequent). A *sequent* is an expression of the form

$$\Gamma \Rightarrow \Delta$$

where Γ and Δ are finite (possibly empty) sequences of sentences of the language $\mathcal{L}$. Γ is called the *antecedent*, while Δ is the *succedent*.

explanation

The intuitive idea behind a sequent is: if all of the sentences in the antecedent hold, then at least one of the sentences in the succedent holds. That is, if $\Gamma = \langle \varphi_1, \ldots, \varphi_m \rangle$ and $\Delta = \langle \psi_1, \ldots, \psi_n \rangle$, then $\Gamma \Rightarrow \Delta$ holds iff

$$(\varphi_1 \wedge \cdots \wedge \varphi_m) \to (\psi_1 \vee \cdots \vee \psi_n)$$

holds. There are two special cases: where Γ is empty and when Δ is empty. When Γ is empty, i.e., $m = 0$, $\Rightarrow \Delta$ holds iff $\psi_1 \vee \cdots \vee \psi_n$ holds. When Δ is empty, i.e., $n = 0$, $\Gamma \Rightarrow$ holds iff $\neg(\varphi_1 \wedge \cdots \wedge \varphi_m)$ does. We say a sequent is valid iff the corresponding sentence is valid.

If Γ is a sequence of sentences, we write Γ, φ for the result of appending φ to the right end of Γ (and φ, Γ for the result of appending φ to the left end of Γ). If Δ is a sequence of sentences also, then Γ, Δ is the concatenation of the two sequences.

Definition 9.2 (Initial Sequent). An *initial sequent* is a sequent of one of the following forms:

1. $\varphi \Rightarrow \varphi$
2. $\Rightarrow \top$
3. $\bot \Rightarrow$

for any sentence φ in the language.

Derivations in the sequent calculus are certain trees of sequents, where the topmost sequents are initial sequents, and if a sequent stands below one or two other sequents, it must follow correctly by a rule of inference. The rules for **LK** are divided into two main types: *logical* rules and *structural* rules. The logical rules are named for the main operator of the sentence containing φ and/or ψ in the lower sequent. Each one comes in two versions, one for inferring a sequent with the sentence containing the logical operator on the left, and one with the sentence on the right.

---

content/propositional-logic/../first-order-logic/sequent-calculus/propositional-rules.tex

9.2 Propositional Rules

pl:seq:prl:sec

Rules for $\neg$

$$\frac{\Gamma \Rightarrow \Delta, \varphi}{\neg\varphi, \Gamma \Rightarrow \Delta}\,\neg\mathrm{L} \qquad \frac{\varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg\varphi}\,\neg\mathrm{R}$$

Rules for $\wedge$

$$\frac{\varphi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\,\wedge\mathrm{L} \qquad \frac{\psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\,\wedge\mathrm{L} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi \qquad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\,\wedge\mathrm{R}$$

Rules for $\vee$

$$\frac{\varphi, \Gamma \Rightarrow \Delta \qquad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta} \vee\text{L} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee\text{R} \quad \frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee\text{R}$$

Rules for $\rightarrow$

$$\frac{\Gamma \Rightarrow \Delta, \varphi \qquad \psi, \Pi \Rightarrow \Lambda}{\varphi \rightarrow \psi, \Gamma, \Pi \Rightarrow \Delta, \Lambda} \rightarrow\text{L} \qquad \frac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi} \rightarrow\text{R}$$

---

content/propositional-logic/../first-order-logic/sequent-calculus/structural-rules.

9.3 Structural Rules

We also need a few rules that allow us to rearrange sentences in the left and right side of a sequent. Since the logical rules require that the sentences in the premise which the rule acts upon stand either to the far left or to the far right, we need an "exchange" rule that allows us to move sentences to the right position. It's also important sometimes to be able to combine two identical sentences into one, and to add a sentence on either side.

pl:seq:srl:sec

Weakening

$$\frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} \text{WL} \qquad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi} \text{WR}$$

Contraction

$$\frac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} \text{CL} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi} \text{CR}$$

Exchange

$$\frac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta}\,\text{XL} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi, \psi, \Lambda}{\Gamma \Rightarrow \Delta, \psi, \varphi, \Lambda}\,\text{XR}$$

A series of weakening, contraction, and exchange inferences will often be indicated by double inference lines.

The following rule, called "cut," is not strictly speaking necessary, but makes it a lot easier to reuse and combine derivations.

$$\frac{\Gamma \Rightarrow \Delta, \varphi \qquad \varphi, \Pi \Rightarrow \Lambda}{\Gamma, \Pi \Rightarrow \Delta, \Lambda}\,\text{Cut}$$

---

content/propositional-logic/../first-order-logic/sequent-calculus/derivations.tex

9.4 Derivations

pl:seq:der:sec

explanation

We've said what an initial sequent looks like, and we've given the rules of inference. Derivations in the sequent calculus are inductively generated from these: each derivation either is an initial sequent on its own, or consists of one or two derivations followed by an inference.

Definition 9.3 (LK derivation). An **LK**-*derivation* of a sequent S is a finite tree of sequents satisfying the following conditions:

1. The topmost sequents of the tree are initial sequents.
2. The bottommost sequent of the tree is S.
3. Every sequent in the tree except S is a premise of a correct application of an inference rule whose conclusion stands directly below that sequent in the tree.

We then say that S is the *end-sequent* of the derivation and that S is *derivable in* **LK** (or **LK**-derivable).

Example 9.4. Every initial sequent, e.g., $\chi \Rightarrow \chi$ is a derivation. We can obtain a new derivation from this by applying, say, the WL rule,

$$\frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\,\text{WL}$$

The rule, however, is meant to be general: we can replace the φ in the rule with any sentence, e.g., also with θ. If the premise matches our initial sequent $\chi \Rightarrow \chi$, that means that both Γ and Δ are just χ, and the conclusion would then be $\theta, \chi \Rightarrow \chi$. So, the following is a derivation:

$$\dfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\,\mathrm{WL}$$

We can now apply another rule, say XL, which allows us to switch two sentences on the left. So, the following is also a correct derivation:

$$\dfrac{\dfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\,\mathrm{WL}}{\chi, \theta \Rightarrow \chi}\,\mathrm{XL}$$

In this application of the rule, which was given as

$$\dfrac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta,}\,\mathrm{XL}$$

both Γ and Π were empty, Δ is χ, and the roles of φ and ψ are played by θ and χ, respectively. In much the same way, we also see that

$$\dfrac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta}\,\mathrm{WL}$$

is a derivation. Now we can take these two derivations, and combine them using $\wedge$R. That rule was

$$\dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\,\wedge\mathrm{R}$$

In our case, the premises must match the last sequents of the derivations ending in the premises. That means that Γ is χ, θ, Δ is empty, φ is χ and ψ is θ. So the conclusion, if the inference should be correct, is $\chi, \theta \Rightarrow \chi \wedge \theta$.

$$\dfrac{\dfrac{\dfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\,\mathrm{WL}}{\chi, \theta \Rightarrow \chi}\,\mathrm{XL} \qquad \dfrac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta}\,\mathrm{WL}}{\chi, \theta \Rightarrow \chi \wedge \theta}\,\wedge\mathrm{R}$$

Of course, we can also reverse the premises, then φ would be θ and ψ would be χ.

$$\dfrac{\dfrac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta}\,\mathrm{WL} \qquad \dfrac{\dfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\,\mathrm{WL}}{\chi, \theta \Rightarrow \chi}\,\mathrm{XL}}{\chi, \theta \Rightarrow \theta \wedge \chi}\,\wedge\mathrm{R}$$

---

content/propositional-logic/../first-order-logic/sequent-calculus/proving-things.te

9.5 Examples of Derivations

pl:seq:pro:sec

Example 9.5. Give an **LK**-derivation for the sequent $\varphi \wedge \psi \Rightarrow \varphi$.

We begin by writing the desired end-sequent at the bottom of the derivation.

$$\frac{}{\varphi \wedge \psi \Rightarrow \varphi}$$

Next, we need to figure out what kind of inference could have a lower sequent of this form. This could be a structural rule, but it is a good idea to start by looking for a logical rule. The only logical connective occurring in the lower sequent is $\wedge$, so we're looking for an $\wedge$ rule, and since the $\wedge$ symbol occurs in the antecedent, we're looking at the $\wedge$L rule.

$$\frac{}{\varphi \wedge \psi \Rightarrow \varphi}\,\wedge\text{L}$$

There are two options for what could have been the upper sequent of the $\wedge$L inference: we could have an upper sequent of $\varphi \Rightarrow \varphi$, or of $\psi \Rightarrow \varphi$. Clearly, $\varphi \Rightarrow \varphi$ is an initial sequent (which is a good thing), while $\psi \Rightarrow \varphi$ is not derivable in general. We fill in the upper sequent:

$$\frac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi}\,\wedge\text{L}$$

We now have a correct **LK**-derivation of the sequent $\varphi \wedge \psi \Rightarrow \varphi$.

Example 9.6. Give an **LK**-derivation for the sequent $\neg\varphi \vee \psi \Rightarrow \varphi \to \psi$.

Begin by writing the desired end-sequent at the bottom of the derivation.

$$\frac{}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}$$

To find a logical rule that could give us this end-sequent, we look at the logical connectives in the end-sequent: $\neg$, $\vee$, and $\to$. We only care at the moment about $\vee$ and $\to$ because they are main operators of sentences in the end-sequent, while $\neg$ is inside the scope of another connective, so we will take care of it later. Our options for logical rules for the final inference are therefore the $\vee$L rule and the $\to$R rule. We could pick either rule, really, but let's pick the $\to$R rule (if for no reason other than it allows us to put off splitting into two branches). According to the form of $\to$R inferences which can yield the lower sequent, this must look like:

$$\frac{\dfrac{}{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}\,\to\text{R}$$

If we move $\neg\varphi \vee \psi$ to the outside of the antecedent, we can apply the $\vee$L rule. According to the schema, this must split into two upper sequents as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\varphi, \varphi \Rightarrow \psi} \quad \dfrac{}{\psi, \varphi \Rightarrow \psi}}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi}\vee\mathrm{L}}{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}\mathrm{XR}}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}\to\mathrm{R}$$

Remember that we are trying to wind our way up to initial sequents; we seem to be pretty close! The right branch is just one weakening and one exchange away from an initial sequent and then it is done:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\varphi, \varphi \Rightarrow \psi} \quad \dfrac{\dfrac{\dfrac{}{\psi \Rightarrow \psi}\mathrm{WL}}{\varphi, \psi \Rightarrow \psi}\mathrm{XL}}{\psi, \varphi \Rightarrow \psi}}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi}\vee\mathrm{L}}{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}\mathrm{XR}}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}\to\mathrm{R}$$

Now looking at the left branch, the only logical connective in any sentence is the $\neg$ symbol in the antecedent sentences, so we're looking at an instance of the $\neg$L rule.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \Rightarrow \psi, \varphi}\neg\mathrm{L}}{\neg\varphi, \varphi \Rightarrow \psi} \quad \dfrac{\dfrac{\dfrac{}{\psi \Rightarrow \psi}\mathrm{WL}}{\varphi, \psi \Rightarrow \psi}\mathrm{XL}}{\psi, \varphi \Rightarrow \psi}}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi}\vee\mathrm{L}}{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}\mathrm{XR}}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}\to\mathrm{R}$$

Similarly to how we finished off the right branch, we are just one weakening and one exchange away from finishing off this left branch as well.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \Rightarrow \varphi}\mathrm{WR}}{\varphi \Rightarrow \varphi, \psi}\mathrm{XR}}{\varphi \Rightarrow \psi, \varphi}\neg\mathrm{L}}{\neg\varphi, \varphi \Rightarrow \psi} \quad \dfrac{\dfrac{\dfrac{}{\psi \Rightarrow \psi}\mathrm{WL}}{\varphi, \psi \Rightarrow \psi}\mathrm{XL}}{\psi, \varphi \Rightarrow \psi}}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi}\vee\mathrm{L}}{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}\mathrm{XR}}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}\to\mathrm{R}$$

Example 9.7. Give an **LK**-derivation of the sequent $\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)$

Using the techniques from above, we start by writing the desired end-sequent at the bottom.

$$\dfrac{}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)}$$

The available main connectives of sentences in the end-sequent are the $\vee$ symbol and the $\neg$ symbol. It would work to apply either the $\vee$L or the $\neg$R rule here, but we start with the $\neg$R rule because it avoids splitting up into two branches for a moment:

$$\dfrac{\dfrac{}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow}}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)}\,\neg\mathrm{R}$$

Now we have a choice of whether to look at the $\wedge$L or the $\vee$L rule. Let's see what happens when we apply the $\wedge$L rule: we have a choice to start with either the sequent $\varphi, \neg\varphi \vee \psi \Rightarrow$ or the sequent $\psi, \neg\varphi \vee \psi \Rightarrow$. Since the derivation is symmetric with regards to φ and ψ, let's go with the former:

$$\dfrac{\dfrac{\dfrac{}{\varphi, \neg\varphi \vee \neg\psi \Rightarrow}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow}\,\wedge\mathrm{L}}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)}\,\neg\mathrm{R}$$

Continuing to fill in the derivation, we see that we run into a problem:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \Rightarrow \varphi}}{\neg\varphi, \varphi \Rightarrow}\,\neg\mathrm{L} \quad \dfrac{\dfrac{}{\varphi \Rightarrow \psi}\,?}{\neg\psi, \varphi \Rightarrow}\,\neg\mathrm{L}}{\neg\varphi \vee \neg\psi, \varphi \Rightarrow}\,\vee\mathrm{L}}{\varphi, \neg\varphi \vee \neg\psi \Rightarrow}\,\mathrm{XL}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow}\,\wedge\mathrm{L}}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)}\,\neg\mathrm{R}$$

The top of the right branch cannot be reduced any further, and it cannot be brought by way of structural inferences to an initial sequent, so this is not the right path to take. So clearly, it was a mistake to apply the $\wedge$L rule above. Going back to what we had before and carrying out the $\vee$L rule instead, we get

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\varphi, \varphi \wedge \psi \Rightarrow} \quad \dfrac{}{\neg\psi, \varphi \wedge \psi \Rightarrow}}{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \Rightarrow}\,\vee\mathrm{L}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow}\,\mathrm{XL}}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)}\,\neg\mathrm{R}$$

Completing each branch as we've done before, we get

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \Rightarrow \varphi}}{\varphi \wedge \psi \Rightarrow \varphi}\,\wedge\mathrm{L}}{\neg\varphi, \varphi \wedge \psi \Rightarrow}\,\neg\mathrm{L} \quad \dfrac{\dfrac{\dfrac{}{\psi \Rightarrow \psi}}{\varphi \wedge \psi \Rightarrow \psi}\,\wedge\mathrm{L}}{\neg\psi, \varphi \wedge \psi \Rightarrow}\,\neg\mathrm{L}}{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \Rightarrow}\,\vee\mathrm{L}}{\varphi, \neg\varphi \vee \neg\psi \Rightarrow}\,\mathrm{XL}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow}\,\wedge\mathrm{L}}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)}\,\neg\mathrm{R}$$

(We could have carried out the $\wedge$ rules lower than the $\neg$ rules in these steps and still obtained a correct derivation).

Example 9.8. So far we haven't used the contraction rule, but it is sometimes required. Here's an example where that happens. Suppose we want to prove $\Rightarrow \varphi \vee \neg\varphi$. Applying $\vee$R backwards would give us one of these two derivations:

$$\dfrac{\dfrac{}{\Rightarrow \varphi}}{\Rightarrow \varphi \vee \neg\varphi}\vee\mathrm{R} \qquad \dfrac{\dfrac{\dfrac{}{\varphi \Rightarrow}}{\Rightarrow \neg\varphi}\neg\mathrm{R}}{\Rightarrow \varphi \vee \neg\varphi}\vee\mathrm{R}$$

Neither of these of course ends in an initial sequent. The trick is to realize that the contraction rule allows us to combine two copies of a sentence into one—and when we're searching for a proof, i.e., going from bottom to top, we can keep a copy of $\varphi \vee \neg\varphi$ in the premise, e.g.,

$$\dfrac{\dfrac{\dfrac{}{\Rightarrow \varphi \vee \neg\varphi, \varphi}}{\Rightarrow \varphi \vee \neg\varphi, \varphi \vee \neg\varphi}\vee\mathrm{R}}{\Rightarrow \varphi \vee \neg\varphi}\mathrm{C}\mathrm{R}$$

Now we can apply $\vee$R a second time, and also get $\neg\varphi$, which leads to a complete derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg\varphi}\neg\mathrm{R}}{\Rightarrow \varphi, \varphi \vee \neg\varphi}\vee\mathrm{R}}{\Rightarrow \varphi \vee \neg\varphi, \varphi}\mathrm{X}\mathrm{R}}{\Rightarrow \varphi \vee \neg\varphi, \varphi \vee \neg\varphi}\vee\mathrm{R}}{\Rightarrow \varphi \vee \neg\varphi}\mathrm{C}\mathrm{R}$$

Problem 9.1. Give derivations of the following sequents:

1. $\varphi \wedge (\psi \wedge \chi) \Rightarrow (\varphi \wedge \psi) \wedge \chi$.
2. $\varphi \vee (\psi \vee \chi) \Rightarrow (\varphi \vee \psi) \vee \chi$.
3. $\varphi \to (\psi \to \chi) \Rightarrow \psi \to (\varphi \to \chi)$.
4. $\varphi \Rightarrow \neg\neg\varphi$.

Problem 9.2. Give derivations of the following sequents:

1. $(\varphi \vee \psi) \to \chi \Rightarrow \varphi \to \chi$.
2. $(\varphi \to \chi) \wedge (\psi \to \chi) \Rightarrow (\varphi \vee \psi) \to \chi$.
3. $\Rightarrow \neg(\varphi \wedge \neg\varphi)$.
4. $\psi \to \varphi \Rightarrow \neg\varphi \to \neg\psi$.
5. $\Rightarrow (\varphi \to \neg\varphi) \to \neg\varphi$.
6. $\Rightarrow \neg(\varphi \to \psi) \to \neg\psi$.
7. $\varphi \to \chi \Rightarrow \neg(\varphi \wedge \neg\chi)$.
8. $\varphi \wedge \neg\chi \Rightarrow \neg(\varphi \to \chi)$.
9. $\varphi \vee \psi, \neg\psi \Rightarrow \varphi$.

10. $\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)$.
11. $\Rightarrow (\neg\varphi \wedge \neg\psi) \to \neg(\varphi \vee \psi)$.
12. $\Rightarrow \neg(\varphi \vee \psi) \to (\neg\varphi \wedge \neg\psi)$.

Problem 9.3. Give derivations of the following sequents:

1. $\neg(\varphi \to \psi) \Rightarrow \varphi$.
2. $\neg(\varphi \wedge \psi) \Rightarrow \neg\varphi \vee \neg\psi$.
3. $\varphi \to \psi \Rightarrow \neg\varphi \vee \psi$.
4. $\Rightarrow \neg\neg\varphi \to \varphi$.
5. $\varphi \to \psi, \neg\varphi \to \psi \Rightarrow \psi$.
6. $(\varphi \wedge \psi) \to \chi \Rightarrow (\varphi \to \chi) \vee (\psi \to \chi)$.
7. $(\varphi \to \psi) \to \varphi \Rightarrow \varphi$.
8. $\Rightarrow (\varphi \to \psi) \vee (\psi \to \chi)$.

(These all require the CR rule.)

This section collects the definitions of the provability relation and consistency for natural deduction.

---

content/propositional-logic/../first-order-logic/sequent-calculus/proof-theoretic-notions.tex

9.6 Proof-Theoretic Notions

pl:seq:ptn:sec

Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain sequents. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorem*.

explanation

Definition 9.9 (Theorems). A sentence φ is a *theorem* if there is a derivation in **LK** of the sequent $\Rightarrow \varphi$. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 9.10 (Derivability). A sentence φ is *derivable from* a set of sentences Γ, $\Gamma \vdash \varphi$, iff there is a finite subset $\Gamma_0 \subseteq \Gamma$ and a sequence Γ_0' of the sentences in Γ_0 such that **LK** derives $\Gamma_0' \Rightarrow \varphi$. If φ is not derivable from Γ we write $\Gamma \nvdash \varphi$.

Because of the contraction, weakening, and exchange rules, the order and number of sentences in Γ_0' does not matter: if a sequent $\Gamma_0' \Rightarrow \varphi$ is derivable, then so is $\Gamma_0'' \Rightarrow \varphi$ for any Γ_0'' that contains the same sentences as Γ_0'. For instance, if $\Gamma_0 = \{\psi, \chi\}$ then both $\Gamma_0' = \langle \psi, \psi, \chi \rangle$ and $\Gamma_0'' = \langle \chi, \chi, \psi \rangle$ are sequences containing just the sentences in Γ_0. If a sequent containing one is derivable, so is the other, e.g.:

$$\dfrac{\dfrac{\dfrac{\vdots}{\psi, \psi, \chi \Rightarrow \varphi}\,\mathrm{CL}}{\psi, \chi \Rightarrow \varphi}\,\mathrm{XL}}{\chi, \psi \Rightarrow \varphi}\,\mathrm{WL}$$

From now on we'll say that if Γ_0 is a finite set of sentences then $\Gamma_0 \Rightarrow \varphi$ is any sequent where the antecedent is a sequence of sentences in Γ_0 and tacitly include contractions, exchanges, and weakenings if necessary.

Definition 9.11 (Consistency). A set of sentences Γ is *inconsistent* iff there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow$. If Γ is not inconsistent, i.e., if for every finite $\Gamma_0 \subseteq \Gamma$, **LK** does not derive $\Gamma_0 \Rightarrow$, we say it is *consistent*.

Proposition 9.12 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.* *pl:seq:ptn: prop:reflexivity*

Proof. The initial sequent $\varphi \Rightarrow \varphi$ is derivable, and $\{\varphi\} \subseteq \Gamma$. □

Proposition 9.13 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.* *pl:seq:ptn: prop:monotonicity*

Proof. Suppose $\Gamma \vdash \varphi$, i.e., there is a finite $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \Rightarrow \varphi$ is derivable. Since $\Gamma \subseteq \Delta$, then Γ_0 is also a finite subset of Δ. The derivation of $\Gamma_0 \Rightarrow \varphi$ thus also shows $\Delta \vdash \varphi$. □

Proposition 9.14 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.* *pl:seq:ptn: prop:transitivity*

Proof. If $\Gamma \vdash \varphi$, there is a finite $\Gamma_0 \subseteq \Gamma$ and a derivation π_0 of $\Gamma_0 \Rightarrow \varphi$. If $\{\varphi\} \cup \Delta \vdash \psi$, then for some finite subset $\Delta_0 \subseteq \Delta$, there is a derivation π_1 of $\varphi, \Delta_0 \Rightarrow \psi$. Consider the following derivation:

$$\dfrac{\begin{array}{cc}\vdots\,\pi_0 & \vdots\,\pi_1 \\ \Gamma_0 \Rightarrow \varphi & \varphi, \Delta_0 \Rightarrow \psi\end{array}}{\Gamma_0, \Delta_0 \Rightarrow \psi}\,\mathrm{Cut}$$

Since $\Gamma_0 \cup \Delta_0 \subseteq \Gamma \cup \Delta$, this shows $\Gamma \cup \Delta \vdash \psi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \ldots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

pl:seq:ptn: prop:incons

Proposition 9.15. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every sentence φ.*

Proof. Exercise. □

Problem 9.4. Prove Proposition 19.16

pl:seq:ptn: prop:proves-compact

Proposition 9.16 (Compactness).

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof. 1. If $\Gamma \vdash \varphi$, then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that the sequent $\Gamma_0 \Rightarrow \varphi$ has a derivation. Consequently, $\Gamma_0 \vdash \varphi$.

2. If Γ is inconsistent, there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow$. But then Γ_0 is a finite subset of Γ that is inconsistent. □

---

content/propositional-logic/../first-order-logic/sequent-calculus/provability-consistency.tex

9.7 Derivability and Consistency

pl:seq:prv: sec

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem.

pl:seq:prv: prop:provability-contr

Proposition 9.17. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.*

Proof. There are finite Γ_0 and $\Gamma_1 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow \varphi$ and $\varphi, \Gamma_1 \Rightarrow$. Let the **LK**-derivation of $\Gamma_0 \Rightarrow \varphi$ be π_0 and the **LK**-derivation of $\Gamma_1, \varphi \Rightarrow$ be π_1. We can then derive

$$\dfrac{\begin{array}{cc}\vdots\,\pi_0 & \vdots\,\pi_1 \\ \Gamma_0 \Rightarrow \varphi & \varphi, \Gamma_1 \Rightarrow\end{array}}{\Gamma_0, \Gamma_1 \Rightarrow}\ \text{Cut}$$

Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$, hence Γ is inconsistent. □

pl:seq:prv: prop:prov-incons

Proposition 9.18. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.*

Proof. First suppose $\Gamma \vdash \varphi$, i.e., there is a derivation π_0 of $\Gamma \Rightarrow \varphi$. By adding a $\neg$L rule, we obtain a derivation of $\neg\varphi, \Gamma \Rightarrow$, i.e., $\Gamma \cup \{\neg\varphi\}$ is inconsistent.

If $\Gamma \cup \{\neg\varphi\}$ is inconsistent, there is a derivation π_1 of $\neg\varphi, \Gamma \Rightarrow$. The following is a derivation of $\Gamma \Rightarrow \varphi$:

$$\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg\varphi}\,\neg\mathrm{R} \qquad \begin{array}{c}\vdots\,\pi_1\\ \neg\varphi, \Gamma \Rightarrow\end{array}}{\Gamma \Rightarrow \varphi}\,\mathrm{Cut}$$

□

Problem 9.5. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

Proposition 9.19. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

pl:seq:prv: prop:explicit-inc

Proof. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there is a derivation π of a sequent $\Gamma_0 \Rightarrow \varphi$. The sequent $\neg\varphi, \Gamma_0 \Rightarrow$ is also derivable:

$$\dfrac{\begin{array}{c}\vdots\,\pi\\ \Gamma_0 \Rightarrow \varphi\end{array} \qquad \dfrac{\dfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow}\,\neg\mathrm{L}}{\varphi, \neg\varphi \Rightarrow}\,\mathrm{XL}}{\Gamma_0, \neg\varphi \Rightarrow}\,\mathrm{Cut}$$

Since $\neg\varphi \in \Gamma$ and $\Gamma_0 \subseteq \Gamma$, this shows that Γ is inconsistent. □

Proposition 9.20. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

pl:seq:prv: prop:provability-exhaustive

Proof. There are finite sets $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$ and **LK**-derivations π_0 and π_1 of $\varphi, \Gamma_0 \Rightarrow$ and $\neg\varphi, \Gamma_1 \Rightarrow$, respectively. We can then derive

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\pi_0\\ \varphi, \Gamma_0 \Rightarrow\end{array}}{\Gamma_0 \Rightarrow \neg\varphi}\,\neg\mathrm{R} \qquad \begin{array}{c}\vdots\,\pi_1\\ \neg\varphi, \Gamma_1 \Rightarrow\end{array}}{\Gamma_0, \Gamma_1 \Rightarrow}\,\mathrm{Cut}$$

Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$. Hence Γ is inconsistent. □

---

content/propositional-logic/../first-order-logic/sequent-calculus/provability-propo

9.8 Derivability and the Propositional Connectives

explanation

We establish that the derivability relation $\vdash$ of the sequent calculus is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem.

pl:seq:ppr: sec

Proposition 9.21.

pl:seq:ppr: prop:provability-land

1. *Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$.*

pl:seq:ppr: prop:provability-land-left

pl:seq:ppr: prop:provability-land-right

2. $\varphi, \psi \vdash \varphi \wedge \psi$.

Proof. 1. Both sequents $\varphi \wedge \psi \Rightarrow \varphi$ and $\varphi \wedge \psi \Rightarrow \psi$ are derivable:

$$\frac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi} \wedge\mathrm{L} \qquad \frac{\psi \Rightarrow \psi}{\varphi \wedge \psi \Rightarrow \psi} \wedge\mathrm{L}$$

2. Here is a derivation of the sequent $\varphi, \psi \Rightarrow \varphi \wedge \psi$:

$$\frac{\varphi \Rightarrow \varphi \qquad \psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \varphi \wedge \psi} \wedge\mathrm{R}$$

□

pl:seq:ppr: prop:provability-lor

Proposition 9.22.

1. *$\varphi \vee \psi, \neg\varphi, \neg\psi$ is inconsistent.*
2. *Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.*

Proof. 1. We give a derivation of the sequent $\varphi \vee \psi, \neg\varphi, \neg\psi \Rightarrow$:

$$\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow}\neg\mathrm{L}}{\varphi, \neg\varphi, \neg\psi \Rightarrow} \qquad \dfrac{\dfrac{\psi \Rightarrow \psi}{\neg\psi, \psi \Rightarrow}\neg\mathrm{L}}{\psi, \neg\varphi, \neg\psi \Rightarrow}}{\varphi \vee \psi, \neg\varphi, \neg\psi \Rightarrow} \vee\mathrm{L}$$

(Recall that double inference lines indicate several weakening, contraction, and exchange inferences.)

2. Both sequents $\varphi \Rightarrow \varphi \vee \psi$ and $\psi \Rightarrow \varphi \vee \psi$ have derivations:

$$\frac{\varphi \Rightarrow \varphi}{\varphi \Rightarrow \varphi \vee \psi} \vee\mathrm{R} \qquad \frac{\psi \Rightarrow \psi}{\psi \Rightarrow \varphi \vee \psi} \vee\mathrm{R}$$

□

pl:seq:ppr: prop:provability-lif

Proposition 9.23.

pl:seq:ppr: prop:provability-lif-left

1. *$\varphi, \varphi \to \psi \vdash \psi$.*

pl:seq:ppr: prop:provability-lif-right

2. *Both $\neg\varphi \vdash \varphi \to \psi$ and $\psi \vdash \varphi \to \psi$.*

Proof. 1. The sequent $\varphi \to \psi, \varphi \Rightarrow \psi$ is derivable:

$$\frac{\varphi \Rightarrow \varphi \qquad \psi \Rightarrow \psi}{\varphi \to \psi, \varphi \Rightarrow \psi} \to\mathrm{L}$$

2. Both sequents $\neg\varphi \Rightarrow \varphi \to \psi$ and $\psi \Rightarrow \varphi \to \psi$ are derivable:

$$\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow}\neg\mathrm{L}}{\varphi, \neg\varphi \Rightarrow}\mathrm{XL}}{\varphi, \neg\varphi \Rightarrow \psi}\mathrm{WR}}{\neg\varphi \Rightarrow \varphi \to \psi}\to\mathrm{R} \qquad \dfrac{\dfrac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi}\mathrm{WL}}{\psi \Rightarrow \varphi \to \psi}\to\mathrm{R}$$ □

---

content/propositional-logic/../first-order-logic/sequent-calculus/soundness.tex

9.9 Soundness

explanation A derivation system, such as the sequent calculus, is *sound* if it cannot derive things that do not actually hold. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that pl:seq:sou:sec

1. every derivable φ is a tautology;
2. if a sentence is derivable from some others, it is also a consequence of them;
3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Because all these proof-theoretic properties are defined via derivability in the sequent calculus of certain sequents, proving (1)–(3) above requires proving something about the semantic properties of derivable sequents. We will first define what it means for a sequent to be *valid*, and then show that every derivable sequent is valid. (1)–(3) then follow as corollaries from this result.

Definition 9.24. A valuation $\mathfrak{v}$ *satisfies* a sequent $\Gamma \Rightarrow \Delta$ iff either $\mathfrak{v} \nvDash \varphi$ for some $\varphi \in \Gamma$ or $\mathfrak{v} \vDash \varphi$ for some $\varphi \in \Delta$.

A sequent is *valid* iff every valuation $\mathfrak{v}$ satisfies it.

Theorem 9.25 (Soundness). *If* **LK** *derives* $\Theta \Rightarrow \Xi$*, then* $\Theta \Rightarrow \Xi$ *is valid.* pl:seq:sou:thm:sequent-soundness

Proof. Let π be a derivation of $\Theta \Rightarrow \Xi$. We proceed by induction on the number of inferences n in π.

If the number of inferences is 0, then π consists only of an initial sequent. Every initial sequent $\varphi \Rightarrow \varphi$ is obviously valid, since for every $\mathfrak{v}$, either $\mathfrak{v} \nvDash \varphi$ or $\mathfrak{v} \vDash \varphi$.

If the number of inferences is greater than 0, we distinguish cases according to the type of the lowermost inference. By induction hypothesis, we can assume that the premises of that inference are valid, since the number of inferences in the derivation of any premise is smaller than n.

First, we consider the possible inferences with only one premise.

1. The last inference is a weakening. Then $\Theta \Rightarrow \Xi$ is either $\varphi, \Gamma \Rightarrow \Delta$ (if the last inference is WL) or $\Gamma \Rightarrow \Delta, \varphi$ (if it's WR), and the derivation ends in one of

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta\end{array}}{\varphi, \Gamma \Rightarrow \Delta}\,\mathrm{WL} \qquad \dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta\end{array}}{\Gamma \Rightarrow \Delta, \varphi}\,\mathrm{WR}$$

   By induction hypothesis, $\Gamma \Rightarrow \Delta$ is valid, i.e., for every valuation $\mathfrak{v}$, either there is some $\chi \in \Gamma$ such that $\mathfrak{v} \nvDash \chi$ or there is some $\chi \in \Delta$ such that $\mathfrak{v} \vDash \chi$.

   If $\mathfrak{v} \nvDash \chi$ for some $\chi \in \Gamma$, then $\chi \in \Theta$ as well since $\Theta = \varphi, \Gamma$, and so $\mathfrak{v} \nvDash \chi$ for some $\chi \in \Theta$. Similarly, if $\mathfrak{v} \vDash \chi$ for some $\chi \in \Delta$, as $\chi \in \Xi$, $\mathfrak{v} \vDash \chi$ for some $\chi \in \Xi$. Consequently, $\Theta \Rightarrow \Xi$ is valid.

2. The last inference is ¬L: Then the premise of the last inference is $\Gamma \Rightarrow \Delta, \varphi$ and the conclusion is $\neg\varphi, \Gamma \Rightarrow \Delta$, i.e., the derivation ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \varphi\end{array}}{\neg\varphi, \Gamma \Rightarrow \Delta}\,\neg\mathrm{L}$$

   and $\Theta = \neg\varphi, \Gamma$ while $\Xi = \Delta$.

   The induction hypothesis tells us that $\Gamma \Rightarrow \Delta, \varphi$ is valid, i.e., for every $\mathfrak{v}$, either (a) for some $\chi \in \Gamma$, $\mathfrak{v} \nvDash \chi$, or (b) for some $\chi \in \Delta$, $\mathfrak{v} \vDash \chi$, or (c) $\mathfrak{v} \vDash \varphi$. We want to show that $\Theta \Rightarrow \Xi$ is also valid. Let $\mathfrak{v}$ be a valuation. If (a) holds, then there is $\chi \in \Gamma$ so that $\mathfrak{v} \nvDash \chi$, but $\chi \in \Theta$ as well. If (b) holds, there is $\chi \in \Delta$ such that $\mathfrak{v} \vDash \chi$, but $\chi \in \Xi$ as well. Finally, if $\mathfrak{v} \vDash \varphi$, then $\mathfrak{v} \nvDash \neg\varphi$. Since $\neg\varphi \in \Theta$, there is $\chi \in \Theta$ such that $\mathfrak{v} \nvDash \chi$. Consequently, $\Theta \Rightarrow \Xi$ is valid.

3. The last inference is ¬R: Exercise.

4. The last inference is ∧L: There are two variants: $\varphi \wedge \psi$ may be inferred on the left from φ or from ψ on the left side of the premise. In the first case, the π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \varphi, \Gamma \Rightarrow \Delta\end{array}}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\,\wedge\mathrm{L}$$

and $\Theta = \varphi \wedge \psi, \Gamma$ while $\Xi = \Delta$. Consider a valuation $\mathfrak{v}$. Since by induction hypothesis, $\varphi, \Gamma \Rightarrow \Delta$ is valid, (a) $\mathfrak{v} \nvDash \varphi$, (b) $\mathfrak{v} \nvDash \chi$ for some $\chi \in \Gamma$, or (c) $\mathfrak{v} \vDash \chi$ for some $\chi \in \Delta$. In case (a), $\mathfrak{v} \nvDash \varphi \wedge \psi$, so there is $\chi \in \Theta$ (namely, $\varphi \wedge \psi$) such that $\mathfrak{v} \nvDash \chi$. In case (b), there is $\chi \in \Gamma$ such that $\mathfrak{v} \nvDash \chi$, and $\chi \in \Theta$ as well. In case (c), there is $\chi \in \Delta$ such that $\mathfrak{v} \vDash \chi$, and $\chi \in \Xi$ as well since $\Xi = \Delta$. So in each case, $\mathfrak{v}$ satisfies $\varphi \wedge \psi, \Gamma \Rightarrow \Delta$. Since $\mathfrak{v}$ was arbitrary, $\Gamma \Rightarrow \Delta$ is valid. The case where $\varphi \wedge \psi$ is inferred from ψ is handled the same, changing φ to ψ.

5. The last inference is $\vee$R: There are two variants: $\varphi \vee \psi$ may be inferred on the right from φ or from ψ on the right side of the premise. In the first case, π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \varphi\end{array}}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee\text{R}$$

Now $\Theta = \Gamma$ and $\Xi = \Delta, \varphi \vee \psi$. Consider a valuation $\mathfrak{v}$. Since $\Gamma \Rightarrow \Delta, \varphi$ is valid, (a) $\mathfrak{v} \vDash \varphi$, (b) $\mathfrak{v} \nvDash \chi$ for some $\chi \in \Gamma$, or (c) $\mathfrak{v} \vDash \chi$ for some $\chi \in \Delta$. In case (a), $\mathfrak{v} \vDash \varphi \vee \psi$. In case (b), there is $\chi \in \Gamma$ such that $\mathfrak{v} \nvDash \chi$. In case (c), there is $\chi \in \Delta$ such that $\mathfrak{v} \vDash \chi$. So in each case, $\mathfrak{v}$ satisfies $\Gamma \Rightarrow \Delta, \varphi \vee \psi$, i.e., $\Theta \Rightarrow \Xi$. Since $\mathfrak{v}$ was arbitrary, $\Theta \Rightarrow \Xi$ is valid. The case where $\varphi \vee \psi$ is inferred from ψ is handled the same, changing φ to ψ.

6. The last inference is $\rightarrow$R: Then π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \varphi, \Gamma \Rightarrow \Delta, \psi\end{array}}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi} \rightarrow\text{R}$$

Again, the induction hypothesis says that the premise is valid; we want to show that the conclusion is valid as well. Let $\mathfrak{v}$ be arbitrary. Since $\varphi, \Gamma \Rightarrow \Delta, \psi$ is valid, at least one of the following cases obtains: (a) $\mathfrak{v} \nvDash \varphi$, (b) $\mathfrak{v} \vDash \psi$, (c) $\mathfrak{v} \nvDash \chi$ for some $\chi \in \Gamma$, or (d) $\mathfrak{v} \vDash \chi$ for some $\chi \in \Delta$. In cases (a) and (b), $\mathfrak{v} \vDash \varphi \rightarrow \psi$ and so there is a $\chi \in \Delta, \varphi \rightarrow \psi$ such that $\mathfrak{v} \vDash \chi$. In case (c), for some $\chi \in \Gamma$, $\mathfrak{v} \nvDash \chi$. In case (d), for some $\chi \in \Delta$, $\mathfrak{v} \vDash \chi$. In each case, $\mathfrak{v}$ satisfies $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$. Since $\mathfrak{v}$ was arbitrary, $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$ is valid.

Now let's consider the possible inferences with two premises.

1. The last inference is a cut: then π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \varphi\end{array} \qquad \begin{array}{c}\vdots\\ \varphi, \Pi \Rightarrow \Lambda\end{array}}{\Gamma, \Pi \Rightarrow \Delta, \Lambda}\ \text{Cut}$$

Let $\mathfrak{v}$ be a valuation. By induction hypothesis, the premises are valid, so $\mathfrak{v}$ satisfies both premises. We distinguish two cases: (a) $\mathfrak{v} \nvDash \varphi$ and (b) $\mathfrak{v} \vDash \varphi$. In case (a), in order for $\mathfrak{v}$ to satisfy the left premise, it must satisfy $\Gamma \Rightarrow \Delta$. But then it also satisfies the conclusion. In case (b), in order for $\mathfrak{v}$ to satisfy the right premise, it must satisfy $\Pi \setminus \Lambda$. Again, $\mathfrak{v}$ satisfies the conclusion.

2. The last inference is $\wedge$R. Then π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \varphi\end{array} \qquad \begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \psi\end{array}}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\ \wedge\text{R}$$

Consider a valuation $\mathfrak{v}$. If $\mathfrak{v}$ satisfies $\Gamma \Rightarrow \Delta$, we are done. So suppose it doesn't. Since $\Gamma \Rightarrow \Delta, \varphi$ is valid by induction hypothesis, $\mathfrak{v} \vDash \varphi$. Similarly, since $\Gamma \Rightarrow \Delta, \psi$ is valid, $\mathfrak{v} \vDash \psi$. But then $\mathfrak{v} \vDash \varphi \wedge \psi$.

3. The last inference is $\vee$L: Exercise.

4. The last inference is $\rightarrow$L. Then π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \varphi\end{array} \qquad \begin{array}{c}\vdots\\ \psi, \Pi \Rightarrow \Lambda\end{array}}{\varphi \rightarrow \psi, \Gamma, \Pi \Rightarrow \Delta, \Lambda}\ \rightarrow\text{L}$$

Again, consider a valuation $\mathfrak{v}$ and suppose $\mathfrak{v}$ doesn't satisfy $\Gamma, \Pi \Rightarrow \Delta, \Lambda$. We have to show that $\mathfrak{v} \nvDash \varphi \rightarrow \psi$. If $\mathfrak{v}$ doesn't satisfy $\Gamma, \Pi \Rightarrow \Delta, \Lambda$, it satisfies neither $\Gamma \Rightarrow \Delta$ nor $\Pi \Rightarrow \Lambda$. Since, $\Gamma \Rightarrow \Delta, \varphi$ is valid, we have $\mathfrak{v} \vDash \varphi$. Since $\psi, \Pi \Rightarrow \Lambda$ is valid, we have $\mathfrak{v} \nvDash \psi$. But then $\mathfrak{v} \nvDash \varphi \rightarrow \psi$, which is what we wanted to show. □

Problem 9.6. Complete the proof of Theorem 9.25.

pl:seq:sou: cor:weak-soundness

Corollary 9.26. *If $\vdash \varphi$ then φ is a tautology.*

pl:seq:sou: cor:entailment-soundness

Corollary 9.27. *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.*

Proof. If $\Gamma \vdash \varphi$ then for some finite subset $\Gamma_0 \subseteq \Gamma$, there is a derivation of $\Gamma_0 \Rightarrow \varphi$. By Theorem 9.25, every valuation $\mathfrak{v}$ either makes some $\psi \in \Gamma_0$ false or makes φ true. Hence, if $\mathfrak{v} \vDash \Gamma$ then also $\mathfrak{v} \vDash \varphi$. □

Corollary 9.28. *If Γ is satisfiable, then it is consistent.*

pl:seq:sou:cor:consistency-soundness

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then there is a finite $\Gamma_0 \subseteq \Gamma$ and a derivation of $\Gamma_0 \Rightarrow$. By Theorem 9.25, $\Gamma_0 \Rightarrow$ is valid. In other words, for every valuation $\mathfrak{v}$, there is $\chi \in \Gamma_0$ so that $\mathfrak{v} \nvDash \chi$, and since $\Gamma_0 \subseteq \Gamma$, that χ is also in Γ. Thus, no $\mathfrak{v}$ satisfies Γ, and Γ is not satisfiable. □

Chapter 10

Natural Deduction

This chapter presents a natural deduction system in the style of Gentzen/Prawitz.

To include or exclude material relevant to natural deduction as a proof system, use the "prfND" tag.

---

content/propositional-logic/../first-order-logic/natural-deduction/rules-and-proofs

10.1 Rules and Derivations

explanation

Natural deduction systems are meant to closely parallel the informal reasoning used in mathematical proof (hence it is somewhat "natural"). Natural deduction proofs begin with assumptions. Inference rules are then applied. Assumptions are "discharged" by the ¬Intro, →Intro, and ∨Elim inference rules, and the label of the discharged assumption is placed beside the inference for clarity.

pl:ntd:rul:sec

Definition 10.1 (Assumption). An *assumption* is any sentence in the topmost position of any branch.

Derivations in natural deduction are certain trees of sentences, where the topmost sentences are assumptions, and if a sentence stands below one, two, or three other sequents, it must follow correctly by a rule of inference. The sentences at the top of the inference are called the *premises* and the sentence below the *conclusion* of the inference. The rules come in pairs, an introduction and an elimination rule for each logical operator. They introduce a logical operator in the conclusion or remove a logical operator from a premise of the rule. Some of the rules allow an assumption of a certain type to be *discharged*. To indicate which assumption is discharged by which inference, we also assign labels to both the assumption and the inference. This is indicated by writing the assumption as "$[\varphi]^n$."

It is customary to consider rules for all the logical operators $\wedge$, $\vee$, $\to$, $\neg$, and $\bot$, even if some of those are defined.

---

content/propositional-logic/../first-order-logic/natural-deduction/propositional-rules.tex

10.2 Propositional Rules

pl:ntd:prl:sec

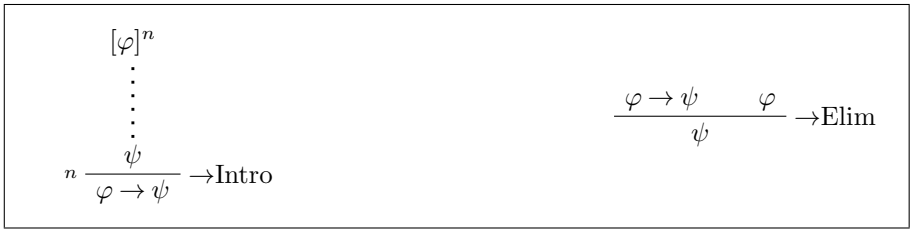
Rules for $\wedge$

$$\frac{\varphi \qquad \psi}{\varphi \wedge \psi}\,\wedge\text{Intro} \qquad \frac{\varphi \wedge \psi}{\varphi}\,\wedge\text{Elim} \qquad \frac{\varphi \wedge \psi}{\psi}\,\wedge\text{Elim}$$

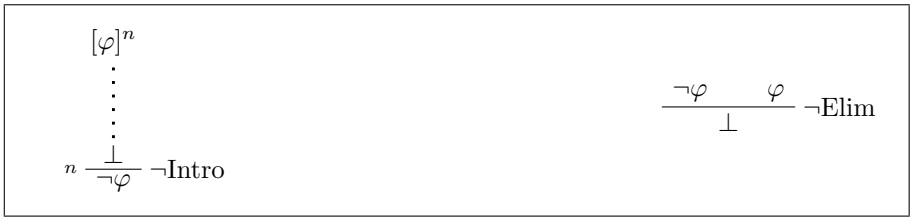
Rules for $\vee$

$$\frac{\varphi}{\varphi \vee \psi}\,\vee\text{Intro} \qquad \frac{\psi}{\varphi \vee \psi}\,\vee\text{Intro} \qquad \begin{array}{c} \varphi \vee \psi \qquad [\varphi]^n \quad [\psi]^n \\ \vdots \qquad\quad \vdots \\ \chi \qquad\quad \chi \\ \hline \chi \end{array}\,\vee\text{Elim}\ (n)$$

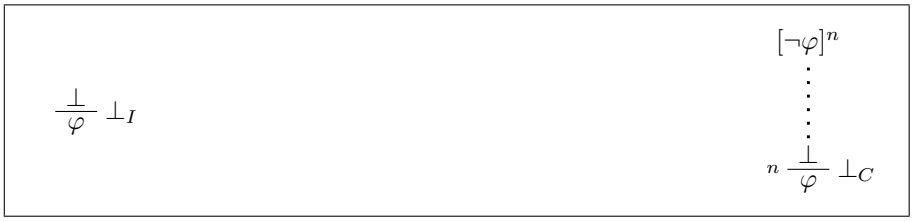
Rules for $\to$



Rules for $\neg$



Rules for $\bot$



Note that $\neg$Intro and $\bot_C$ are very similar: The difference is that $\neg$Intro derives a negated sentence $\neg\varphi$ but $\bot_C$ a positive sentence φ.

Whenever a rule indicates that some assumption may be discharged, we take this to be a permission, but not a requirement. E.g., in the $\to$Intro rule, we may discharge any number of assumptions of the form φ in the derivation of the premise ψ, including zero.

---

content/propositional-logic/../first-order-logic/natural-deduction/derivations.tex

10.3 Derivations

explanation

We've said what an assumption is, and we've given the rules of inference. Derivations in natural deduction are inductively generated from these: each derivation either is an assumption on its own, or consists of one, two, or three derivations followed by a correct inference.

pl:ntd:der:sec

Definition 10.2 (Derivation). A *derivation* of a sentence φ from assumptions Γ is a finite tree of sentences satisfying the following conditions:

1. The topmost sentences of the tree are either in Γ or are discharged by an inference in the tree.

2. The bottommost sentence of the tree is φ.

3. Every sentence in the tree except the sentence φ at the bottom is a premise of a correct application of an inference rule whose conclusion stands directly below that sentence in the tree.

We then say that φ is the *conclusion* of the derivation and Γ its undischarged assumptions.

If a derivation of φ from Γ exists, we say that φ is *derivable* from Γ, or in symbols: $\Gamma \vdash \varphi$. If there is a derivation of φ in which every assumption is discharged, we write $\vdash \varphi$.

Example 10.3. Every assumption on its own is a derivation. So, e.g., φ by itself is a derivation, and so is ψ by itself. We can obtain a new derivation from these by applying, say, the $\wedge$Intro rule,

$$\dfrac{\varphi \qquad \psi}{\varphi \wedge \psi}\,\wedge\text{Intro}$$

These rules are meant to be general: we can replace the φ and ψ in it with any sentences, e.g., by χ and θ. Then the conclusion would be $\chi \wedge \theta$, and so

$$\dfrac{\chi \qquad \theta}{\chi \wedge \theta}\,\wedge\text{Intro}$$

is a correct derivation. Of course, we can also switch the assumptions, so that θ plays the role of φ and χ that of ψ. Thus,

$$\dfrac{\theta \qquad \chi}{\theta \wedge \chi}\,\wedge\text{Intro}$$

is also a correct derivation.

We can now apply another rule, say, $\rightarrow$Intro, which allows us to conclude a conditional and allows us to discharge any assumption that is identical to the antecedent of that conditional. So both of the following would be correct derivations:

$$1\,\dfrac{\dfrac{[\chi]^1 \qquad \theta}{\chi \wedge \theta}\,\wedge\text{Intro}}{\chi \rightarrow (\chi \wedge \theta)}\,\rightarrow\text{Intro} \qquad 1\,\dfrac{\dfrac{\chi \qquad [\theta]^1}{\chi \wedge \theta}\,\wedge\text{Intro}}{\theta \rightarrow (\chi \wedge \theta)}\,\rightarrow\text{Intro}$$

They show, respectively, that $\theta \vdash \chi \rightarrow (\chi \wedge \theta)$ and $\chi \vdash \theta \rightarrow (\chi \wedge \theta)$.

Remember that discharging of assumptions is a permission, not a requirement: we don't have to discharge the assumptions. In particular, we can apply a rule even if the assumptions are not present in the derivation. For instance, the following is legal, even though there is no assumption φ to be discharged:

$$\dfrac{\dfrac{\psi}{\varphi \to \psi}\,{\scriptstyle 1}}{\to\text{Intro}}$$

---

content/propositional-logic/../first-order-logic/natural-deduction/proving-things.t

10.4 Examples of Derivations

pl:ntd:pro:sec

Example 10.4. Let's give a derivation of the sentence $(\varphi \wedge \psi) \to \varphi$.

We begin by writing the desired conclusion at the bottom of the derivation.

$$\dfrac{}{(\varphi \wedge \psi) \to \varphi}$$

Next, we need to figure out what kind of inference could result in a sentence of this form. The main operator of the conclusion is $\to$, so we'll try to arrive at the conclusion using the $\to$Intro rule. It is best to write down the assumptions involved and label the inference rules as you progress, so it is easy to see whether all assumptions have been discharged at the end of the proof.

$$\dfrac{\begin{array}{c}[\varphi \wedge \psi]^1 \\ \vdots \\ \varphi\end{array}}{(\varphi \wedge \psi) \to \varphi}\,{\scriptstyle 1}\ \to\text{Intro}$$

We now need to fill in the steps from the assumption $\varphi \wedge \psi$ to φ. Since we only have one connective to deal with, $\wedge$, we must use the $\wedge$ elim rule. This gives us the following proof:

$$\dfrac{\dfrac{\dfrac{[\varphi \wedge \psi]^1}{\varphi}\,\wedge\text{Elim}}{(\varphi \wedge \psi) \to \varphi}\,{\scriptstyle 1}\ \to\text{Intro}}{}$$

We now have a correct derivation of $(\varphi \wedge \psi) \to \varphi$.

Example 10.5. Now let's give a derivation of $(\neg\varphi \vee \psi) \to (\varphi \to \psi)$.

We begin by writing the desired conclusion at the bottom of the derivation.

$$\dfrac{}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}$$

To find a logical rule that could give us this conclusion, we look at the logical connectives in the conclusion: $\neg$, $\vee$, and $\to$. We only care at the moment about the first occurrence of $\to$ because it is the main operator of the sentence in the end-sequent, while $\neg$, $\vee$ and the second occurrence of $\to$ are inside the scope of another connective, so we will take care of those later. We therefore start with the $\to$Intro rule. A correct application must look like this:

$$\dfrac{\begin{array}{c}[\neg\varphi \vee \psi]^1 \\ \vdots \\ \varphi \to \psi\end{array}}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}\ {\scriptstyle 1}\ \to\text{Intro}$$

This leaves us with two possibilities to continue. Either we can keep working from the bottom up and look for another application of the →Intro rule, or we can work from the top down and apply a ∨Elim rule. Let us apply the latter. We will use the assumption $\neg\varphi \vee \psi$ as the leftmost premise of ∨Elim. For a valid application of ∨Elim, the other two premises must be identical to the conclusion $\varphi \to \psi$, but each may be derived in turn from another assumption, namely one of the two disjuncts of $\neg\varphi \vee \psi$. So our derivation will look like this:

$$\dfrac{\dfrac{\begin{array}{ccc}[\neg\varphi \vee \psi]^1 & \begin{array}{c}[\neg\varphi]^2 \\ \vdots \\ \varphi \to \psi\end{array} & \begin{array}{c}[\psi]^2 \\ \vdots \\ \varphi \to \psi\end{array}\end{array}}{\varphi \to \psi}\ {\scriptstyle 2}\ \vee\text{Elim}}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}\ {\scriptstyle 1}\ \to\text{Intro}$$

In each of the two branches on the right, we want to derive $\varphi \to \psi$, which is best done using →Intro.

$$\dfrac{\dfrac{\begin{array}{ccc}[\neg\varphi \vee \psi]^1 & \dfrac{\begin{array}{c}[\neg\varphi]^2, [\varphi]^3 \\ \vdots \\ \psi\end{array}}{\varphi \to \psi}\ {\scriptstyle 3}\ \to\text{Intro} & \dfrac{\begin{array}{c}[\psi]^2, [\varphi]^4 \\ \vdots \\ \psi\end{array}}{\varphi \to \psi}\ {\scriptstyle 4}\ \to\text{Intro}\end{array}}{\varphi \to \psi}\ {\scriptstyle 2}\ \vee\text{Elim}}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}\ {\scriptstyle 1}\ \to\text{Intro}$$

For the two missing parts of the derivation, we need derivations of ψ from $\neg\varphi$ and φ in the middle, and from φ and ψ on the left. Let's take the former first. $\neg\varphi$ and φ are the two premises of ¬Elim:

$$\dfrac{\begin{array}{cc}[\neg\varphi]^2 & [\varphi]^3\end{array}}{\begin{array}{c}\bot \\ \vdots \\ \psi\end{array}}\ \neg\text{Elim}$$

By using $\bot_I$, we can obtain ψ as a conclusion and complete the branch.

$$\dfrac{\dfrac{\dfrac{[\neg\varphi]^2 \qquad [\varphi]^3}{\dfrac{\bot}{\psi}\,\bot_I}\,\bot\text{Intro}}{\varphi \to \psi}\,{}^{3}\to\text{Intro} \qquad \dfrac{[\psi]^2,[\varphi]^4}{\varphi \to \psi}\,{}^{4}\to\text{Intro}}{\varphi \to \psi}\,{}^{2}\vee\text{Elim}}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}\,{}^{1}\to\text{Intro}$$

Let's now look at the rightmost branch. Here it's important to realize that the definition of derivation *allows assumptions to be discharged* but *does not require* them to be. In other words, if we can derive ψ from one of the assumptions φ and ψ without using the other, that's ok. And to derive ψ from ψ is trivial: ψ by itself is such a derivation, and no inferences are needed. So we can simply delete the assumption φ.

$$\dfrac{\dfrac{\dfrac{[\neg\varphi]^2 \qquad [\varphi]^3}{\dfrac{\bot}{\psi}\,\bot_I}\,\neg\text{Elim}}{\varphi \to \psi}\,{}^{3}\to\text{Intro} \qquad \dfrac{[\psi]^2}{\varphi \to \psi}\,\to\text{Intro}}{\varphi \to \psi}\,{}^{2}\vee\text{Elim}}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}\,{}^{1}\to\text{Intro}$$

Note that in the finished derivation, the rightmost →Intro inference does not actually discharge any assumptions.

Example 10.6. So far we have not needed the $\bot_C$ rule. It is special in that it allows us to discharge an assumption that isn't a sub-formula of the conclusion of the rule. It is closely related to the $\bot_I$ rule. In fact, the $\bot_I$ rule is a special case of the $\bot_C$ rule—there is a logic called "intuitionistic logic" in which only $\bot_I$ is allowed. The $\bot_C$ rule is a last resort when nothing else works. For instance, suppose we want to derive $\varphi \vee \neg\varphi$. Our usual strategy would be to attempt to derive $\varphi \vee \neg\varphi$ using ∨Intro. But this would require us to derive either φ or $\neg\varphi$ from no assumptions, and this can't be done. $\bot_C$ to the rescue!

$$\dfrac{\dfrac{[\neg(\varphi \vee \neg\varphi)]^1}{\bot}}{\varphi \vee \neg\varphi}\,{}^{1}\bot_C$$

Now we're looking for a derivation of $\bot$ from $\neg(\varphi \vee \neg\varphi)$. Since $\bot$ is the conclusion of ¬Elim we might try that:

$$\dfrac{\dfrac{\dfrac{}{\neg(\varphi \vee \neg\varphi)}^{1} \qquad \dfrac{}{\neg(\varphi \vee \neg\varphi)}^{1}}{\dfrac{\neg\varphi \qquad \varphi}{\bot}\,\neg\text{Elim}}}{\dfrac{\bot}{\varphi \vee \neg\varphi}\,\bot_C}\,1$$

Our strategy for finding a derivation of $\neg\varphi$ calls for an application of $\neg$Intro:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg(\varphi \vee \neg\varphi)}^{1}, \dfrac{}{[\varphi]}^{2}}{\dfrac{\bot}{\neg\varphi}\,\neg\text{Intro}} \qquad \dfrac{}{\neg(\varphi \vee \neg\varphi)}^{1}}{\dfrac{\neg\varphi \qquad \varphi}{\bot}\,\neg\text{Elim}}}{\dfrac{\bot}{\varphi \vee \neg\varphi}\,\bot_C}\,1$$

Here, we can get $\bot$ easily by applying $\neg$Elim to the assumption $\neg(\varphi \vee \neg\varphi)$ and $\varphi \vee \neg\varphi$ which follows from our new assumption φ by $\vee$Intro:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg(\varphi \vee \neg\varphi)}^{1} \qquad \dfrac{\dfrac{}{[\varphi]}^{2}}{\varphi \vee \neg\varphi}\,\vee\text{Intro}}{\bot}\,\neg\text{Elim}}{\dfrac{\bot}{\neg\varphi}\,\neg\text{Intro}} \qquad \dfrac{}{\neg(\varphi \vee \neg\varphi)}^{1}}{\dfrac{\neg\varphi \qquad \varphi}{\bot}\,\neg\text{Elim}}}{\dfrac{\bot}{\varphi \vee \neg\varphi}\,\bot_C}\,1$$

On the right side we use the same strategy, except we get φ by $\bot_C$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg(\varphi \vee \neg\varphi)}^{1} \qquad \dfrac{\dfrac{}{[\varphi]}^{2}}{\varphi \vee \neg\varphi}\,\vee\text{Intro}}{\bot}\,\neg\text{Elim}}{\dfrac{\bot}{\neg\varphi}\,\neg\text{Intro}} \qquad \dfrac{\dfrac{\dfrac{}{\neg(\varphi \vee \neg\varphi)}^{1} \qquad \dfrac{\dfrac{}{[\neg\varphi]}^{3}}{\varphi \vee \neg\varphi}\,\vee\text{Intro}}{\bot}\,\neg\text{Elim}}{\dfrac{\bot}{\varphi}\,\bot_C}\,3}{\dfrac{\bot}{\varphi \vee \neg\varphi}\,\bot_C}\,1$$

Problem 10.1. Give derivations that show the following:

1. $\varphi \wedge (\psi \wedge \chi) \vdash (\varphi \wedge \psi) \wedge \chi$.
2. $\varphi \vee (\psi \vee \chi) \vdash (\varphi \vee \psi) \vee \chi$.
3. $\varphi \to (\psi \to \chi) \vdash \psi \to (\varphi \to \chi)$.
4. $\varphi \vdash \neg\neg\varphi$.

Problem 10.2. Give derivations that show the following:

1. $(\varphi \vee \psi) \to \chi \vdash \varphi \to \chi$.

2. $(\varphi \to \chi) \land (\psi \to \chi) \vdash (\varphi \lor \psi) \to \chi$.
3. $\vdash \neg(\varphi \land \neg\varphi)$.
4. $\psi \to \varphi \vdash \neg\varphi \to \neg\psi$.
5. $\vdash (\varphi \to \neg\varphi) \to \neg\varphi$.
6. $\vdash \neg(\varphi \to \psi) \to \neg\psi$.
7. $\varphi \to \chi \vdash \neg(\varphi \land \neg\chi)$.
8. $\varphi \land \neg\chi \vdash \neg(\varphi \to \chi)$.
9. $\varphi \lor \psi, \neg\psi \vdash \varphi$.
10. $\neg\varphi \lor \neg\psi \vdash \neg(\varphi \land \psi)$.
11. $\vdash (\neg\varphi \land \neg\psi) \to \neg(\varphi \lor \psi)$.
12. $\vdash \neg(\varphi \lor \psi) \to (\neg\varphi \land \neg\psi)$.

Problem 10.3. Give derivations that show the following:

1. $\neg(\varphi \to \psi) \vdash \varphi$.
2. $\neg(\varphi \land \psi) \vdash \neg\varphi \lor \neg\psi$.
3. $\varphi \to \psi \vdash \neg\varphi \lor \psi$.
4. $\vdash \neg\neg\varphi \to \varphi$.
5. $\varphi \to \psi, \neg\varphi \to \psi \vdash \psi$.
6. $(\varphi \land \psi) \to \chi \vdash (\varphi \to \chi) \lor (\psi \to \chi)$.
7. $(\varphi \to \psi) \to \varphi \vdash \varphi$.
8. $\vdash (\varphi \to \psi) \lor (\psi \to \chi)$.

(These all require the $\bot_C$ rule.)

---

content/propositional-logic/../first-order-logic/natural-deduction/proof-theoretic-

10.5 Proof-Theoretic Notions

pl:ntd:ptn:sec

This section collects the definitions the provability relation and consistency for natural deduction.

explanation

Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain sentences from others. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

Definition 10.7 (Theorems). A sentence φ is a *theorem* if there is a derivation of φ in natural deduction in which all assumptions are discharged. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 10.8 (Derivability). A sentence φ is *derivable from* a set of sentences Γ, $\Gamma \vdash \varphi$, if there is a derivation with conclusion φ and in which every assumption is either discharged or is in Γ. If φ is not derivable from Γ we write $\Gamma \nvdash \varphi$.

Definition 10.9 (Consistency). A set of sentences Γ is *inconsistent* iff $\Gamma \vdash \bot$. If Γ is not inconsistent, i.e., if $\Gamma \nvdash \bot$, we say it is *consistent*.

pl:ntd:ptn:prop:reflexivity

Proposition 10.10 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.*

Proof. The assumption φ by itself is a derivation of φ where every undischarged assumption (i.e., φ) is in Γ. □

pl:ntd:ptn:prop:monotonicity

Proposition 10.11 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.*

Proof. Any derivation of φ from Γ is also a derivation of φ from Δ. □

pl:ntd:ptn:prop:transitivity

Proposition 10.12 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.*

Proof. If $\Gamma \vdash \varphi$, there is a derivation δ_0 of φ with all undischarged assumptions in Γ. If $\{\varphi\} \cup \Delta \vdash \psi$, then there is a derivation δ_1 of ψ with all undischarged assumptions in $\{\varphi\} \cup \Delta$. Now consider:

$$\dfrac{\dfrac{\begin{array}{c}\Delta, [\varphi]^1 \\ \vdots\,\delta_1 \\ \psi\end{array}}{\varphi \to \psi}\,1 \to\text{Intro} \qquad \begin{array}{c}\Gamma \\ \vdots\,\delta_0 \\ \varphi\end{array}}{\psi} \to\text{Elim}$$

The undischarged assumptions are now all among $\Gamma \cup \Delta$, so this shows $\Gamma \cup \Delta \vdash \psi$. □

When $\Gamma = \{\varphi_1, \varphi_2, \ldots, \varphi_k\}$ is a finite set we may use the simplified notation $\varphi_1, \varphi_2, \ldots, \varphi_k \vdash \psi$ for $\Gamma \vdash \psi$, in particular $\varphi \vdash \psi$ means that $\{\varphi\} \vdash \psi$.

Note that if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \ldots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

Proposition 10.13. *The following are equivalent.* pl:ntd:ptn: prop:incons

1. *Γ is inconsistent.*
2. *$\Gamma \vdash \varphi$ for every sentence φ.*
3. *$\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\varphi$ for some sentence φ.*

Proof. Exercise. □

Problem 10.4. Prove Proposition 10.13

Proposition 10.14 (Compactness). pl:ntd:ptn: prop:proves-compact

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof.

1. If $\Gamma \vdash \varphi$, then there is a derivation δ of φ from Γ. Let Γ_0 be the set of undischarged assumptions of δ. Since any derivation is finite, Γ_0 can only contain finitely many sentences. So, δ is a derivation of φ from a finite $\Gamma_0 \subseteq \Gamma$.
2. This is the contrapositive of (1) for the special case $\varphi \equiv \bot$. □

---

content/propositional-logic/../first-order-logic/natural-deduction/provability-cons

10.6 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem. pl:ntd:prv: sec

Proposition 10.15. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.* pl:ntd:prv: prop:provability-contr

Proof. Let the derivation of φ from Γ be δ_1 and the derivation of $\bot$ from $\Gamma \cup \{\varphi\}$ be δ_2. We can then derive:

$$\dfrac{\dfrac{\dfrac{\Gamma, [\varphi]^1}{\vdots\,\delta_2}}{\dfrac{\bot}{\neg\varphi}\,1\,\neg\text{Intro}} \qquad \dfrac{\Gamma}{\vdots\,\delta_1}}{\bot}\,\neg\text{Elim}$$

In the new derivation, the assumption φ is discharged, so it is a derivation from Γ. □

pl:ntd:prv: prop:prov-incons

Proposition 10.16. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.*

Proof. First suppose $\Gamma \vdash \varphi$, i.e., there is a derivation δ_0 of φ from undischarged assumptions Γ. We obtain a derivation of $\bot$ from $\Gamma \cup \{\neg\varphi\}$ as follows:

$$\dfrac{\dfrac{\neg\varphi \qquad \dfrac{\Gamma}{\vdots\,\delta_0}}{\varphi}}{\bot}\,\neg\text{Elim}$$

Now assume $\Gamma \cup \{\neg\varphi\}$ is inconsistent, and let δ_1 be the corresponding derivation of $\bot$ from undischarged assumptions in $\Gamma \cup \{\neg\varphi\}$. We obtain a derivation of φ from Γ alone by using $\bot_C$:

$$\dfrac{\dfrac{\Gamma, [\neg\varphi]^1}{\vdots\,\delta_1}}{\dfrac{\bot}{\varphi}\,1\,\bot_C}$$

□

Problem 10.5. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

pl:ntd:prv: prop:explicit-inc

Proposition 10.17. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

Proof. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there is a derivation δ of φ from Γ. Consider this simple application of the ¬Elim rule:

$$\dfrac{\dfrac{\neg\varphi \qquad \dfrac{\Gamma}{\vdots\,\delta}}{\varphi}}{\bot}\,\neg\text{Elim}$$

Since $\neg\varphi \in \Gamma$, all undischarged assumptions are in Γ, this shows that $\Gamma \vdash \bot$.□

pl:ntd:prv: prop:provability-exhaustive

Proposition 10.18. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

Proof. There are derivations δ_1 and δ_2 of $\bot$ from $\Gamma \cup \{\varphi\}$ and $\bot$ from $\Gamma \cup \{\neg\varphi\}$, respectively. We can then derive

$$\dfrac{\dfrac{\Gamma, [\neg\varphi]^2 \quad \vdots\ \delta_2 \quad \bot}{\neg\neg\varphi}\ 2\ \neg\text{Intro} \qquad \dfrac{\Gamma, [\varphi]^1 \quad \vdots\ \delta_1 \quad \bot}{\neg\varphi}\ 1\ \neg\text{Intro}}{\bot}\ \neg\text{Elim}$$

Since the assumptions φ and $\neg\varphi$ are discharged, this is a derivation of $\bot$ from Γ alone. Hence Γ is inconsistent. □

---

content/propositional-logic/../first-order-logic/natural-deduction/provability-prop

10.7 Derivability and the Propositional Connectives

explanation

We establish that the derivability relation $\vdash$ of natural deduction is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem.

pl:ntd:ppr:sec

Proposition 10.19.

pl:ntd:ppr:prop:provability-land

1. *Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$*

pl:ntd:ppr:prop:provability-land-left

2. *$\varphi, \psi \vdash \varphi \wedge \psi$.*

pl:ntd:ppr:prop:provability-land-right

Proof. 1. We can derive both

$$\dfrac{\varphi \wedge \psi}{\varphi}\ \wedge\text{Elim} \qquad \dfrac{\varphi \wedge \psi}{\psi}\ \wedge\text{Elim}$$

2. We can derive:

$$\dfrac{\varphi \qquad \psi}{\varphi \wedge \psi}\ \wedge\text{Intro}$$

□

Proposition 10.20.

pl:ntd:ppr:prop:provability-lor

1. *$\varphi \vee \psi, \neg\varphi, \neg\psi$ is inconsistent.*

2. *Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.*

Proof. 1. Consider the following derivation:

$$\dfrac{\varphi \vee \psi \qquad \dfrac{\neg\varphi \qquad [\varphi]^1}{\bot}\neg\text{Elim} \qquad \dfrac{\neg\psi \qquad [\psi]^1}{\bot}\neg\text{Elim}}{\bot}\vee\text{Elim}\ {}^1$$

This is a derivation of $\bot$ from undischarged assumptions $\varphi \vee \psi$, $\neg\varphi$, and $\neg\psi$.

2. We can derive both

$$\dfrac{\varphi}{\varphi \vee \psi}\vee\text{Intro} \qquad \dfrac{\psi}{\varphi \vee \psi}\vee\text{Intro}$$

□

pl:ntd:ppr: prop:provability-lif

Proposition 10.21.

pl:ntd:ppr: prop:provability-lif-left

1. *$\varphi, \varphi \to \psi \vdash \psi$.*

pl:ntd:ppr: prop:provability-lif-right

2. *Both $\neg\varphi \vdash \varphi \to \psi$ and $\psi \vdash \varphi \to \psi$.*

Proof. 1. We can derive:

$$\dfrac{\varphi \to \psi \qquad \varphi}{\psi}\to\text{Elim}$$

2. This is shown by the following two derivations:

$$\dfrac{\dfrac{\neg\varphi \qquad [\varphi]^1}{\dfrac{\bot}{\psi}\bot_I}\neg\text{Elim}}{\varphi \to \psi}\to\text{Intro}\ {}^1 \qquad \dfrac{\psi}{\varphi \to \psi}\to\text{Intro}$$

Note that →Intro may, but does not have to, discharge the assumption φ. □

---

content/propositional-logic/../first-order-logic/natural-deduction/soundness.tex

10.8 Soundness

pl:ntd:sou: sec

A derivation system, such as natural deduction, is *sound* if it cannot derive things that do not actually follow. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that explanation

1. every derivable sentence is a tautology;
2. if a sentence is derivable from some others, it is also a consequence of them;

3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Theorem 10.22 (Soundness). *If φ is derivable from the undischarged assumptions Γ, then $\Gamma \vDash \varphi$.* pl:ntd:sou:thm:soundness

Proof. Let δ be a derivation of φ. We proceed by induction on the number of inferences in δ.

For the induction basis we show the claim if the number of inferences is 0. In this case, δ consists only of a single sentence φ, i.e., an assumption. That assumption is undischarged, since assumptions can only be discharged by inferences, and there are no inferences. So, any valuation $\mathfrak{v}$ that satisfies all of the undischarged assumptions of the proof also satisfies φ.

Now for the inductive step. Suppose that δ contains n inferences. The premise(s) of the lowermost inference are derived using sub-derivations, each of which contains fewer than n inferences. We assume the induction hypothesis: The premises of the lowermost inference follow from the undischarged assumptions of the sub-derivations ending in those premises. We have to show that the conclusion φ follows from the undischarged assumptions of the entire proof.

We distinguish cases according to the type of the lowermost inference. First, we consider the possible inferences with only one premise.

1. Suppose that the last inference is ¬Intro: The derivation has the form

$$\dfrac{\begin{array}{c}\Gamma, [\varphi]^n \\ \vdots \\ \delta_1 \\ \vdots \\ \bot\end{array}}{\neg\varphi}\, {}^{n}\neg\text{Intro}$$

   By inductive hypothesis, $\bot$ follows from the undischarged assumptions $\Gamma \cup \{\varphi\}$ of δ_1. Consider a valuation $\mathfrak{v}$. We need to show that, if $\mathfrak{v} \vDash \Gamma$, then $\mathfrak{v} \vDash \neg\varphi$. Suppose for reductio that $\mathfrak{v} \vDash \Gamma$, but $\mathfrak{v} \nvDash \neg\varphi$, i.e., $\mathfrak{v} \vDash \varphi$. This would mean that $\mathfrak{v} \vDash \Gamma \cup \{\varphi\}$. This is contrary to our inductive hypothesis. So, $\mathfrak{v} \vDash \neg\varphi$.

2. The last inference is ∧Elim: There are two variants: φ or ψ may be inferred from the premise $\varphi \wedge \psi$. Consider the first case. The derivation δ looks like this:

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ \delta_1\\ \vdots\\ \varphi \wedge \psi\end{array}}{\varphi}\ \wedge\text{Elim}$$

By inductive hypothesis, $\varphi \wedge \psi$ follows from the undischarged assumptions Γ of δ_1. Consider a structure $\mathfrak{v}$. We need to show that, if $\mathfrak{v} \vDash \Gamma$, then $\mathfrak{v} \vDash \varphi$. Suppose $\mathfrak{v} \vDash \Gamma$. By our inductive hypothesis ($\Gamma \vDash \varphi \wedge \psi$), we know that $\mathfrak{v} \vDash \varphi \wedge \psi$. By definition, $\mathfrak{v} \vDash \varphi \wedge \psi$ iff $\mathfrak{v} \vDash \varphi$ and $\mathfrak{v} \vDash \psi$. (The case where ψ is inferred from $\varphi \wedge \psi$ is handled similarly.)

3. The last inference is $\vee$Intro: There are two variants: $\varphi \vee \psi$ may be inferred from the premise φ or the premise ψ. Consider the first case. The derivation has the form

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ \delta_1\\ \vdots\\ \varphi\end{array}}{\varphi \vee \psi}\ \vee\text{Intro}$$

By inductive hypothesis, φ follows from the undischarged assumptions Γ of δ_1. Consider a valuation $\mathfrak{v}$. We need to show that, if $\mathfrak{v} \vDash \Gamma$, then $\mathfrak{v} \vDash \varphi \vee \psi$. Suppose $\mathfrak{v} \vDash \Gamma$; then $\mathfrak{v} \vDash \varphi$ since $\Gamma \vDash \varphi$ (the inductive hypothesis). So it must also be the case that $\mathfrak{v} \vDash \varphi \vee \psi$. (The case where $\varphi \vee \psi$ is inferred from ψ is handled similarly.)

4. The last inference is $\rightarrow$Intro: $\varphi \rightarrow \psi$ is inferred from a subproof with assumption φ and conclusion ψ, i.e.,

$$\dfrac{\begin{array}{c}\Gamma, [\varphi]^n\\ \vdots\\ \delta_1\\ \vdots\\ \psi\end{array}}{\varphi \rightarrow \psi}\ {}^{n}\ \rightarrow\text{Intro}$$

By inductive hypothesis, ψ follows from the undischarged assumptions of δ_1, i.e., $\Gamma \cup \{\varphi\} \vDash \psi$. Consider a valuation $\mathfrak{v}$. The undischarged assumptions of δ are just Γ, since φ is discharged at the last inference. So we need to show that $\Gamma \vDash \varphi \rightarrow \psi$. For reductio, suppose that for some valuation $\mathfrak{v}$, $\mathfrak{v} \vDash \Gamma$ but $\mathfrak{v} \nvDash \varphi \rightarrow \psi$. So, $\mathfrak{v} \vDash \varphi$ and $\mathfrak{v} \nvDash \psi$. But by hypothesis, ψ is a consequence of $\Gamma \cup \{\varphi\}$, i.e., $\mathfrak{v} \vDash \psi$, which is a contradiction. So, $\Gamma \vDash \varphi \rightarrow \psi$.

5. The last inference is $\bot_I$: Here, δ ends in

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\,\delta_1\\ \vdots\\ \bot\end{array}}{\varphi}\,\bot_I$$

By induction hypothesis, $\Gamma \vDash \bot$. We have to show that $\Gamma \vDash \varphi$. Suppose not; then for some $\mathfrak{v}$ we have $\mathfrak{v} \vDash \Gamma$ and $\mathfrak{v} \nvDash \varphi$. But we always have $\mathfrak{v} \nvDash \bot$, so this would mean that $\Gamma \nvDash \bot$, contrary to the induction hypothesis.

6. The last inference is $\bot_C$: Exercise.

Now let's consider the possible inferences with several premises: $\vee$Elim, $\wedge$Intro, and $\to$Elim.

1. The last inference is $\wedge$Intro. $\varphi \wedge \psi$ is inferred from the premises φ and ψ and δ has the form

$$\dfrac{\begin{array}{cc}\Gamma_1 & \Gamma_2\\ \vdots\,\delta_1 & \vdots\,\delta_2\\ \varphi & \psi\end{array}}{\varphi \wedge \psi}\,\wedge\text{Intro}$$

By induction hypothesis, φ follows from the undischarged assumptions Γ_1 of δ_1 and ψ follows from the undischarged assumptions Γ_2 of δ_2. The undischarged assumptions of δ are $\Gamma_1 \cup \Gamma_2$, so we have to show that $\Gamma_1 \cup \Gamma_2 \vDash \varphi \wedge \psi$. Consider a valuation $\mathfrak{v}$ with $\mathfrak{v} \vDash \Gamma_1 \cup \Gamma_2$. Since $\mathfrak{v} \vDash \Gamma_1$, it must be the case that $\mathfrak{v} \vDash \varphi$ as $\Gamma_1 \vDash \varphi$, and since $\mathfrak{v} \vDash \Gamma_2$, $\mathfrak{v} \vDash \psi$ since $\Gamma_2 \vDash \psi$. Together, $\mathfrak{v} \vDash \varphi \wedge \psi$.

2. The last inference is $\vee$Elim: Exercise.

3. The last inference is $\to$Elim. ψ is inferred from the premises $\varphi \to \psi$ and φ. The derivation δ looks like this:

$$\dfrac{\begin{array}{cc}\Gamma_1 & \Gamma_2\\ \vdots\,\delta_1 & \vdots\,\delta_2\\ \varphi \to \psi & \varphi\end{array}}{\psi}\,\to\text{Elim}$$

□

By induction hypothesis, $\varphi \to \psi$ follows from the undischarged assumptions Γ_1 of δ_1 and φ follows from the undischarged assumptions Γ_2 of δ_2. Consider a valuation $\mathfrak{v}$. We need to show that, if $\mathfrak{v} \vDash \Gamma_1 \cup \Gamma_2$, then $\mathfrak{v} \vDash \psi$. Suppose $\mathfrak{v} \vDash \Gamma_1 \cup \Gamma_2$. Since $\Gamma_1 \vDash \varphi \to \psi$, $\mathfrak{v} \vDash \varphi \to \psi$. Since $\Gamma_2 \vDash \varphi$, we have $\mathfrak{v} \vDash \varphi$. This means that $\mathfrak{v} \vDash \psi$ (For if $\mathfrak{v} \nvDash \psi$, since $\mathfrak{v} \vDash \varphi$, we'd have $\mathfrak{v} \nvDash \varphi \to \psi$, contradicting $\mathfrak{v} \vDash \varphi \to \psi$).

4. The last inference is ¬Elim: Exercise.

Problem 10.6. Complete the proof of Theorem 10.22.

pl:ntd:sou: cor:weak-soundness

Corollary 10.23. *If $\vdash \varphi$, then φ is a tautology.*

pl:ntd:sou: cor:consistency-soundness

Corollary 10.24. *If Γ is satisfiable, then it is consistent.*

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then $\Gamma \vdash \bot$, i.e., there is a derivation of $\bot$ from undischarged assumptions in Γ. By Theorem 10.22, any valuation $\mathfrak{v}$ that satisfies Γ must satisfy $\bot$. Since $\mathfrak{v} \not\models \bot$ for every valuation $\mathfrak{v}$, no $\mathfrak{v}$ can satisfy Γ, i.e., Γ is not satisfiable. □

Chapter 11

Tableaux

This chapter presents a signed analytic tableaux system.
To include or exclude material relevant to natural deduction as a proof system, use the "prfTab" tag.

---

content/propositional-logic/../first-order-logic/tableaux/rules-and-proofs.tex

11.1 Rules and Tableaux

pl:tab:rul: sec

A tableau is a systematic survey of the possible ways a sentence can be true or false in a structure. The building blocks of a tableau are signed formulas: sentences plus a truth value "sign," either $\mathbb{T}$ or $\mathbb{F}$. These signed formulas are arranged in a (downward growing) tree.

Definition 11.1. A *signed formula* is a pair consisting of a truth value and a sentence, i.e., either:

$$\mathbb{T}\varphi \text{ or } \mathbb{F}\varphi.$$

Intuitively, we might read $\mathbb{T}\,\varphi$ as "φ might be true" and $\mathbb{F}\,\varphi$ as "φ might be false" (in some structure).

Each signed formula in the tree is either an *assumption* (which are listed at the very top of the tree), or it is obtained from a signed formula above it by one of a number of rules of inference. There are two rules for each possible main operator of the preceding formula, one for the case where the sign is $\mathbb{T}$, and one for the case where the sign is $\mathbb{F}$. Some rules allow the tree to branch, and some only add signed formulas to the branch. A rule may be (and often must be) applied not to the immediately preceding signed formula, but to any signed formula in the branch from the root to the place the rule is applied.

A branch is *closed* when it contains both $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$. A closed tableau is one where every branch is closed. Under the intuitive interpretation, any branch describes a joint possibility, but $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$ are not jointly possible. In other words, if a branch is closed, the possibility it describes has been ruled out. In particular, that means that a closed tableau rules out all possibilities of simultaneously making every assumption of the form $\mathbb{T}\,\varphi$ true and every assumption of the form $\mathbb{F}\,\varphi$ false.

A closed tableau *for* φ is a closed tableau with root $\mathbb{F}\,\varphi$. If such a closed tableau exists, all possibilities for φ being false have been ruled out; i.e., φ must be true in every structure.

---

content/propositional-logic/../first-order-logic/tableaux/propositional-rules.tex

11.2 Propositional Rules

Rules for $\neg$

pl:tab:prl:sec

$$\frac{\mathbb{T}\,\neg\varphi}{\mathbb{F}\,\varphi}\,\neg\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\neg\varphi}{\mathbb{T}\,\varphi}\,\neg\mathbb{F}$$

Rules for $\wedge$

$$\frac{\mathbb{T}\,\varphi \wedge \psi}{\begin{array}{c}\mathbb{T}\,\varphi\\ \mathbb{T}\,\psi\end{array}}\,\wedge\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\varphi \wedge \psi}{\mathbb{F}\,\varphi \quad\big|\quad \mathbb{F}\,\psi}\,\wedge\mathbb{F}$$

Rules for $\vee$

$$\frac{\mathbb{T}\,\varphi \vee \psi}{\mathbb{T}\,\varphi \quad | \quad \mathbb{T}\,\psi}\,\vee\mathbb{T} \qquad \frac{\mathbb{F}\,\varphi \vee \psi}{\begin{array}{c}\mathbb{F}\,\varphi\\ \mathbb{F}\,\psi\end{array}}\,\vee\mathbb{F}$$

Rules for $\rightarrow$

$$\frac{\mathbb{T}\,\varphi \rightarrow \psi}{\mathbb{F}\,\varphi \quad | \quad \mathbb{T}\,\psi}\,\rightarrow\mathbb{T} \qquad \frac{\mathbb{F}\,\varphi \rightarrow \psi}{\begin{array}{c}\mathbb{T}\,\varphi\\ \mathbb{F}\,\psi\end{array}}\,\rightarrow\mathbb{F}$$

The Cut Rule

$$\frac{}{\mathbb{T}\,\varphi \quad | \quad \mathbb{F}\,\varphi}\,\text{Cut}$$

The Cut rule is not applied "to" a previous signed formula; rather, it allows every branch in a tableau to be split in two, one branch containing $\mathbb{T}\,\varphi$, the other $\mathbb{F}\,\varphi$. It is not necessary—any set of signed formulas with a closed tableau has one not using Cut—but it allows us to combine tableaux in a convenient way.

---

content/propositional-logic/../first-order-logic/tableaux/derivations.tex

11.3 Tableaux

pl:tab:der:sec

We've said what an assumption is, and we've given the rules of inference. Tableaux are inductively generated from these: each tableau either is a single branch consisting of one or more assumptions, or it results from a tableau by applying one of the rules of inference on a branch.

explanation

Definition 11.2 (Tableau). A tableau for assumptions $S_1\varphi_1, \ldots, S_n\varphi_n$ (where each S_i is either $\mathbb{T}$ or $\mathbb{F}$) is a finite tree of signed formulas satisfying the following conditions:

1. The n topmost signed formulas of the tree are $S_i\varphi_i$, one below the other.
2. Every signed formula in the tree that is not one of the assumptions results from a correct application of an inference rule to a signed formula in the branch above it.

A branch of a tableau is *closed* iff it contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, and *open* otherwise. A tableau in which every branch is closed is a *closed tableau* (for its set of assumptions). If a tableau is not closed, i.e., if it contains at least one open branch, it is *open*.

Example 11.3. Every set of assumptions on its own is a tableau, but it will generally not be closed. (Obviously, it is closed only if the assumptions already contain a pair of signed formulas $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$.)

From a tableau (open or closed) we can obtain a new, larger one by applying one of the rules of inference to a signed formula φ in it. The rule will append one or more signed formulas to the end of any branch containing the occurrence of φ to which we apply the rule.

For instance, consider the assumption $\mathbb{T}\varphi \wedge \neg\varphi$. Here is the (open) tableau consisting of just that assumption:

1.	$\mathbb{T}\varphi \wedge \neg\varphi$	Assumption

We obtain a new tableau from it by applying the $\wedge\mathbb{T}$ rule to the assumption. That rule allows us to add two new lines to the tableau, $\mathbb{T}\varphi$ and $\mathbb{T}\neg\varphi$:

1.	$\mathbb{T}\varphi \wedge \neg\varphi$	Assumption
2.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}\,1$
3.	$\mathbb{T}\neg\varphi$	$\wedge\mathbb{T}\,1$

When we write down tableaux, we record the rules we've applied on the right (e.g., $\wedge\mathbb{T}1$ means that the signed formula on that line is the result of applying the $\wedge\mathbb{T}$ rule to the signed formula on line 1). This new tableau now contains additional signed formulas, but to only one ($\mathbb{T}\neg\varphi$) can we apply a rule (in this case, the $\neg\mathbb{T}$ rule). This results in the closed tableau

1.	$\mathbb{T}\varphi \wedge \neg\varphi$	Assumption
2.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}\,1$
3.	$\mathbb{T}\neg\varphi$	$\wedge\mathbb{T}\,1$
4.	$\mathbb{F}\varphi$	$\neg\mathbb{T}\,3$
	$\otimes$	

---

content/propositional-logic/../first-order-logic/tableaux/proving-things.tex

11.4 Examples of Tableaux

pl:tab:pro:sec

Example 11.4. Let's find a closed tableau for the sentence $(\varphi \wedge \psi) \to \varphi$.

We begin by writing the corresponding assumption at the top of the tableau.

1.	$\mathbb{F}(\varphi \wedge \psi) \to \varphi$	Assumption

There is only one assumption, so only one signed formula to which we can apply a rule. (For every signed formula, there is always at most one rule that can be applied: it's the rule for the corresponding sign and main operator of the sentence.) In this case, this means, we must apply $\to\mathbb{F}$.

1.	$\mathbb{F}\,(\varphi \wedge \psi) \to \varphi$ ✓	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\varphi$	$\to\mathbb{F}\,1$

To keep track of which signed formulas we have applied their corresponding rules to, we write a checkmark next to the sentence. However, *only* write a checkmark if the rule has been applied to all open branches. Once a signed formula has had the corresponding rule applied in every open branch, we will not have to return to it and apply the rule again. In this case, there is only one branch, so the rule only has to be applied once. (Note that checkmarks are only a convenience for constructing tableaux and are not officially part of the syntax of tableaux.)

There is one new signed formula to which we can apply a rule: the $\mathbb{T}\,\varphi \wedge \psi$ on line 2. Applying the $\wedge\mathbb{T}$ rule results in:

1.	$\mathbb{F}\,(\varphi \wedge \psi) \to \varphi$ ✓	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\varphi$	$\wedge\mathbb{T}\,2$
5.	$\mathbb{T}\,\psi$	$\wedge\mathbb{T}\,2$
	$\otimes$	

Since the branch now contains both $\mathbb{T}\,\varphi$ (on line 4) and $\mathbb{F}\,\varphi$ (on line 3), the branch is closed. Since it is the only branch, the tableau is closed. We have found a closed tableau for $(\varphi \wedge \psi) \to \varphi$.

Example 11.5. Now let's find a closed tableau for $(\neg\varphi \vee \psi) \to (\varphi \to \psi)$.

We begin with the corresponding assumption:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$	Assumption

The one signed formula in this tableau has main operator $\to$ and sign $\mathbb{F}$, so we apply the $\to\mathbb{F}$ rule to it to obtain:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$ ✓	Assumption
2.	$\mathbb{T}\,\neg\varphi \vee \psi$	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,(\varphi \to \psi)$	$\to\mathbb{F}\,1$

We now have a choice as to whether to apply $\vee\mathbb{T}$ to line 2 or $\to\mathbb{F}$ to line 3. It actually doesn't matter which order we pick, as long as each signed formula has its corresponding rule applied in every branch. So let's pick the first one. The $\vee\mathbb{T}$ rule allows the tableau to branch, and the two conclusions of the rule

will be the new signed formulas added to the two new branches. This results in:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$ ✓	Assumption
2.	$\mathbb{T}\,\neg\varphi \vee \psi$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,(\varphi \to \psi)$	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\neg\varphi$ | $\mathbb{T}\,\psi$	$\vee\mathbb{T}\,2$

We have not applied the $\to\mathbb{F}$ rule to line 3 yet: let's do that now. To save time, we apply it to both branches. Recall that we write a checkmark next to a signed formula only if we have applied the corresponding rule in every open branch. So it's a good idea to apply a rule at the end of every branch that contains the signed formula the rule applies to. That way we won't have to return to that signed formula lower down in the various branches.

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$ ✓	Assumption
2.	$\mathbb{T}\,\neg\varphi \vee \psi$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,(\varphi \to \psi)$ ✓	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\neg\varphi$ | $\mathbb{T}\,\psi$	$\vee\mathbb{T}\,2$
5.	$\mathbb{T}\,\varphi$ | $\mathbb{T}\,\varphi$	$\to\mathbb{F}\,3$
6.	$\mathbb{F}\,\psi$ | $\mathbb{F}\,\psi$	$\to\mathbb{F}\,3$
	| $\otimes$	

The right branch is now closed. On the left branch, we can still apply the $\neg\mathbb{T}$ rule to line 4. This results in $\mathbb{F}\,\varphi$ and closes the left branch:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$ ✓	Assumption
2.	$\mathbb{T}\,\neg\varphi \vee \psi$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,(\varphi \to \psi)$ ✓	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\neg\varphi$ | $\mathbb{T}\,\psi$	$\vee\mathbb{T}\,2$
5.	$\mathbb{T}\,\varphi$ | $\mathbb{T}\,\varphi$	$\to\mathbb{F}\,3$
6.	$\mathbb{F}\,\psi$ | $\mathbb{F}\,\psi$	$\to\mathbb{F}\,3$
7.	$\mathbb{F}\,\varphi$ | $\otimes$	$\neg\mathbb{T}\,4$
	$\otimes$	

Example 11.6. We can give tableaux for any number of signed formulas as assumptions. Often it is also necessary to apply more than one rule that allows branching; and in general a tableau can have any number of branches. For instance, consider a tableau for $\{\mathbb{T}\,\varphi \vee (\psi \wedge \chi), \mathbb{F}\,(\varphi \vee \psi) \wedge (\varphi \vee \chi)\}$. We start by applying the $\vee\mathbb{T}$ to the first assumption:

1.	$\mathbb{T}\,\varphi \vee (\psi \wedge \chi)$ ✓	Assumption
2.	$\mathbb{F}\,(\varphi \vee \psi) \wedge (\varphi \vee \chi)$	Assumption
3.	$\mathbb{T}\,\varphi$ | $\mathbb{T}\,\psi \wedge \chi$	$\vee\mathbb{T}\,1$

Now we can apply the $\wedge\mathbb{F}$ rule to line 2. We do this on both branches simultaneously, and can therefore check off line 2:

1.	$\mathbb{T}\,\varphi \vee (\psi \wedge \chi)$ ✓	Assumption
2.	$\mathbb{F}\,(\varphi \vee \psi) \wedge (\varphi \vee \chi)$ ✓	Assumption
3.	$\mathbb{T}\,\varphi$ | $\mathbb{T}\,\psi \wedge \chi$	$\vee\mathbb{T}\,1$
4.	$\mathbb{F}\,\varphi \vee \psi$ | $\mathbb{F}\,\varphi \vee \chi$ | $\mathbb{F}\,\varphi \vee \psi$ | $\mathbb{F}\,\varphi \vee \chi$	$\wedge\mathbb{F}\,2$

Now we can apply $\vee\mathbb{F}$ to all the branches containing $\varphi \vee \psi$:

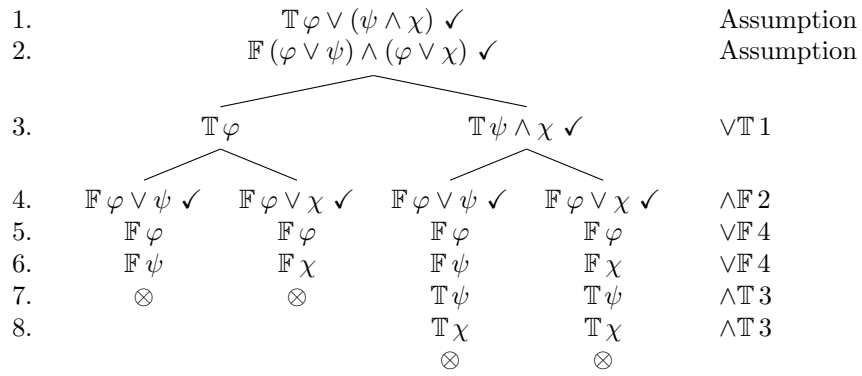
1.	$\mathbb{T}\,\varphi \vee (\psi \wedge \chi)$ ✓	Assumption
2.	$\mathbb{F}\,(\varphi \vee \psi) \wedge (\varphi \vee \chi)$ ✓	Assumption
3.	$\mathbb{T}\,\varphi$ | $\mathbb{T}\,\psi \wedge \chi$	$\vee\mathbb{T}\,1$
4.	$\mathbb{F}\,\varphi \vee \psi$ ✓ | $\mathbb{F}\,\varphi \vee \chi$ | $\mathbb{F}\,\varphi \vee \psi$ ✓ | $\mathbb{F}\,\varphi \vee \chi$	$\wedge\mathbb{F}\,2$
5.	$\mathbb{F}\,\varphi$ | $\mathbb{F}\,\varphi$	$\vee\mathbb{F}\,4$
6.	$\mathbb{F}\,\psi$ | $\mathbb{F}\,\psi$	$\vee\mathbb{F}\,4$
	$\otimes$	

The leftmost branch is now closed. Let's now apply $\vee\mathbb{F}$ to $\varphi \vee \chi$:

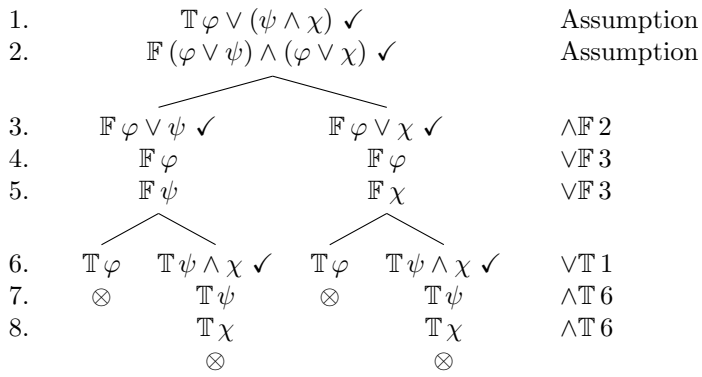
1.	$\mathbb{T}\,\varphi \vee (\psi \wedge \chi)$ ✓	Assumption
2.	$\mathbb{F}\,(\varphi \vee \psi) \wedge (\varphi \vee \chi)$ ✓	Assumption
3.	$\mathbb{T}\,\varphi$ | $\mathbb{T}\,\psi \wedge \chi$	$\vee\mathbb{T}\,1$
4.	$\mathbb{F}\,\varphi \vee \psi$ ✓ | $\mathbb{F}\,\varphi \vee \chi$ ✓ | $\mathbb{F}\,\varphi \vee \psi$ ✓ | $\mathbb{F}\,\varphi \vee \chi$ ✓	$\wedge\mathbb{F}\,2$
5.	$\mathbb{F}\,\varphi$ | | $\mathbb{F}\,\varphi$ |	$\vee\mathbb{F}\,4$
6.	$\mathbb{F}\,\psi$ | | $\mathbb{F}\,\psi$ |	$\vee\mathbb{F}\,4$
7.	$\otimes$ | $\mathbb{F}\,\varphi$ | $\mathbb{F}\,\varphi$	$\vee\mathbb{F}\,4$
8.	| $\mathbb{F}\,\chi$ | $\mathbb{F}\,\chi$	$\vee\mathbb{F}\,4$
	| $\otimes$ |	

Note that we moved the result of applying $\vee\mathbb{F}$ a second time below for clarity. In this instance it would not have been needed, since the justifications would have been the same.

Two branches remain open, and $\mathbb{T}\,\psi \wedge \chi$ on line 3 remains unchecked. We apply $\wedge\mathbb{T}$ to it to obtain a closed tableau:



For comparison, here's a closed tableau for the same set of assumptions in which the rules are applied in a different order:



Problem 11.1. Give closed tableaux of the following:

1. $\mathbb{T}\,\varphi \wedge (\psi \wedge \chi), \mathbb{F}\,(\varphi \wedge \psi) \wedge \chi$.
2. $\mathbb{T}\,\varphi \vee (\psi \vee \chi), \mathbb{F}\,(\varphi \vee \psi) \vee \chi$.
3. $\mathbb{T}\,\varphi \to (\psi \to \chi), \mathbb{F}\,\psi \to (\varphi \to \chi)$.
4. $\mathbb{T}\,\varphi, \mathbb{F}\,\neg\neg\varphi$.

Problem 11.2. Give closed tableaux of the following:

1. $\mathbb{T}\,(\varphi \vee \psi) \to \chi, \mathbb{F}\,\varphi \to \chi$.
2. $\mathbb{T}\,(\varphi \to \chi) \wedge (\psi \to \chi), \mathbb{F}\,(\varphi \vee \psi) \to \chi$.
3. $\mathbb{F}\,\neg(\varphi \wedge \neg\varphi)$.

4. $\mathbb{T}\,\psi \to \varphi, \mathbb{F}\,\neg\varphi \to \neg\psi$.
5. $\mathbb{F}\,(\varphi \to \neg\varphi) \to \neg\varphi$.
6. $\mathbb{F}\,\neg(\varphi \to \psi) \to \neg\psi$.
7. $\mathbb{T}\,\varphi \to \chi, \mathbb{F}\,\neg(\varphi \wedge \neg\chi)$.
8. $\mathbb{T}\,\varphi \wedge \neg\chi, \mathbb{F}\,\neg(\varphi \to \chi)$.
9. $\mathbb{T}\,\varphi \vee \psi, \neg\psi, \mathbb{F}\,\varphi$.
10. $\mathbb{T}\,\neg\varphi \vee \neg\psi, \mathbb{F}\,\neg(\varphi \wedge \psi)$.
11. $\mathbb{F}\,(\neg\varphi \wedge \neg\psi) \to \neg(\varphi \vee \psi)$.
12. $\mathbb{F}\,\neg(\varphi \vee \psi) \to (\neg\varphi \wedge \neg\psi)$.

Problem 11.3. Give closed tableaux of the following:

1. $\mathbb{T}\,\neg(\varphi \to \psi), \mathbb{F}\,\varphi$.
2. $\mathbb{T}\,\neg(\varphi \wedge \psi), \mathbb{F}\,\neg\varphi \vee \neg\psi$.
3. $\mathbb{T}\,\varphi \to \psi, \mathbb{F}\,\neg\varphi \vee \psi$.
4. $\mathbb{F}\,\neg\neg\varphi \to \varphi$.
5. $\mathbb{T}\,\varphi \to \psi, \mathbb{T}\,\neg\varphi \to \psi, \mathbb{F}\,\psi$.
6. $\mathbb{T}\,(\varphi \wedge \psi) \to \chi, \mathbb{F}\,(\varphi \to \chi) \vee (\psi \to \chi)$.
7. $\mathbb{T}\,(\varphi \to \psi) \to \varphi, \mathbb{F}\,\varphi$.
8. $\mathbb{F}\,(\varphi \to \psi) \vee (\psi \to \chi)$.

---

content/propositional-logic/../first-order-logic/tableaux/proof-theoretic-notions.tex

11.5 Proof-Theoretic Notions

pl:tab:ptn:sec

This section collects the definitions of the provability relation and consistency for tableaux.

explanation

Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the existence of certain closed tableaux. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

Definition 11.7 (Theorems). A sentence φ is a *theorem* if there is a closed tableau for $\mathbb{F}\,\varphi$. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 11.8 (Derivability). A sentence φ is *derivable from* a set of sentences Γ, $\Gamma \vdash \varphi$ iff there is a finite set $\{\psi_1, \ldots, \psi_n\} \subseteq \Gamma$ and a closed tableau for the set

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}.$$

If φ is not derivable from Γ we write $\Gamma \nvdash \varphi$.

Definition 11.9 (Consistency). A set of sentences Γ is *inconsistent* iff there is a finite set $\{\psi_1, \ldots, \psi_n\} \subseteq \Gamma$ and a closed tableau for the set

$$\{\mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}.$$

If Γ is not inconsistent, we say it is *consistent*.

Proposition 11.10 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.* pl:tab:ptn: prop:reflexivity

Proof. If $\varphi \in \Gamma$, $\{\varphi\}$ is a finite subset of Γ and the tableau

1.	$\mathbb{F}\,\varphi$	Assumption
2.	$\mathbb{T}\,\varphi$	Assumption
	$\otimes$	

is closed. □

Proposition 11.11 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.* pl:tab:ptn: prop:monotonicity

Proof. Any finite subset of Γ is also a finite subset of Δ. □

Proposition 11.12 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.* pl:tab:ptn: prop:transitivity

Proof. If $\{\varphi\} \cup \Delta \vdash \psi$, then there is a finite subset $\Delta_0 = \{\chi_1, \ldots, \chi_n\} \subseteq \Delta$ such that

$$\{\mathbb{F}\,\psi, \mathbb{T}\,\varphi, \mathbb{T}\,\chi_1, \ldots, \mathbb{T}\,\chi_n\}$$

has a closed tableau. If $\Gamma \vdash \varphi$ then there are $\theta_1, \ldots, \theta_m \subseteq \Gamma$ such that

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\theta_1, \ldots, \mathbb{T}\,\theta_m\}$$

has a closed tableau.

Now consider the tableau with assumptions

$$\mathbb{F}\,\psi, \mathbb{T}\,\chi_1, \ldots, \mathbb{T}\,\chi_n, \mathbb{T}\,\theta_1, \ldots, \mathbb{T}\,\theta_m.$$

Apply the Cut rule on φ. This generates two branches, one has $\mathbb{T}\varphi$ in it, the other $\mathbb{F}\varphi$. Thus, on the one branch, all of

$$\{\mathbb{F}\psi, \mathbb{T}\varphi, \mathbb{T}\chi_1, \ldots, \mathbb{T}\chi_n\}$$

are available. Since there is a closed tableau for these assumptions, we can attach it to that branch; every branch through $\mathbb{T}\varphi$ closes. On the other branch, all of

$$\{\mathbb{F}\varphi, \mathbb{T}\theta_1, \ldots, \mathbb{T}\theta_m\}$$

are available, so we can also complete the other side to obtain a closed tableau. This shows $\Gamma \cup \Delta \vdash \psi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \ldots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

pl:tab:ptn: prop:incons

Proposition 11.13. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every sentence φ.*

Proof. Exercise. □

Problem 11.4. Prove Proposition 11.13

pl:tab:ptn: prop:proves-compact

Proposition 11.14 (Compactness).

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof. 1. If $\Gamma \vdash \varphi$, then there is a finite subset $\Gamma_0 = \{\psi_1, \ldots, \psi_n\}$ and a closed tableau for

$$\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n\}$$

This tableau also shows $\Gamma_0 \vdash \varphi$.

2. If Γ is inconsistent, then for some finite subset $\Gamma_0 = \{\psi_1, \ldots, \psi_n\}$ there is a closed tableau for

$$\{\mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n\}$$

This closed tableau shows that Γ_0 is inconsistent. □

---

content/propositional-logic/../first-order-logic/tableaux/provability-consistency.tex

11.6 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem. pl:tab:prv: sec

Proposition 11.15. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.* pl:tab:prv: prop:provability-contr

Proof. There are finite $\Gamma_0 = \{\psi_1, \ldots, \psi_n\}$ and $\Gamma_1 = \{\chi_1, \ldots, \chi_n\} \subseteq \Gamma$ such that

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$
$$\{\mathbb{T}\,\varphi, \mathbb{T}\,\chi_1, \ldots, \mathbb{T}\,\chi_m\}$$

have closed tableaux. Using the Cut rule on φ we can combine these into a single closed tableau that shows $\Gamma_0 \cup \Gamma_1$ is inconsistent. Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$, hence Γ is inconsistent. □

Proposition 11.16. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.* pl:tab:prv: prop:prov-incons

Proof. First suppose $\Gamma \vdash \varphi$, i.e., there is a closed tableau for

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$

Using the $\neg\mathbb{T}$ rule, this can be turned into a closed tableau for

$$\{\mathbb{T}\,\neg\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}.$$

On the other hand, if there is a closed tableau for the latter, we can turn it into a closed tableau of the former by removing every formula that results from $\neg\mathbb{T}$ applied to the first assumption $\mathbb{T}\,\neg\varphi$ as well as that assumption, and adding the assumption $\mathbb{F}\,\varphi$. For if a branch was closed before because it contained the conclusion of $\neg\mathbb{T}$ applied to $\mathbb{T}\,\neg\varphi$, i.e., $\mathbb{F}\,\varphi$, the corresponding branch in the new tableau is also closed. If a branch in the old tableau was closed because it contained the assumption $\mathbb{T}\,\neg\varphi$ as well as $\mathbb{F}\,\neg\varphi$ we can turn it into a closed branch by applying $\neg\mathbb{F}$ to $\mathbb{F}\,\neg\varphi$ to obtain $\mathbb{T}\,\varphi$. This closes the branch since we added $\mathbb{F}\,\varphi$ as an assumption. □

Problem 11.5. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

Proposition 11.17. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.* pl:tab:prv: prop:explicit-inc

Proof. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there are $\psi_1, \ldots, \psi_n \in \Gamma$ such that

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$

has a closed tableau. Replace the assumption $\mathbb{F}\,\varphi$ by $\mathbb{T}\,\neg\varphi$, and insert the conclusion of $\neg\mathbb{T}$ applied to $\mathbb{F}\,\varphi$ after the assumptions. Any sentence in the tableau justified by appeal to line 1 in the old tableau is now justified by appeal to line $n+1$. So if the old tableau was closed, the new one is. It shows that Γ is inconsistent, since all assumptions are in Γ. □

pl:tab:prv: prop:provability-exhaustive

Proposition 11.18. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

Proof. If there are $\psi_1, \ldots, \psi_n \in \Gamma$ and $\chi_1, \ldots, \chi_m \in \Gamma$ such that

$$\{\mathbb{T}\varphi, \mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n\} \text{ and}$$
$$\{\mathbb{T}\neg\varphi, \mathbb{T}\chi_1, \ldots, \mathbb{T}\chi_m\}$$

both have closed tableaux, we can construct a single, combined tableau that shows that Γ is inconsistent by using as assumptions $\mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n$ together with $\mathbb{T}\chi_1, \ldots, \mathbb{T}\chi_m$, followed by an application of the Cut rule. This yields two branches, one starting with $\mathbb{T}\varphi$, the other with $\mathbb{F}\varphi$.

On the left left side, add the part of the first tableau below its assumptions. Here, every rule application is still correct, since each of the assumptions of the first tableau, including $\mathbb{T}\varphi$, is available. Thus, every branch below $\mathbb{T}\varphi$ closes.

On the right side, add the part of the second tableau below its assumption, with the results of any applications of $\neg\mathbb{T}$ to $\mathbb{T}\neg\varphi$ removed. The conclusion of $\neg\mathbb{T}$ to $\mathbb{T}\neg\varphi$ is $\mathbb{F}\varphi$, which is nevertheless available, as it is the conclusion of the Cut rule on the right side of the combined tableau.

If a branch in the second tableau was closed because it contained the assumption $\mathbb{T}\neg\varphi$ (which no longer appears as an assumption in the combined tableau) as well as $\mathbb{F}\neg\varphi$, we can applying $\neg\mathbb{F}$ to $\mathbb{F}\neg\varphi$ to obtain $\mathbb{T}\varphi$. Now the corresponding branch in the combined tableau also closes, because it contains the right-hand conclusion of the Cut rule, $\mathbb{F}\varphi$. If a branch in the second tableau closed for any other reason, the corresponding branch in the combined tableau also closes, since any signed formulas other than $\mathbb{T}\neg\varphi$ occurring on the branch in the old, second tableau also occur on the corresponding branch in the combined tableau. □

---

content/propositional-logic/../first-order-logic/tableaux/provability-propositional.tex

11.7 Derivability and the Propositional Connectives

pl:tab:ppr: sec

We establish that the derivability relation $\vdash$ of tableaux is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. explanation

pl:tab:ppr: prop:provability-land

Proposition 11.19.

pl:tab:ppr: prop:provability-land-left

1. *Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$.*

pl:tab:ppr: prop:provability-land-right

2. *$\varphi, \psi \vdash \varphi \wedge \psi$.*

Proof. 1. Both $\{\mathbb{F}\varphi, \mathbb{T}\varphi \wedge \psi\}$ and $\{\mathbb{F}\psi, \mathbb{T}\varphi \wedge \psi\}$ have closed tableaux

1.	$\mathbb{F}\,\varphi$	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	Assumption
3.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}\,2$
4.	$\mathbb{T}\psi$	$\wedge\mathbb{T}\,2$
	$\otimes$	

1.	$\mathbb{F}\,\psi$	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	Assumption
3.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}\,2$
4.	$\mathbb{T}\psi$	$\wedge\mathbb{T}\,2$
	$\otimes$	

2. Here is a closed tableau for $\{\mathbb{T}\,\varphi, \mathbb{T}\,\psi, \mathbb{F}\,\varphi \wedge \psi\}$:

1.	$\mathbb{F}\,\varphi \wedge \psi$		Assumption
2.	$\mathbb{T}\,\varphi$		Assumption
3.	$\mathbb{T}\,\psi$		Assumption
4.	$\mathbb{F}\,\varphi$	$\mathbb{F}\,\psi$	$\wedge\mathbb{F}\,1$
	$\otimes$	$\otimes$	

Proposition 11.20.

pl:tab:ppr: prop:provability-lor

1. $\{\varphi \vee \psi, \neg\varphi, \neg\psi\}$ is inconsistent.

2. Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.

Proof. 1. We give a closed tableau of $\{\mathbb{T}\,\varphi \vee \psi, \mathbb{T}\,\neg\varphi, \mathbb{T}\,\neg\psi\}$:

1.	$\mathbb{T}\,\varphi \vee \psi$		Assumption
2.	$\mathbb{T}\,\neg\varphi$		Assumption
3.	$\mathbb{T}\,\neg\psi$		Assumption
4.	$\mathbb{F}\,\varphi$		$\neg\mathbb{T}\,2$
5.	$\mathbb{F}\,\psi$		$\neg\mathbb{T}\,3$
6.	$\mathbb{T}\,\varphi$	$\mathbb{T}\,\psi$	$\vee\mathbb{T}\,1$
	$\otimes$	$\otimes$	

2. Both $\{\mathbb{F}\,\varphi \vee \psi, \mathbb{T}\,\varphi\}$ and $\{\mathbb{F}\,\varphi \vee \psi, \mathbb{T}\,\psi\}$ have closed tableaux:

1.	$\mathbb{F}\,\varphi \vee \psi$	Assumption
2.	$\mathbb{T}\,\varphi$	Assumption
3.	$\mathbb{F}\varphi$	$\vee\mathbb{F}\,1$
4.	$\mathbb{F}\psi$	$\vee\mathbb{F}\,1$
	$\otimes$	

1.	$\mathbb{F}\,\varphi \vee \psi$	Assumption
2.	$\mathbb{T}\,\psi$	Assumption
3.	$\mathbb{F}\varphi$	$\vee\mathbb{F}\,1$
4.	$\mathbb{F}\psi$	$\vee\mathbb{F}\,1$
	$\otimes$	

pl:tab:ppr: prop:provability-lif pl:tab:ppr: prop:provability-lif-left pl:tab:ppr: prop:provability-lif-right

Proposition 11.21.

1. $\varphi, \varphi \to \psi \vdash \psi$.
2. *Both* $\neg\varphi \vdash \varphi \to \psi$ *and* $\psi \vdash \varphi \to \psi$.

Proof. 1. $\{\mathbb{F}\,\psi, \mathbb{T}\,\varphi \to \psi, \mathbb{T}\,\varphi\}$ has a closed tableau:

1.	$\mathbb{F}\,\psi$	Assumption
2.	$\mathbb{T}\,\varphi \to \psi$	Assumption
3.	$\mathbb{T}\,\varphi$	Assumption
4.	$\mathbb{F}\,\varphi$ $\quad$ $\mathbb{T}\,\psi$	$\to\mathbb{T}\,2$
	$\otimes$ $\quad$ $\otimes$	

2. Both $\{\mathbb{F}\,\varphi \to \psi, \mathbb{T}\,\neg\varphi\}$ and $\{\mathbb{F}\,\varphi \to \psi, \mathbb{T}\,\psi\}$ have closed tableaux:

1.	$\mathbb{F}\,\varphi \to \psi$	Assumption
2.	$\mathbb{T}\,\neg\varphi$	Assumption
3.	$\mathbb{T}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{F}\,\psi$	$\to\mathbb{F}\,1$
5.	$\mathbb{F}\,\varphi$	$\neg\mathbb{T}\,2$
	$\otimes$	

1.	$\mathbb{F}\,\varphi \to \psi$	Assumption
2.	$\mathbb{T}\,\psi$	Assumption
3.	$\mathbb{T}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{F}\,\psi$	$\to\mathbb{F}\,1$
	$\otimes$	

---

content/propositional-logic/../first-order-logic/tableaux/soundness.tex

11.8 Soundness

explanation

A derivation system, such as tableaux, is *sound* if it cannot derive things that do not actually hold. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that pl:tab:sou:sec

1. every derivable φ is a tautology;
2. if a sentence is derivable from some others, it is also a consequence of them;
3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Because all these proof-theoretic properties are defined via closed tableaux of some kind or other, proving (1)–(3) above requires proving something about the semantic properties of closed tableaux. We will first define what it means for a signed formula to be satisfied in a structure, and then show that if a tableau is closed, no structure satisfies all its assumptions. (1)–(3) then follow as corollaries from this result.

Definition 11.22. A valuation $\mathfrak{v}$ *satisfies* a signed formula $\mathbb{T}\varphi$ iff $\mathfrak{v} \vDash \varphi$, and it satisfies $\mathbb{F}\varphi$ iff $\mathfrak{v} \nvDash \varphi$. $\mathfrak{v}$ satisfies a set of signed formulas Γ iff it satisfies every $S\varphi \in \Gamma$. Γ is *satisfiable* if there is a valuation that satisfies it, and *unsatisfiable* otherwise.

Theorem 11.23 (Soundness). *If Γ has a closed tableau, Γ is unsatisfiable.* pl:tab:sou:thm:tableau-soundness

Proof. Let's call a branch of a tableau satisfiable iff the set of signed formulas on it is satisfiable, and let's call a tableau satisfiable if it contains at least one satisfiable branch.

We show the following: Extending a satisfiable tableau by one of the rules of inference always results in a satisfiable tableau. This will prove the theorem: any closed tableau results by applying rules of inference to the tableau consisting only of assumptions from Γ. So if Γ were satisfiable, any tableau for it would be satisfiable. A closed tableau, however, is clearly not satisfiable: every branch contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, and no structure can both satisfy and not satisfy φ.

Suppose we have a satisfiable tableau, i.e., a tableau with at least one satisfiable branch. Applying a rule of inference either adds signed formulas to a branch, or splits a branch in two. If the tableau has a satisfiable branch which is not extended by the rule application in question, it remains a satisfiable branch in the extended tableau, so the extended tableau is satisfiable. So we only have to consider the case where a rule is applied to a satisfiable branch.

Let Γ be the set of signed formulas on that branch, and let $S\varphi \in \Gamma$ be the signed formula to which the rule is applied. If the rule does not result in a split branch, we have to show that the extended branch, i.e., Γ together with the conclusions of the rule, is still satisfiable. If the rule results in a split branch, we have to show that at least one of the two resulting branches is satisfiable.

First, we consider the possible inferences that do not result in a split branch.

1. The branch is expanded by applying $\neg\mathbb{T}$ to $\mathbb{T}\neg\psi \in \Gamma$. Then the extended branch contains the signed formulas $\Gamma \cup \{\mathbb{F}\psi\}$. Suppose $\mathfrak{v} \vDash \Gamma$. In particular, $\mathfrak{v} \vDash \neg\psi$. Thus, $\mathfrak{v} \nvDash \psi$, i.e., $\mathfrak{v}$ satisfies $\mathbb{F}\psi$.

2. The branch is expanded by applying $\neg\mathbb{F}$ to $\mathbb{F}\neg\psi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\wedge\mathbb{T}$ to $\mathbb{T}\psi \wedge \chi \in \Gamma$, which results in two new signed formulas on the branch: $\mathbb{T}\psi$ and $\mathbb{T}\chi$. Suppose $\mathfrak{v} \vDash \Gamma$, in particular $\mathfrak{v} \vDash \psi \wedge \chi$. Then $\mathfrak{v} \vDash \psi$ and $\mathfrak{v} \vDash \chi$. This means that $\mathfrak{v}$ satisfies both $\mathbb{T}\psi$ and $\mathbb{T}\chi$.

4. The branch is expanded by applying $\vee\mathbb{F}$ to $\mathbb{F}\psi \vee \chi \in \Gamma$: Exercise.

5. The branch is expanded by applying $\to\mathbb{F}$ to $\mathbb{F}\psi \to \chi \in \Gamma$: This results in two new signed formulas on the branch: $\mathbb{T}\psi$ and $\mathbb{F}\chi$. Suppose $\mathfrak{v} \vDash \Gamma$, in particular $\mathfrak{v} \nvDash \psi \to \chi$. Then $\mathfrak{v} \vDash \psi$ and $\mathfrak{v} \nvDash \chi$. This means that $\mathfrak{v}$ satisfies both $\mathbb{T}\psi$ and $\mathbb{F}\chi$.

Now let's consider the possible inferences that result in a split branch.

1. The branch is expanded by applying $\wedge\mathbb{F}$ to $\mathbb{F}\psi \wedge \chi \in \Gamma$, which results in two branches, a left one continuing through $\mathbb{F}\psi$ and a right one through $\mathbb{F}\chi$. Suppose $\mathfrak{v} \vDash \Gamma$, in particular $\mathfrak{v} \nvDash \psi \wedge \chi$. Then $\mathfrak{v} \nvDash \psi$ or $\mathfrak{v} \nvDash \chi$. In the former case, $\mathfrak{v}$ satisfies $\mathbb{F}\psi$, i.e., $\mathfrak{v}$ satisfies the formulas on the left branch. In the latter, $\mathfrak{v}$ satisfies $\mathbb{F}\chi$, i.e., $\mathfrak{v}$ satisfies the formulas on the right branch.

2. The branch is expanded by applying $\vee\mathbb{T}$ to $\mathbb{T}\psi \vee \chi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\to\mathbb{T}$ to $\mathbb{T}\psi \to \chi \in \Gamma$: Exercise.

4. The branch is expanded by Cut: This results in two branches, one containing $\mathbb{T}\psi$, the other containing $\mathbb{F}\psi$. Since $\mathfrak{v} \vDash \Gamma$ and either $\mathfrak{v} \vDash \psi$ or $\mathfrak{v} \nvDash \psi$, $\mathfrak{v}$ satisfies either the left or the right branch. □

Problem 11.6. Complete the proof of Theorem 11.23.

pl:tab:sou: cor:weak-soundness **Corollary 11.24.** *If $\vdash \varphi$ then φ is a tautology.*

pl:tab:sou: cor:entailment-soundness **Corollary 11.25.** *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.*

Proof. If $\Gamma \vdash \varphi$ then for some $\psi_1, \ldots, \psi_n \in \Gamma$, $\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n\}$ has a closed tableau. By Theorem 11.23, every valuation $\mathfrak{v}$ either makes some ψ_i false or makes φ true. Hence, if $\mathfrak{v} \vDash \Gamma$ then also $\mathfrak{v} \vDash \varphi$. □

Corollary 11.26. *If Γ is satisfiable, then it is consistent.* pl:tab:sou: cor:consistency-soundness

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then there are $\psi_1, \ldots, \psi_n \in \Gamma$ and a closed tableau for $\{\mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n\}$. By Theorem 11.23, there is no $\mathfrak{v}$ such that $\mathfrak{v} \vDash \psi_i$ for all $i = 1, \ldots, n$. But then Γ is not satisfiable. □

Chapter 12

Axiomatic Derivations

No effort has been made yet to ensure that the material in this chapter respects various tags indicating which connectives and quantifiers are primitive or defined: all are assumed to be primitive, except $\leftrightarrow$ which is assumed to be defined. If the FOL tag is true, we produce a version with quantifiers, otherwise without.

content/propositional-logic/../first-order-logic/axiomatic-deduction/rules-and-proo

12.1 Rules and Derivations

explanation Axiomatic derivations are perhaps the simplest derivation system for logic. A derivation is just a sequence of formulas. To count as a derivation, every formula in the sequence must either be an instance of an axiom, or must follow from one or more formulas that precede it in the sequence by a rule of inference. A derivation derives its last formula. pl:axd:rul: sec

Definition 12.1 (Derivability). If Γ is a set of formulas of $\mathcal{L}$ then a *derivation* from Γ is a finite sequence $\varphi_1, \ldots, \varphi_n$ of formulas where for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. φ_i follows from some φ_j (and φ_k) with $j < i$ (and $k < i$) by a rule of inference.

What counts as a correct derivation depends on which inference rules we allow (and of course what we take to be axioms). And an inference rule is an if-then statement that tells us that, under certain conditions, a step A_i in a derivation is a correct inference step.

Definition 12.2 (Rule of inference). A *rule of inference* gives a sufficient condition for what counts as a correct inference step in a derivation from Γ.

For instance, since any one-element sequence φ with $\varphi \in \Gamma$ trivially counts as a derivation, the following might be a very simple rule of inference:

> If $\varphi \in \Gamma$, then φ is always a correct inference step in any derivation from Γ.

Similarly, if φ is one of the axioms, then φ by itself is a derivation, and so this is also a rule of inference:

> If φ is an axiom, then φ is a correct inference step.

It gets more interesting if the rule of inference appeals to formulas that appear before the step considered. The following rule is called *modus ponens:*

> If $\psi \to \varphi$ and ψ occur higher up in the derivation, then φ is a correct inference step.

If this is the only rule of inference, then our definition of derivation above amounts to this: $\varphi_1, \ldots, \varphi_n$ is a derivation iff for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. for some $j < i$, φ_j is $\psi \to \varphi_i$, and for some $k < i$, φ_k is ψ.

The last clause says that φ_i follows from φ_j ($\psi \to \varphi_i$) and φ_k (ψ) by modus ponens. If we can go from 1 to n, and each time we find a formula φ_i that is either in Γ, an axiom, or which a rule of inference tells us that it is a correct inference step, then the entire sequence counts as a correct derivation.

Definition 12.3 (Derivability). A formula φ is *derivable* from Γ, written $\Gamma \vdash \varphi$, if there is a derivation from Γ ending in φ.

Definition 12.4 (Theorems). A formula φ is a *theorem* if there is a derivation of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

---

content/propositional-logic/../first-order-logic/axiomatic-deduction/axioms-rules-p

12.2 Axioms and Rules for the Propositional Connectives

pl:axd:prp:sec

Definition 12.5 (Axioms). The set of Ax_0 of *axioms* for the propositional connectives comprises all formulas of the following forms:

$$(\varphi \wedge \psi) \to \varphi \tag{12.1}$$ pl:axd:prp:ax:land1

$$(\varphi \wedge \psi) \to \psi \tag{12.2}$$ pl:axd:prp:ax:land2

$$\varphi \to (\psi \to (\varphi \wedge \psi)) \tag{12.3}$$ pl:axd:prp:ax:land3

$$\varphi \to (\varphi \vee \psi) \tag{12.4}$$ pl:axd:prp:ax:lor1

$$\varphi \to (\psi \vee \varphi) \tag{12.5}$$ pl:axd:prp:ax:lor2

$$(\varphi \to \chi) \to ((\psi \to \chi) \to ((\varphi \vee \psi) \to \chi)) \tag{12.6}$$ pl:axd:prp:ax:lor3

$$\varphi \to (\psi \to \varphi) \tag{12.7}$$ pl:axd:prp:ax:lif1

$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \tag{12.8}$$ pl:axd:prp:ax:lif2

$$(\varphi \to \psi) \to ((\varphi \to \neg\psi) \to \neg\varphi) \tag{12.9}$$ pl:axd:prp:ax:lnot1

$$\neg\varphi \to (\varphi \to \psi) \tag{12.10}$$ pl:axd:prp:ax:lnot2

$$\top \tag{12.11}$$ pl:axd:prp:ax:ltrue

$$\bot \to \varphi \tag{12.12}$$ pl:axd:prp:ax:lfalse1

$$(\varphi \to \bot) \to \neg\varphi \tag{12.13}$$ pl:axd:prp:ax:lfalse2

$$\neg\neg\varphi \to \varphi \tag{12.14}$$ pl:axd:prp:ax:dne

Definition 12.6 (Modus ponens). If ψ and $\psi \to \varphi$ already occur in a derivation, then φ is a correct inference step.

We'll abbreviate the rule modus ponens as "MP."

---

content/propositional-logic/../first-order-logic/axiomatic-deduction/proving-things

12.3 Examples of Derivations

pl:axd:pro:sec

Example 12.7. Suppose we want to prove $(\neg\theta \vee \alpha) \to (\theta \to \alpha)$. Clearly, this is not an instance of any of our axioms, so we have to use the MP rule to derive it. Our only rule is MP, which given φ and $\varphi \to \psi$ allows us to justify ψ. One

strategy would be to use eq. (12.6) with φ being $\neg\theta$, ψ being α, and χ being $\theta \to \alpha$, i.e., the instance

$$(\neg\theta \to (\theta \to \alpha)) \to ((\alpha \to (\theta \to \alpha)) \to ((\neg\theta \vee \alpha) \to (\theta \to \alpha))).$$

Why? Two applications of MP yield the last part, which is what we want. And we easily see that $\neg\theta \to (\theta \to \alpha)$ is an instance of eq. (12.10), and $\alpha \to (\theta \to \alpha)$ is an instance of eq. (12.7). So our derivation is:

1.	$\neg\theta \to (\theta \to \alpha)$	eq. (12.10)
2.	$(\neg\theta \to (\theta \to \alpha)) \to$ $((\alpha \to (\theta \to \alpha)) \to ((\neg\theta \vee \alpha) \to (\theta \to \alpha)))$	eq. (12.6)
3.	$(\alpha \to (\theta \to \alpha)) \to ((\neg\theta \vee \alpha) \to (\theta \to \alpha))$	1, 2, MP
4.	$\alpha \to (\theta \to \alpha)$	eq. (12.7)
5.	$(\neg\theta \vee \alpha) \to (\theta \to \alpha)$	3, 4, MP

pl:axd:pro: ex:identity

Example 12.8. Let's try to find a derivation of $\theta \to \theta$. It is not an instance of an axiom, so we have to use MP to derive it. eq. (12.7) is an axiom of the form $\varphi \to \psi$ to which we could apply MP. To be useful, of course, the ψ which MP would justify as a correct step in this case would have to be $\theta \to \theta$, since this is what we want to derive. That means φ would also have to be θ, i.e., we might look at this instance of eq. (12.7):

$$\theta \to (\theta \to \theta)$$

In order to apply MP, we would also need to justify the corresponding second premise, namely φ. But in our case, that would be θ, and we won't be able to derive θ by itself. So we need a different strategy.

The other axiom involving just $\to$ is eq. (12.8), i.e.,

$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$$

We could get to the last nested conditional by applying MP twice. Again, that would mean that we want an instance of eq. (12.8) where $\varphi \to \chi$ is $\theta \to \theta$, the formula we are aiming for. Then of course, φ and χ are both θ. How should we pick ψ so that both $\varphi \to (\psi \to \chi)$ and $\varphi \to \psi$, i.e., in our case $\theta \to (\psi \to \theta)$ and $\theta \to \psi$, are also derivable? Well, the first of these is already an instance of eq. (12.7), whatever we decide ψ to be. And $\theta \to \psi$ would be another instance of eq. (12.7) if ψ were $(\theta \to \theta)$. So, our derivation is:

1.	$\theta \to ((\theta \to \theta) \to \theta)$	eq. (12.7)
2.	$(\theta \to ((\theta \to \theta) \to \theta)) \to$ $((\theta \to (\theta \to \theta)) \to (\theta \to \theta))$	eq. (12.8)
3.	$(\theta \to (\theta \to \theta)) \to (\theta \to \theta)$	1, 2, MP
4.	$\theta \to (\theta \to \theta)$	eq. (12.7)
5.	$\theta \to \theta$	3, 4, MP

pl:axd:pro: ex:chain

Example 12.9. Sometimes we want to show that there is a derivation of some formula from some other formulas Γ. For instance, let's show that we can derive $\varphi \to \chi$ from $\Gamma = \{\varphi \to \psi, \psi \to \chi\}$.

1.	$\varphi \to \psi$	Hyp
2.	$\psi \to \chi$	Hyp
3.	$(\psi \to \chi) \to (\varphi \to (\psi \to \chi))$	eq. (12.7)
4.	$\varphi \to (\psi \to \chi)$	2, 3, MP
5.	$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$	eq. (12.8)
6.	$((\varphi \to \psi) \to (\varphi \to \chi))$	4, 5, MP
7.	$\varphi \to \chi$	1, 6, MP

The lines labelled "Hyp" (for "hypothesis") indicate that the formula on that line is an element of Γ.

Proposition 12.10. *If $\Gamma \vdash \varphi \to \psi$ and $\Gamma \vdash \psi \to \chi$, then $\Gamma \vdash \varphi \to \chi$*

pl:axd:pro:prop:chain

Proof. Suppose $\Gamma \vdash \varphi \to \psi$ and $\Gamma \vdash \psi \to \chi$. Then there is a derivation of $\varphi \to \psi$ from Γ; and a derivation of $\psi \to \chi$ from Γ as well. Combine these into a single derivation by concatenating them. Now add lines 3–7 of the derivation in the preceding example. This is a derivation of $\varphi \to \chi$—which is the last line of the new derivation—from Γ. Note that the justifications of lines 4 and 7 remain valid if the reference to line number 1 is replaced by reference to the last line of the derivation of $\varphi \to \psi$, and reference to line number 2 by reference to the last line of the derivation of $\psi \to \chi$. □

Problem 12.1. Show that the following hold by exhibiting derivations from the axioms:

1. $(\varphi \wedge \psi) \to (\psi \wedge \varphi)$
2. $((\varphi \wedge \psi) \to \chi) \to (\varphi \to (\psi \to \chi))$
3. $\neg(\varphi \vee \psi) \to \neg\varphi$

---

content/propositional-logic/../first-order-logic/axiomatic-deduction/proof-theoreti

12.4 Proof-Theoretic Notions

explanation

pl:axd:ptn:sec

Just as we've defined a number of important semantic notions (tautology, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain formulas. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

Definition 12.11 (Derivability). A formula φ is *derivable* from Γ, written $\Gamma \vdash \varphi$, if there is a derivation from Γ ending in φ.

Definition 12.12 (Theorems). A formula φ is a *theorem* if there is a derivation of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 12.13 (Consistency). A set Γ of formulas is *consistent* if and only if $\Gamma \nvdash \bot$; it is *inconsistent* otherwise.

pl:axd:ptn: prop:reflexivity **Proposition 12.14 (Reflexivity).** *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.*

Proof. The formula φ by itself is a derivation of φ from Γ. □

pl:axd:ptn: prop:monotonicity **Proposition 12.15 (Monotonicity).** *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.*

Proof. Any derivation of φ from Γ is also a derivation of φ from Δ. □

pl:axd:ptn: prop:transitivity **Proposition 12.16 (Transitivity).** *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.*

Proof. Suppose $\{\varphi\} \cup \Delta \vdash \psi$. Then there is a derivation $\psi_1, \ldots, \psi_l = \psi$ from $\{\varphi\} \cup \Delta$. Some of the steps in that derivation will be correct because of a rule which refers to a prior line $\psi_i = \varphi$. By hypothesis, there is a derivation of φ from Γ, i.e., a derivation $\varphi_1, \ldots, \varphi_k = \varphi$ where every φ_i is an axiom, an element of Γ, or correct by a rule of inference. Now consider the sequence

$$\varphi_1, \ldots, \varphi_k = \varphi, \psi_1, \ldots, \psi_l = \psi.$$

This is a correct derivation of ψ from $\Gamma \cup \Delta$ since every $B_i = \varphi$ is now justified by the same rule which justifies $\varphi_k = \varphi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \ldots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

pl:axd:ptn: prop:incons **Proposition 12.17.** *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every φ.*

Proof. Exercise. □

Problem 12.2. Prove Proposition 12.17.

pl:axd:ptn: prop:proves-compact **Proposition 12.18 (Compactness).**

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof. 1. If $\Gamma \vdash \varphi$, then there is a finite sequence of formulas $\varphi_1, \ldots, \varphi_n$ so that $\varphi \equiv \varphi_n$ and each φ_i is either a logical axiom, an element of Γ or follows from previous formulas by modus ponens. Take Γ_0 to be those φ_i which are in Γ. Then the derivation is likewise a derivation from Γ_0, and so $\Gamma_0 \vdash \varphi$.

2. This is the contrapositive of (1) for the special case $\varphi \equiv \bot$. □

---

content/propositional-logic/../first-order-logic/axiomatic-deduction/deduction-theorem.tex

12.5 The Deduction Theorem

pl:axd:ded:sec

As we've seen, giving derivations in an axiomatic system is cumbersome, and derivations may be hard to find. Rather than actually write out long lists of formulas, it is generally easier to argue that such derivations exist, by making use of a few simple results. We've already established three such results: Proposition 12.14 says we can always assert that $\Gamma \vdash \varphi$ when we know that $\varphi \in \Gamma$. Proposition 12.15 says that if $\Gamma \vdash \varphi$ then also $\Gamma \cup \{\psi\} \vdash \varphi$. And Proposition 12.16 implies that if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. Here's another simple result, a "meta"-version of modus ponens:

Proposition 12.19. *If $\Gamma \vdash \varphi$ and $\Gamma \vdash \varphi \to \psi$, then $\Gamma \vdash \psi$.*

pl:axd:ded:prop:mp

Proof. We have that $\{\varphi, \varphi \to \psi\} \vdash \psi$:

1.	φ	Hyp.
2.	$\varphi \to \psi$	Hyp.
3.	ψ	1, 2, MP

By Proposition 12.16, $\Gamma \vdash \psi$. □

The most important result we'll use in this context is the deduction theorem:

Theorem 12.20 (Deduction Theorem). *$\Gamma \cup \{\varphi\} \vdash \psi$ if and only if $\Gamma \vdash \varphi \to \psi$.*

pl:axd:ded:thm:deduction-thm

Proof. The "if" direction is immediate. If $\Gamma \vdash \varphi \to \psi$ then also $\Gamma \cup \{\varphi\} \vdash \varphi \to \psi$ by Proposition 12.15. Also, $\Gamma \cup \{\varphi\} \vdash \varphi$ by Proposition 12.14. So, by Proposition 12.19, $\Gamma \cup \{\varphi\} \vdash \psi$.

For the "only if" direction, we proceed by induction on the length of the derivation of ψ from $\Gamma \cup \{\varphi\}$.

For the induction basis, we prove the claim for every derivation of length 1. A derivation of ψ from $\Gamma \cup \{\varphi\}$ of length 1 consists of ψ by itself; and if it is correct ψ is either $\in \Gamma \cup \{\varphi\}$ or is an axiom. If $\psi \in \Gamma$ or is an axiom, then $\Gamma \vdash \psi$. We also have that $\Gamma \vdash \psi \to (\varphi \to \psi)$ by eq. (12.7), and Proposition 12.19 gives $\Gamma \vdash \varphi \to \psi$. If $\psi \in \{\varphi\}$ then $\Gamma \vdash \varphi \to \psi$ because the last sentence $\varphi \to \psi$ is the same as $\varphi \to \varphi$, and we have derived that in Example 12.8.

For the inductive step, suppose a derivation of ψ from $\Gamma \cup \{\varphi\}$ ends with a step ψ which is justified by modus ponens. (If it is not justified by modus ponens, $\psi \in \Gamma$, $\psi \equiv \varphi$, or ψ is an axiom, and the same reasoning as in the induction basis applies.) Then some previous steps in the derivation are $\chi \to \psi$ and χ, for some formula χ, i.e., $\Gamma \cup \{\varphi\} \vdash \chi \to \psi$ and $\Gamma \cup \{\varphi\} \vdash \chi$, and the respective derivations are shorter, so the inductive hypothesis applies to them. We thus have both:

$$\Gamma \vdash \varphi \to (\chi \to \psi);$$
$$\Gamma \vdash \varphi \to \chi.$$

But also

$$\Gamma \vdash (\varphi \to (\chi \to \psi)) \to ((\varphi \to \chi) \to (\varphi \to \psi)),$$

by eq. (12.8), and two applications of Proposition 12.19 give $\Gamma \vdash \varphi \to \psi$, as required. □

Notice how eq. (12.7) and eq. (12.8) were chosen precisely so that the Deduction Theorem would hold.

The following are some useful facts about derivability, which we leave as exercises.

pl:axd:ded: prop:derivfacts

Proposition 12.21.

pl:axd:ded: derivfacts:a
1. $\vdash (\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$;

pl:axd:ded: derivfacts:b
2. *If* $\Gamma \cup \{\neg\varphi\} \vdash \neg\psi$ *then* $\Gamma \cup \{\psi\} \vdash \varphi$ *(Contraposition);*

pl:axd:ded: derivfacts:c
3. $\{\varphi, \neg\varphi\} \vdash \psi$ *(Ex Falso Quodlibet, Explosion);*

pl:axd:ded: derivfacts:d
4. $\{\neg\neg\varphi\} \vdash \varphi$ *(Double Negation Elimination);*

pl:axd:ded: derivfacts:e
5. *If* $\Gamma \vdash \neg\neg\varphi$ *then* $\Gamma \vdash \varphi$;

Problem 12.3. Prove Proposition 12.21

---

content/propositional-logic/../first-order-logic/axiomatic-deduction/provability-consistency.t

12.6 Derivability and Consistency

pl:axd:prv: sec

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem.

pl:axd:prv: prop:provability-contr

Proposition 12.22. *If* $\Gamma \vdash \varphi$ *and* $\Gamma \cup \{\varphi\}$ *is inconsistent, then* Γ *is inconsistent.*

Proof. If $\Gamma \cup \{\varphi\}$ is inconsistent, then $\Gamma \cup \{\varphi\} \vdash \bot$. By Proposition 12.14, $\Gamma \vdash \psi$ for every $\psi \in \Gamma$. Since also $\Gamma \vdash \varphi$ by hypothesis, $\Gamma \vdash \psi$ for every $\psi \in \Gamma \cup \{\varphi\}$. By Proposition 12.16, $\Gamma \vdash \bot$, i.e., Γ is inconsistent. □

pl:axd:prv: prop:prov-incons

Proposition 12.23. $\Gamma \vdash \varphi$ *iff* $\Gamma \cup \{\neg\varphi\}$ *is inconsistent.*

Proof. First suppose $\Gamma \vdash \varphi$. Then $\Gamma \cup \{\neg\varphi\} \vdash \varphi$ by Proposition 12.15. $\Gamma \cup \{\neg\varphi\} \vdash \neg\varphi$ by Proposition 12.14. We also have $\vdash \neg\varphi \to (\varphi \to \bot)$ by eq. (12.10). So by two applications of Proposition 12.19, we have $\Gamma \cup \{\neg\varphi\} \vdash \bot$.

Now assume $\Gamma \cup \{\neg\varphi\}$ is inconsistent, i.e., $\Gamma \cup \{\neg\varphi\} \vdash \bot$. By the deduction theorem, $\Gamma \vdash \neg\varphi \to \bot$. $\Gamma \vdash (\neg\varphi \to \bot) \to \neg\neg\varphi$ by eq. (12.13), so $\Gamma \vdash \neg\neg\varphi$ by Proposition 12.19. Since $\Gamma \vdash \neg\neg\varphi \to \varphi$ (eq. (12.14)), we have $\Gamma \vdash \varphi$ by Proposition 12.19 again. □

Problem 12.4. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

Proposition 12.24. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

pl:axd:prv: prop:explicit-inc

Proof. $\Gamma \vdash \neg\varphi \to (\varphi \to \bot)$ by eq. (12.10). $\Gamma \vdash \bot$ by two applications of Proposition 12.19. □

Proposition 12.25. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

pl:axd:prv: prop:provability-exhaustive

Proof. Exercise. □

Problem 12.5. Prove Proposition 12.25

---

content/propositional-logic/../first-order-logic/axiomatic-deduction/provability-pr

12.7 Derivability and the Propositional Connectives

explanation

We establish that the derivability relation $\vdash$ of axiomatic deduction is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem.

pl:axd:ppr: sec

Proposition 12.26.

pl:axd:ppr: prop:provability-land

1. *Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$*

pl:axd:ppr: prop:provability-land-left

2. *$\varphi, \psi \vdash \varphi \wedge \psi$.*

pl:axd:ppr: prop:provability-land-right

Proof. 1. From eq. (12.1) and eq. (12.1) by modus ponens.

2. From eq. (12.3) by two applications of modus ponens. □

Proposition 12.27.

pl:axd:ppr: prop:provability-lor

1. *$\varphi \vee \psi, \neg\varphi, \neg\psi$ is inconsistent.*

2. *Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.*

Proof. 1. From eq. (12.9) we get $\vdash \neg\varphi \to (\varphi \to \bot)$ and $\vdash \neg\psi \to (\psi \to \bot)$. So by the deduction theorem, we have $\{\neg\varphi\} \vdash \varphi \to \bot$ and $\{\neg\psi\} \vdash \psi \to \bot$. From eq. (12.6) we get $\{\neg\varphi, \neg\psi\} \vdash (\varphi \vee \psi) \to \bot$. By the deduction theorem, $\{\varphi \vee \psi, \neg\varphi, \neg\psi\} \vdash \bot$.

2. From eq. (12.4) and eq. (12.5) by modus ponsens. □

Proposition 12.28.

pl:axd:ppr: prop:provability-lif

pl:axd:ppr: prop:provability-lif-left
1. $\varphi, \varphi \to \psi \vdash \psi$.

pl:axd:ppr: prop:provability-lif-right
2. *Both* $\neg\varphi \vdash \varphi \to \psi$ *and* $\psi \vdash \varphi \to \psi$.

Proof. 1. We can derive:

1.	φ	Hyp
2.	$\varphi \to \psi$	Hyp
3.	ψ	1, 2, MP

2. By eq. (12.10) and eq. (12.7) and the deduction theorem, respectively. □

---

content/propositional-logic/../first-order-logic/axiomatic-deduction/soundness.tex

12.8 Soundness

pl:axd:sou: sec

A derivation system, such as axiomatic deduction, is *sound* if it cannot derive things that do not actually hold. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that explanation

1. every derivable φ is valid;
2. if φ is derivable from some others Γ, it is also a consequence of them;
3. if a set of formulas Γ is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Proposition 12.29. *If φ is an axiom, then $\mathfrak{v} \vDash \varphi$ for each valuation $\mathfrak{v}$.*

Proof. Do truth tables for each axiom to verify that they are tautologies. □

pl:axd:sou: thm:soundness

Theorem 12.30 (Soundness). *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.*

Proof. By induction on the length of the derivation of φ from Γ. If there are no steps justified by inferences, then all formulas in the derivation are either instances of axioms or are in Γ. By the previous proposition, all the axioms are tautologies, and hence if φ is an axiom then $\Gamma \vDash \varphi$. If $\varphi \in \Gamma$, then trivially $\Gamma \vDash \varphi$.

If the last step of the derivation of φ is justified by modus ponens, then there are formulas ψ and $\psi \to \varphi$ in the derivation, and the induction hypothesis applies to the part of the derivation ending in those formulas (since they contain at least one fewer step justified by an inference). So, by induction hypothesis, $\Gamma \vDash \psi$ and $\Gamma \vDash \psi \to \varphi$. Then $\Gamma \vDash \varphi$ by Theorem 7.21.

Corollary 12.31. *If $\vdash \varphi$, then φ is a tautology.*

pl:axd:sou:cor:weak-soundness

Corollary 12.32. *If Γ is satisfiable, then it is consistent.*

pl:axd:sou:cor:consistency-soundness

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then $\Gamma \vdash \bot$, i.e., there is a derivation of $\bot$ from Γ. By Theorem 12.30, any valuation $\mathfrak{v}$ that satisfies Γ must satisfy $\bot$. Since $\mathfrak{v} \nvDash \bot$ for every valuation $\mathfrak{v}$, no $\mathfrak{v}$ can satisfy Γ, i.e., Γ is not satisfiable. □

Chapter 13

The Completeness Theorem

---

content/propositional-logic/../first-order-logic/completeness/introduction.tex

13.1 Introduction

pl:com:int:sec

The completeness theorem is one of the most fundamental results about logic. It comes in two formulations, the equivalence of which we'll prove. In its first formulation it says something fundamental about the relationship between semantic consequence and our derivation system: if a sentence φ follows from some sentences Γ, then there is also a derivation that establishes $\Gamma \vdash \varphi$. Thus, the derivation system is as strong as it can possibly be without proving things that don't actually follow.

In its second formulation, it can be stated as a model existence result: every consistent set of sentences is satisfiable. Consistency is a proof-theoretic notion: it says that our derivation system is unable to produce certain derivations. But who's to say that just because there are no derivations of a certain sort from Γ, it's guaranteed that there is valuation $\mathfrak{v}$ with $\mathfrak{v} \vDash \Gamma$? Before the completeness theorem was first proved—in fact before we had the derivation systems we now do—the great German mathematician David Hilbert held the view that consistency of mathematical theories guarantees the existence of the objects they are about. He put it as follows in a letter to Gottlob Frege:

> If the arbitrarily given axioms do not contradict one another with all their consequences, then they are true and the things defined by the axioms exist. This is for me the criterion of truth and existence.

Frege vehemently disagreed. The second formulation of the completeness theorem shows that Hilbert was right in at least the sense that if the axioms are consistent, then *some* valuation exists that makes them all true.

These aren't the only reasons the completeness theorem—or rather, its proof—is important. It has a number of important consequences, some of which we'll discuss separately. For instance, since any derivation that shows $\Gamma \vdash \varphi$ is finite and so can only use finitely many of the sentences in Γ, it follows by the completeness theorem that if φ is a consequence of Γ, it is already a consequence of a finite subset of Γ. This is called *compactness*. Equivalently, if every finite subset of Γ is consistent, then Γ itself must be consistent.

Although the compactness theorem follows from the completeness theorem via the detour through derivations, it is also possible to use the *the proof of* the completeness theorem to establish it directly. For what the proof does is take a set of sentences with a certain property—consistency—and constructs a structure out of this set that has certain properties (in this case, that it satisfies the set). Almost the very same construction can be used to directly establish compactness, by starting from "finitely satisfiable" sets of sentences instead of consistent ones.

---

content/propositional-logic/../first-order-logic/completeness/outline.tex

13.2 Outline of the Proof

pl:com:out:sec

The proof of the completeness theorem is a bit complex, and upon first reading it, it is easy to get lost. So let us outline the proof. The first step is a shift of perspective, that allows us to see a route to a proof. When completeness is thought of as "whenever $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$," it may be hard to even come up with an idea: for to show that $\Gamma \vdash \varphi$ we have to find a derivation, and it does not look like the hypothesis that $\Gamma \vDash \varphi$ helps us for this in any way. For some proof systems it is possible to directly construct a derivation, but we will take a slightly different approach. The shift in perspective required is this: completeness can also be formulated as: "if Γ is consistent, it is satisfiable." Perhaps we can use the information in Γ together with the hypothesis that it is consistent to construct a valuation that satisfies every formula in Γ. After all, we know what kind of valuation we are looking for: one that is as Γ describes it!

If Γ contains only propositional variables, it is easy to construct a model for it. All we have to do is come up with a valuation $\mathfrak{v}$ such that $\mathfrak{v} \vDash p$ for all $p \in \Gamma$. Well, let $\mathfrak{v}(p) = \mathbb{T}$ iff $p \in \Gamma$.

Now suppose Γ contains some formula $\neg\psi$, with ψ atomic. We might worry that the construction of $\mathfrak{v}$ interferes with the possibility of making $\neg\psi$ true. But here's where the consistency of Γ comes in: if $\neg\psi \in \Gamma$, then $\psi \notin \Gamma$, or else Γ would be inconsistent. And if $\psi \notin \Gamma$, then according to our construction of $\mathfrak{v}$, $\mathfrak{v} \nvDash \psi$, so $\mathfrak{v} \vDash \neg\psi$. So far so good.

What if Γ contains complex, non-atomic formulas? Say it contains $\varphi \wedge \psi$. To make that true, we should proceed as if both φ and ψ were in Γ. And if $\varphi \vee \psi \in \Gamma$, then we will have to make at least one of them true, i.e., proceed as if one of them was in Γ.

This suggests the following idea: we add additional formulas to Γ so as to (a) keep the resulting set consistent and (b) make sure that for every possible atomic sentence φ, either φ is in the resulting set, or $\neg\varphi$ is, and (c) such that, whenever $\varphi \wedge \psi$ is in the set, so are both φ and ψ, if $\varphi \vee \psi$ is in the set, at least one of φ or ψ is also, etc. We keep doing this (potentially forever). Call the set of all formulas so added Γ^*. Then our construction above would provide us with a valuation $\mathfrak{v}$ for which we could prove, by induction, that it satisfies all sentences in Γ^*, and hence also all sentence in Γ since $\Gamma \subseteq \Gamma^*$. It turns out that guaranteeing (a) and (b) is enough. A set of sentences for which (b) holds is called *complete*. So our task will be to extend the consistent set Γ to a consistent and complete set Γ^*.

So here's what we'll do. First we investigate the properties of complete consistent sets, in particular we prove that a complete consistent set contains $\varphi \wedge \psi$ iff it contains both φ and ψ, $\varphi \vee \psi$ iff it contains at least one of them, etc. (Proposition 13.2). We'll then take the consistent set Γ and show that it can be extended to a consistent and complete set Γ^* (Lemma 13.3). This set Γ^* is what we'll use to define our valuation $\mathfrak{v}(\Gamma^*)$. The valuation is determined by the propositional variables in Γ^* (Definition 13.4). We'll use the properties of complete consistent sets to show that indeed $\mathfrak{v}(\Gamma^*) \vDash \varphi$ iff $\varphi \in \Gamma^*$ (Lemma 13.5), and thus in particular, $\mathfrak{v}(\Gamma^*) \vDash \Gamma$.

---

content/propositional-logic/../first-order-logic/completeness/complete-consistent-s

13.3 Complete Consistent Sets of Sentences

pl:com:ccs:sec
pl:com:ccs:def:complete-set

Definition 13.1 (Complete set). A set Γ of sentences is *complete* iff for any sentence φ, either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$.

explanation

Complete sets of sentences leave no questions unanswered. For any sentence φ, Γ "says" if φ is true or false. The importance of complete sets extends beyond the proof of the completeness theorem. A theory which is complete and axiomatizable, for instance, is always decidable.

explanation

Complete consistent sets are important in the completeness proof since we can guarantee that every consistent set of sentences Γ is contained in a complete consistent set Γ^*. A complete consistent set contains, for each sentence φ, either φ or its negation $\neg\varphi$, but not both. This is true in particular for propositional variables, so from a complete consistent set, we can construct a valuation where the truth value assigned to propositional variables is defined according to which propositional variables are in Γ^*. This valuation can then be shown to make all sentences in Γ^* (and hence also all those in Γ) true. The proof of

this latter fact requires that $\neg\varphi \in \Gamma^*$ iff $\varphi \notin \Gamma^*$, $(\varphi \vee \psi) \in \Gamma^*$ iff $\varphi \in \Gamma^*$ or $\psi \in \Gamma^*$, etc.

In what follows, we will often tacitly use the properties of reflexivity, monotonicity, and transitivity of $\vdash$ (see sections 9.6, 10.5, 11.5 and 12.4).

pl:com:ccs: prop:ccs

Proposition 13.2. *Suppose Γ is complete and consistent. Then:*

pl:com:ccs: prop:ccs-prov-in
1. *If $\Gamma \vdash \varphi$, then $\varphi \in \Gamma$.*

pl:com:ccs: prop:ccs-and
2. *$\varphi \wedge \psi \in \Gamma$ iff both $\varphi \in \Gamma$ and $\psi \in \Gamma$.*

pl:com:ccs: prop:ccs-or
3. *$\varphi \vee \psi \in \Gamma$ iff either $\varphi \in \Gamma$ or $\psi \in \Gamma$.*

pl:com:ccs: prop:ccs-if
4. *$\varphi \to \psi \in \Gamma$ iff either $\varphi \notin \Gamma$ or $\psi \in \Gamma$.*

Proof. Let us suppose for all of the following that Γ is complete and consistent.

1. If $\Gamma \vdash \varphi$, then $\varphi \in \Gamma$.

   Suppose that $\Gamma \vdash \varphi$. Suppose to the contrary that $\varphi \notin \Gamma$. Since Γ is complete, $\neg\varphi \in \Gamma$. By Propositions 9.19, 10.17, 11.17 and 12.24, Γ is inconsistent. This contradicts the assumption that Γ is consistent. Hence, it cannot be the case that $\varphi \notin \Gamma$, so $\varphi \in \Gamma$.

2. $\varphi \wedge \psi \in \Gamma$ iff both $\varphi \in \Gamma$ and $\psi \in \Gamma$:

   For the forward direction, suppose $\varphi \wedge \psi \in \Gamma$. Then by Propositions 9.21, 10.19, 11.19 and 12.26, item (1), $\Gamma \vdash \varphi$ and $\Gamma \vdash \psi$. By (1), $\varphi \in \Gamma$ and $\psi \in \Gamma$, as required.

   For the reverse direction, let $\varphi \in \Gamma$ and $\psi \in \Gamma$. By Propositions 9.21, 10.19, 11.19 and 12.26, item (2), $\Gamma \vdash \varphi \wedge \psi$. By (1), $\varphi \wedge \psi \in \Gamma$.

3. First we show that if $\varphi \vee \psi \in \Gamma$, then either $\varphi \in \Gamma$ or $\psi \in \Gamma$. Suppose $\varphi \vee \psi \in \Gamma$ but $\varphi \notin \Gamma$ and $\psi \notin \Gamma$. Since Γ is complete, $\neg\varphi \in \Gamma$ and $\neg\psi \in \Gamma$. By Propositions 9.22, 10.20, 11.20 and 12.27, item (1), Γ is inconsistent, a contradiction. Hence, either $\varphi \in \Gamma$ or $\psi \in \Gamma$.

   For the reverse direction, suppose that $\varphi \in \Gamma$ or $\psi \in \Gamma$. By Propositions 9.22, 10.20, 11.20 and 12.27, item (2), $\Gamma \vdash \varphi \vee \psi$. By (1), $\varphi \vee \psi \in \Gamma$, as required.

4. For the forward direction, suppose $\varphi \to \psi \in \Gamma$, and suppose to the contrary that $\varphi \in \Gamma$ and $\psi \notin \Gamma$. On these assumptions, $\varphi \to \psi \in \Gamma$ and $\varphi \in \Gamma$. By Propositions 9.23, 10.21, 11.21 and 12.28, item (1), $\Gamma \vdash \psi$. But then by (1), $\psi \in \Gamma$, contradicting the assumption that $\psi \notin \Gamma$.

   For the reverse direction, first consider the case where $\varphi \notin \Gamma$. Since Γ is complete, $\neg\varphi \in \Gamma$. By Propositions 9.23, 10.21, 11.21 and 12.28, item (2), $\Gamma \vdash \varphi \to \psi$. Again by (1), we get that $\varphi \to \psi \in \Gamma$, as required.

   Now consider the case where $\psi \in \Gamma$. By Propositions 9.23, 10.21, 11.21 and 12.28, item (2) again, $\Gamma \vdash \varphi \to \psi$. By (1), $\varphi \to \psi \in \Gamma$. □

Problem 13.1. Complete the proof of Proposition 13.2.

---

content/propositional-logic/../first-order-logic/completeness/lindenbaums-lemma.tex

13.4 Lindenbaum's Lemma

explanation

We now prove a lemma that shows that any consistent set of sentences is contained in some set of sentences which is not just consistent, but also complete. The proof works by adding one sentence at a time, guaranteeing at each step that the set remains consistent. We do this so that for every φ, either φ or $\neg\varphi$ gets added at some stage. The union of all stages in that construction then contains either φ or its negation $\neg\varphi$ and is thus complete. It is also consistent, since we make sure at each stage not to introduce an inconsistency.

pl:com:lin:sec

Lemma 13.3 (Lindenbaum's Lemma). *Every consistent set Γ in a language $\mathcal{L}$ can be extended to a complete and consistent set Γ^*.*

pl:com:lin:lem:lindenbaum

Proof. Let Γ be consistent. Let φ_0, φ_1, ... be an enumeration of all the sentences of $\mathcal{L}$. Define $\Gamma_0 = \Gamma$, and

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\varphi_n\} & \text{if } \Gamma_n \cup \{\varphi_n\} \text{ is consistent;} \\ \Gamma_n \cup \{\neg\varphi_n\} & \text{otherwise.} \end{cases}$$

Let $\Gamma^* = \bigcup_{n \geq 0} \Gamma_n$.

Each Γ_n is consistent: Γ_0 is consistent by definition. If $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$, this is because the latter is consistent. If it isn't, $\Gamma_{n+1} = \Gamma_n \cup \{\neg\varphi_n\}$. We have to verify that $\Gamma_n \cup \{\neg\varphi_n\}$ is consistent. Suppose it's not. Then *both* $\Gamma_n \cup \{\varphi_n\}$ and $\Gamma_n \cup \{\neg\varphi_n\}$ are inconsistent. This means that Γ_n would be inconsistent by Propositions 9.20, 10.18, 11.18 and 12.25, contrary to the induction hypothesis.

For every n and every $i < n$, $\Gamma_i \subseteq \Gamma_n$. This follows by a simple induction on n. For $n = 0$, there are no $i < 0$, so the claim holds automatically. For the inductive step, suppose it is true for n. We show that if $i < n + 1$ then $\Gamma_i \subseteq \Gamma_{n+1}$. We have $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$ or $= \Gamma_n \cup \{\neg\varphi_n\}$ by construction. So $\Gamma_n \subseteq \Gamma_{n+1}$. If $i < n + 1$, then $\Gamma_i \subseteq \Gamma_n$ by inductive hypothesis (if $i < n$) or the trivial fact that $\Gamma_n \subseteq \Gamma_n$ (if $i = n$). We get that $\Gamma_i \subseteq \Gamma_{n+1}$ by transitivity of $\subseteq$.

From this it follows that Γ^* is consistent. Here's why: Let $\Gamma' \subseteq \Gamma^*$ be finite. Each $\psi \in \Gamma'$ is also in Γ_i for some i. Let n be the largest of these. Since $\Gamma_i \subseteq \Gamma_n$ if $i \leq n$, every $\psi \in \Gamma'$ is also $\in \Gamma_n$, i.e., $\Gamma' \subseteq \Gamma_n$, and Γ_n is consistent. So, every finite subset $\Gamma' \subseteq \Gamma^*$ is consistent. By Propositions 9.16, 10.14, 11.14 and 12.18, Γ^* is consistent.

Every sentence of $\mathrm{Frm}(\mathcal{L})$ appears on the list used to define Γ^*. If $\varphi_n \notin \Gamma^*$, then that is because $\Gamma_n \cup \{\varphi_n\}$ was inconsistent. But then $\neg\varphi_n \in \Gamma^*$, so Γ^* is complete. □

---

content/propositional-logic/../first-order-logic/completeness/construction-of-model

13.5 Construction of a Model

pl:com:mod: sec

We are now ready to define a valuation that makes all $\varphi \in \Gamma$ true. To do this, we first apply Lindenbaum's Lemma: we get a complete consistent $\Gamma^* \supseteq \Gamma$. We let the propositional variables in Γ^* determine $\mathfrak{v}(\Gamma^*)$. explanation

pl:com:mod: defn:termmodel

Definition 13.4. Suppose Γ^* is a complete consistent set of formulas. Then we let

$$\mathfrak{v}(\Gamma^*)(p) = \begin{cases} \mathbb{T} & \text{if } p \in \Gamma^* \\ \mathbb{F} & \text{if } p \notin \Gamma^* \end{cases}$$

pl:com:mod: lem:truth

Lemma 13.5 (Truth Lemma). *$\mathfrak{v}(\Gamma^*) \vDash \varphi$ iff $\varphi \in \Gamma^*$.*

Proof. We prove both directions simultaneously, and by induction on φ.

1. $\varphi \equiv \bot$: $\mathfrak{v}(\Gamma^*) \nvDash \bot$ by definition of satisfaction. On the other hand, $\bot \notin \Gamma^*$ since Γ^* is consistent.
2. $\varphi \equiv \top$: $\mathfrak{v}(\Gamma^*) \vDash \top$ by definition of satisfaction. On the other hand, $\top \in \Gamma^*$ since Γ^* is consistent and complete, and $\Gamma^* \vdash \top$.
3. $\varphi \equiv p$: $\mathfrak{v}(\Gamma^*) \vDash p$ iff $\mathfrak{v}(\Gamma^*)(p) = \mathbb{T}$ (by the definition of satisfaction) iff $p \in \Gamma^*$ (by the construction of $\mathfrak{v}(\Gamma^*)$).
4. $\varphi \equiv \neg\psi$: $\mathfrak{v}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{v}(\Gamma^*) \nvDash \psi$ (by definition of satisfaction). By induction hypothesis, $\mathfrak{v}(\Gamma^*) \nvDash \psi$ iff $\psi \notin \Gamma^*$. Since Γ^* is consistent and complete, $\psi \notin \Gamma^*$ iff $\neg\psi \in \Gamma^*$.
5. $\varphi \equiv \psi \wedge \chi$: $\mathfrak{v}(\Gamma^*) \vDash \varphi$ iff we have both $\mathfrak{v}(\Gamma^*) \vDash \psi$ and $\mathfrak{v}(\Gamma^*) \vDash \chi$ (by definition of satisfaction) iff both $\psi \in \Gamma^*$ and $\chi \in \Gamma^*$ (by the induction hypothesis). By Proposition 13.2(2), this is the case iff $(\psi \wedge \chi) \in \Gamma^*$.
6. $\varphi \equiv \psi \vee \chi$: $\mathfrak{v}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{v}(\Gamma^*) \vDash \psi$ or $\mathfrak{v}(\Gamma^*) \vDash \chi$ (by definition of satisfaction) iff $\psi \in \Gamma^*$ or $\chi \in \Gamma^*$ (by induction hypothesis). This is the case iff $(\psi \vee \chi) \in \Gamma^*$ (by Proposition 13.2(3)).
7. $\varphi \equiv \psi \to \chi$: $\mathfrak{v}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{v}(\Gamma^*) \nvDash \psi$ or $\mathfrak{v}(\Gamma^*) \vDash \chi$ (by definition of satisfaction) iff $\psi \notin \Gamma^*$ or $\chi \in \Gamma^*$ (by induction hypothesis). This is the case iff $(\psi \to \chi) \in \Gamma^*$ (by Proposition 13.2(4)).

---

content/propositional-logic/../first-order-logic/completeness/completeness-thm.tex

13.6 The Completeness Theorem

pl:com:cth: sec

Let's combine our results: we arrive at the completeness theorem. explanation

pl:com:cth: thm:completeness

Theorem 13.6 (Completeness Theorem). *Let Γ be a set of sentences. If Γ is consistent, it is satisfiable.*

Proof. Suppose Γ is consistent. By Lemma 13.3, there is a $\Gamma^* \supseteq \Gamma$ which is consistent and complete. By Lemma 13.5, $\mathfrak{v}(\Gamma^*) \vDash \varphi$ iff $\varphi \in \Gamma^*$. From this it follows in particular that for all $\varphi \in \Gamma$, $\mathfrak{v}(\Gamma^*) \vDash \varphi$, so Γ is satisfiable. □

Corollary 13.7 (Completeness Theorem, Second Version). *For all Γ and sentences φ: if $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$.* pl:com:cth:cor:completeness

Proof. Note that the Γ's in Corollary 13.7 and Theorem 13.6 are universally quantified. To make sure we do not confuse ourselves, let us restate Theorem 13.6 using a different variable: for any set of sentences Δ, if Δ is consistent, it is satisfiable. By contraposition, if Δ is not satisfiable, then Δ is inconsistent. We will use this to prove the corollary.

Suppose that $\Gamma \vDash \varphi$. Then $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable by Proposition 7.20. Taking $\Gamma \cup \{\neg\varphi\}$ as our Δ, the previous version of Theorem 13.6 gives us that $\Gamma \cup \{\neg\varphi\}$ is inconsistent. By Propositions 9.18, 10.16, 11.16 and 12.23, $\Gamma \vdash \varphi$.□

Problem 13.2. Use Corollary 13.7 to prove Theorem 13.6, thus showing that the two formulations of the completeness theorem are equivalent.

Problem 13.3. In order for a derivation system to be complete, its rules must be strong enough to prove every unsatisfiable set inconsistent. Which of the rules of derivation were necessary to prove completeness? Are any of these rules not used anywhere in the proof? In order to answer these questions, make a list or diagram that shows which of the rules of derivation were used in which results that lead up to the proof of Theorem 13.6. Be sure to note any tacit uses of rules in these proofs.

---

content/propositional-logic/../first-order-logic/completeness/compactness.tex

13.7 The Compactness Theorem

pl:com:com:sec

One important consequence of the completeness theorem is the compactness theorem. The compactness theorem states that if each *finite* subset of a set of sentences is satisfiable, the entire set is satisfiable—even if the set itself is infinite. This is far from obvious. There is nothing that seems to rule out, at first glance at least, the possibility of there being infinite sets of sentences which are contradictory, but the contradiction only arises, so to speak, from the infinite number. The compactness theorem says that such a scenario can be ruled out: there are no unsatisfiable infinite sets of sentences each finite subset of which is satisfiable. Like the completeness theorem, it has a version related to entailment: if an infinite set of sentences entails something, already a finite subset does.

Definition 13.8. A set Γ of formulas is *finitely satisfiable* iff every finite $\Gamma_0 \subseteq \Gamma$ is satisfiable.

pl:com:com: thm:compactness **Theorem 13.9 (Compactness Theorem).** *The following hold for any sentences Γ and φ:*

1. *$\Gamma \vDash \varphi$ iff there is a finite $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vDash \varphi$.*
2. *Γ is satisfiable iff it is finitely satisfiable.*

Proof. We prove (2). If Γ is satisfiable, then there is a valuation $\mathfrak{v}$ such that $\mathfrak{v} \vDash \varphi$ for all $\varphi \in \Gamma$. Of course, this $\mathfrak{v}$ also satisfies every finite subset of Γ, so Γ is finitely satisfiable.

Now suppose that Γ is finitely satisfiable. Then every finite subset $\Gamma_0 \subseteq \Gamma$ is satisfiable. By soundness (Corollaries 9.28, 10.24, 11.26 and 12.32), every finite subset is consistent. Then Γ itself must be consistent by Propositions 9.16, 10.14, 11.14 and 12.18. By completeness (Theorem 13.6), since Γ is consistent, it is satisfiable. □

Problem 13.4. Prove (1) of Theorem 13.9.

---

content/propositional-logic/../first-order-logic/completeness/compactness-direct.tex

13.8 A Direct Proof of the Compactness Theorem

pl:com:cpd: sec We can prove the Compactness Theorem directly, without appealing to the Completeness Theorem, using the same ideas as in the proof of the completeness theorem. In the proof of the Completeness Theorem we started with a consistent set Γ of sentences, expanded it to a consistent and complete set Γ^* of sentences, and then showed that in the valuation $\mathfrak{v}(\Gamma^*)$ constructed from Γ^*, all sentences of Γ are true, so Γ is satisfiable.

We can use the same method to show that a finitely satisfiable set of sentences is satisfiable. We just have to prove the corresponding versions of the results leading to the truth lemma where we replace "consistent" with "finitely satisfiable."

pl:com:cpd: prop:fsat-ccs **Proposition 13.10.** *Suppose Γ is complete and finitely satisfiable. Then:*

1. *$(\varphi \wedge \psi) \in \Gamma$ iff both $\varphi \in \Gamma$ and $\psi \in \Gamma$.*
2. *$(\varphi \vee \psi) \in \Gamma$ iff either $\varphi \in \Gamma$ or $\psi \in \Gamma$.*
3. *$(\varphi \to \psi) \in \Gamma$ iff either $\varphi \notin \Gamma$ or $\psi \in \Gamma$.*

Problem 13.5. Prove Proposition 13.10. Avoid the use of $\vdash$.

pl:com:cpd: lem:fsat-lindenbaum **Lemma 13.11.** *Every finitely satisfiable set Γ can be extended to a complete and finitely satisfiable set Γ^*.*

Problem 13.6. Prove Lemma 13.11. (Hint: the crucial step is to show that if Γ_n is finitely satisfiable, then either $\Gamma_n \cup \{\varphi_n\}$ or $\Gamma_n \cup \{\neg\varphi_n\}$ is finitely satisfiable.)

Theorem 13.12 (Compactness). *Γ is satisfiable if and only if it is finitely satisfiable.*

pl:com:cpd:thm:compactness-direct

Proof. If Γ is satisfiable, then there is a valuation $\mathfrak{v}$ such that $\mathfrak{v} \vDash \varphi$ for all $\varphi \in \Gamma$. Of course, this $\mathfrak{v}$ also satisfies every finite subset of Γ, so Γ is finitely satisfiable.

Now suppose that Γ is finitely satisfiable. By Lemma 13.11, Γ can be extended to a complete and finitely satisfiable set Γ^*. Construct the valuation $\mathfrak{v}(\Gamma^*)$ as in Definition 13.4. The proof of the Truth Lemma (Lemma 13.5) goes through if we replace references to Proposition 13.2. □

Problem 13.7. Write out the complete proof of the Truth Lemma (Lemma 13.5) in the version required for the proof of Theorem 13.12.

Part III

First-order Logic

This part covers the metatheory of first-order logic through completeness. Currently it does not rely on a separate treatment of propositional logic; everything is proved. The source files will exclude the material on quantifiers (and replace "structure" with "valuation", $\mathfrak{M}$ with $\mathfrak{v}$, etc.) if the "FOL" tag is false. In fact, most of the material in the part on propositional logic is simply the first-order material with the "FOL" tag turned off.

If the part on propositional logic is included, this results in a lot of repetition. It is planned, however, to make it possible to let this part take into account the material on propositional logic (and exclude the material already covered, as well as shorten proofs with references to the respective places in the propositional part).

Chapter 14

Introduction to First-Order Logic

content/first-order-logic/introduction/first-order-logic.tex

14.1 First-Order Logic

fol:int:fol:sec

You are probably familiar with first-order logic from your first introduction to formal logic.[1] You may know it as "quantificational logic" or "predicate logic." First-order logic, first of all, is a formal language. That means, it has a certain vocabulary, and its expressions are strings from this vocabulary. But not every string is permitted. There are different kinds of permitted expressions: terms,

[1] In fact, we more or less assume you are! If you're not, you could review a more elementary textbook, such as *forall x* (Magnus et al., 2021).

formulas, and sentences. We are mainly interested in sentences of first-order logic: they provide us with a formal analogue of sentences of English, and about them we can ask the questions a logician typically is interested in. For instance:

- Does ψ follow from φ logically?
- Is φ logically true, logically false, or contingent?
- Are φ and ψ equivalent?

These questions are primarily questions about the "meaning" of sentences of first-order logic. For instance, a philosopher would analyze the question of whether ψ follows logically from φ as asking: is there a case where φ is true but ψ is false (ψ doesn't follow from φ), or does every case that makes φ true also make ψ true (ψ does follow from φ)? But we haven't been told yet what a "case" is—that is the job of *semantics*. The semantics of first-order logic provides a mathematically precise model of the philosopher's intuitive idea of "case," and also—and this is important—of what it is for a sentence φ to be *true in* a case. We call the mathematically precise model that we will develop a structure. The relation which makes "true in" precise, is called the relation of *satisfaction*. So what we will define is "φ is satisfied in $\mathfrak{M}$" (in symbols: $\mathfrak{M} \vDash \varphi$) for sentences φ and structures $\mathfrak{M}$. Once this is done, we can also give precise definitions of the other semantical terms such as "follows from" or "is logically true." These definitions will make it possible to settle, again with mathematical precision, whether, e.g., $\forall x\,(\varphi(x) \to \psi(x)), \exists x\,\varphi(x) \vDash \exists x\,\psi(x)$. The answer will, of course, be "yes." If you've already been trained to symbolize sentences of English in first-order logic, you will recognize this as, e.g., the symbolizations of, say, "All ants are insects, there are ants, therefore there are insects." That is obviously a valid argument, and so our mathematical model of "follows from" for our formal language should give the same answer.

Another topic you probably remember from your first introduction to formal logic is that there are *derivations*. If you have taken a first formal logic course, your instructor will have made you practice finding such derivations, perhaps even a derivation that shows that the above entailment holds. There are many different ways to give derivations: you may have done something called "natural deduction" or "truth trees," but there are many others. The purpose of derivation systems is to provide tools using which the logicians' questions above can be answered: e.g., a natural deduction derivation in which $\forall x\,(\varphi(x) \to \psi(x))$ and $\exists x\,\varphi(x)$ are premises and $\exists x\,\psi(x)$ is the conclusion (last line) *verifies* that $\exists x\,\psi(x)$ logically follows from $\forall x\,(\varphi(x) \to \psi(x))$ and $\exists x\,\varphi(x)$.

But why is that? On the face of it, derivation systems have nothing to do with semantics: giving a formal derivation merely involves arranging symbols in certain rule-governed ways; they don't mention "cases" or "true in" at all. The connection between derivation systems and semantics has to be established by a meta-logical investigation. What's needed is a mathematical proof, e.g., that a formal derivation of $\exists x\,\psi(x)$ from premises $\forall x\,(\varphi(x) \to \psi(x))$ and $\exists x\,\varphi(x)$ is

possible, if, and only if, $\forall x\,(\varphi(x) \to \psi(x))$ and $\exists x\,\varphi(x)$ together entail $\exists x\,\psi(x)$. Before this can be done, however, a lot of painstaking work has to be carried out to get the definitions of syntax and semantics correct.

---

content/first-order-logic/introduction/syntax.tex

14.2 Syntax

fol:int:syn:sec

We first must make precise what strings of symbols count as sentences of first-order logic. We'll do this later; for now we'll just proceed by example. The basic building blocks—the vocabulary—of first-order logic divides into two parts. The first part is the symbols we use to say specific things or to pick out specific things. We pick out things using constant symbols, and we say stuff about the things we pick out using predicate symbols. E.g, we might use a as a constant symbol to pick out a single thing, and then say something about it using the sentence $P(a)$. If you have meanings for "a" and "P" in mind, you can read $P(a)$ as a sentence of English (and you probably have done so when you first learned formal logic). Once you have such simple sentences of first-order logic, you can build more complex ones using the second part of the vocabulary: the logical symbols (connectives and quantifiers). So, for instance, we can form expressions like $(P(a) \land Q(b))$ or $\exists x\,P(x)$.

In order to provide the precise definitions of semantics and the rules of our derivation systems required for rigorous meta-logical study, we first of all have to give a precise definition of what counts as a sentence of first-order logic. The basic idea is easy enough to understand: there are some simple sentences we can form from just predicate symbols and constant symbols, such as $P(a)$. And then from these we form more complex ones using the connectives and quantifiers. But what exactly are the rules by which we are allowed to form more complex sentences? These must be specified, otherwise we have not defined "sentence of first-order logic" precisely enough. There are a few issues. The first one is to get the right strings to count as sentences. The second one is to do this in such a way that we can give mathematical proofs about *all* sentences. Finally, we'll have to also give precise definitions of some rudimentary operations with sentences, such as "replace every x in φ by b." The trouble is that the quantifiers and variables we have in first-order logic make it not entirely obvious how this should be done. E.g., should $\exists x\,P(a)$ count as a sentence? What about $\exists x\,\exists x\,P(x)$? What should the result of "replace x by b in $(P(x) \land \exists x\,P(x))$" be?

---

content/first-order-logic/introduction/formulas.tex

14.3 Formulas

Here is the approach we will use to rigorously specify sentences of first-order logic and to deal with the issues arising from the use of variables. We first

define a *different* set of expressions: formulas. Once we've done that, we can consider the role variables play in them—and on the basis of some other ideas, namely those of "free" and "bound" variables, we can define what a sentence is (namely, a formula without free variables). We do this not just because it makes the definition of "sentence" more manageable, but also because it will be crucial to the way we define the semantic notion of satisfaction.

Let's define "formula" for a simple first-order language, one containing only a single predicate symbol P and a single constant symbol a, and only the logical symbols $\neg$, $\wedge$, and $\exists$. Our full definitions will be much more general: we'll allow infinitely many predicate symbols and constant symbols. In fact, we will also consider function symbols which can be combined with constant symbols and variables to form "terms." For now, a and the variables will be our only terms. We do need infinitely many variables. We'll officially use the symbols v_0, v_1, ..., as variables.

Definition 14.1. The set of *formulas* Frm is defined as follows:

1. $P(a)$ and $P(v_i)$ are formulas ($i \in \mathbb{N}$). fol:int:fml:fmls-atom
2. If φ is a formula, then $\neg\varphi$ is formula. fol:int:fml:fmls-not
3. If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.
4. If φ is a formula and x is a variable, then $\exists x\, \varphi$ is a formula. fol:int:fml:fmls-ex
5. Nothing else is a formula. fol:int:fml:fmls-limit

(1) tells us that $P(a)$ and $P(v_i)$ are formulas, for any $i \in \mathbb{N}$. These are the so-called *atomic* formulas. They give us something to start from. The other clauses give us ways of forming new formulas from ones we have already formed. So for instance, by (2), we get that $\neg P(v_2)$ is a formula, since $P(v_2)$ is already a formula by (1). Then, by (4), we get that $\exists v_2\, \neg P(v_2)$ is another formula, and so on. (5) tells us that *only* strings we can form in this way count as formulas. In particular, $\exists v_0\, P(a)$ and $\exists v_0\, \exists v_0\, P(a)$ *do* count as formulas, and $(\neg P(a))$ does not, because of the extraneous outer parentheses.

This way of defining formulas is called an *inductive definition*, and it allows us to prove things about formulas using a version of proof by induction called *structural induction*. These are discussed in a general way in section 74.4 and section 74.5, which you should review before delving into the proofs later on. Basically, the idea is that if you want to give a proof that something is true for all formulas, you show first that it is true for the atomic formulas, and then that *if* it's true for any formula φ (and ψ), it's *also* true for $\neg\varphi$, $(\varphi \wedge \psi)$, and $\exists x\, \varphi$. For instance, this proves that it's true for $\exists v_2\, \neg P(v_2)$: from the first part you know that it's true for the atomic formula $P(v_2)$. Then you get that it's true for $\neg P(v_2)$ by the second part, and then again that it's true for $\exists v_2\, \neg P(v_2)$ itself. Since all formulas are inductively generated from atomic formulas, this works for any of them.

---

content/first-order-logic/introduction/satisfaction.tex

14.4 Satisfaction

fol:int:sat:sec

We can already skip ahead to the semantics of first-order logic once we know what formulas are: here, the basic definition is that of a structure. For our simple language, a structure $\mathfrak{M}$ has just three components: a non-empty set $|\mathfrak{M}|$ called the *domain*, what a picks out in $\mathfrak{M}$, and what P is true of in $\mathfrak{M}$. The object picked out by a is denoted $a^{\mathfrak{M}}$ and the set of things P is true of by $P^{\mathfrak{M}}$. A structure $\mathfrak{M}$ consists of just these three things: $|\mathfrak{M}|$, $a^{\mathfrak{M}} \in |\mathfrak{M}|$ and $P^{\mathfrak{M}} \subseteq |\mathfrak{M}|$. The general case will be more complicated, since there will be many predicate symbols and constant symbols, the constant symbols can have more than one place, and there will also be function symbols.

This is enough to give a definition of satisfaction for formulas that don't contain variables. The idea is to give an inductive definition that mirrors the way we have defined formulas. We specify when an atomic formula is satisfied in $\mathfrak{M}$, and then when, e.g., $\neg\varphi$ is satisfied in $\mathfrak{M}$ on the basis of whether or not φ is satisfied in $\mathfrak{M}$. E.g., we could define:

1. $P(a)$ is satisfied in $\mathfrak{M}$ iff $a^{\mathfrak{M}} \in P^{\mathfrak{M}}$.
2. $\neg\varphi$ is satisfied in $\mathfrak{M}$ iff φ is not satisfied in $\mathfrak{M}$.
3. $(\varphi \wedge \psi)$ is satisfied in $\mathfrak{M}$ iff φ is satisfied in $\mathfrak{M}$, and ψ is satisfied in $\mathfrak{M}$ as well.

Let's say that $|\mathfrak{M}| = \{0, 1, 2\}$, $a^{\mathfrak{M}} = 1$, and $P^{\mathfrak{M}} = \{1, 2\}$. This definition would tell us that $P(a)$ is satisfied in $\mathfrak{M}$ (since $a^{\mathfrak{M}} = 1 \in \{1, 2\} = P^{\mathfrak{M}}$). It tells us further that $\neg P(a)$ is not satisfied in $\mathfrak{M}$, and that in turn $\neg\neg P(a)$ is and $(\neg P(a) \wedge P(a))$ is not satisfied, and so on.

The trouble comes when we want to give a definition for the quantifiers: we'd like to say something like, "$\exists v_0\, P(v_0)$ is satisfied iff $P(v_0)$ is satisfied." But the structure $\mathfrak{M}$ doesn't tell us what to do about variables. What we actually want to say is that $P(v_0)$ is satisfied *for some value of* v_0. To make this precise we need a way to assign elements of $|\mathfrak{M}|$ not just to a but also to v_0. To this end, we introduce variable *assignments*. A variable assignment is simply a function s that maps variables to elements of $|\mathfrak{M}|$ (in our example, to one of 1, 2, or 3). Since we don't know beforehand which variables might appear in a formula we can't limit which variables s assigns values to. The simple solution is to require that s assigns values to *all* variables v_0, v_1, ... We'll just use only the ones we need.

Instead of defining satisfaction of formulas just relative to a structure, we'll define it relative to a structure $\mathfrak{M}$ *and* a variable assignment s, and write $\mathfrak{M}, s \vDash \varphi$ for short. Our definition will now include an additional clause to deal with atomic formulas containing variables:

1. $\mathfrak{M}, s \vDash P(a)$ iff $a^{\mathfrak{M}} \in P^{\mathfrak{M}}$.
2. $\mathfrak{M}, s \vDash P(v_i)$ iff $s(v_i) \in P^{\mathfrak{M}}$.
3. $\mathfrak{M}, s \vDash \neg\varphi$ iff not $\mathfrak{M}, s \vDash \varphi$.

4. $\mathfrak{M}, s \vDash (\varphi \wedge \psi)$ iff $\mathfrak{M}, s \vDash \varphi$ and $\mathfrak{M}, s \vDash \psi$.

Ok, this solves one problem: we can now say when $\mathfrak{M}$ satisfies $P(v_0)$ for the value $s(v_0)$. To get the definition right for $\exists v_0\, P(v_0)$ we have to do one more thing: We want to have that $\mathfrak{M}, s \vDash \exists v_0\, P(v_0)$ iff $\mathfrak{M}, s' \vDash P(v_0)$ for *some* way s' of assigning a value to v_0. But the value assigned to v_0 does not necessarily have to be the value that $s(v_0)$ picks out. We'll introduce a notation for that: if $m \in |\mathfrak{M}|$, then we let $s[m/v_0]$ be the assignment that is just like s (for all variables other than v_0), except to v_0 it assigns m. Now our definition can be:

5. $\mathfrak{M}, s \vDash \exists v_i\, \varphi$ iff $\mathfrak{M}, s[m/v_i] \vDash \varphi$ for some $m \in |\mathfrak{M}|$.

Does it work out? Let's say we let $s(v_i) = 0$ for all $i \in \mathbb{N}$. $\mathfrak{M}, s \vDash \exists v_0\, P(v_0)$ iff there is an $m \in |\mathfrak{M}|$ so that $\mathfrak{M}, s[m/v_0] \vDash P(v_0)$. And there is: we can choose $m = 1$ or $m = 2$. Note that this is true even if the value $s(v_0)$ assigned to v_0 by s itself—in this case, 0—doesn't do the job. We have $\mathfrak{M}, s[1/v_0] \vDash P(v_0)$ but not $\mathfrak{M}, s \vDash P(v_0)$.

If this looks confusing and cumbersome: it is. But the added complexity is required to give a precise, inductive definition of satisfaction for all formulas, and we need something like it to precisely define the semantic notions. There are other ways of doing it, but they are all equally (in)elegant.

---

content/first-order-logic/introduction/sentences.tex

14.5 Sentences

fol:int:snt:sec

Ok, now we have a (sketch of a) definition of satisfaction ("true in") for structures and formulas. But it needs this additional bit—a variable assignment—and what we wanted is a definition of sentences. How do we get rid of assignments, and what are sentences?

You probably remember a discussion in your first introduction to formal logic about the relation between variables and quantifiers. A quantifier is always followed by a variable, and then in the part of the sentence to which that quantifier applies (its "scope"), we understand that the variable is "bound" by that quantifier. In formulas it was not required that every variable has a matching quantifier, and variables without matching quantifiers are "free" or "unbound." We will take sentences to be all those formulas that have no free variables.

Again, the intuitive idea of when an occurrence of a variable in a formula φ is bound, which quantifier binds it, and when it is free, is not difficult to get. You may have learned a method for testing this, perhaps involving counting parentheses. We have to insist on a precise definition—and because we have defined formulas by induction, we can give a definition of the free and bound occurrences of a variable x in a formula φ also by induction. E.g., it might look like this for our simplified language:

1. If φ is atomic, all occurrences of x in it are free (that is, the occurrence of x in $P(x)$ is free).

2. If φ is of the form $\neg\psi$, then an occurrence of x in $\neg\psi$ is free iff the corresponding occurrence of x is free in ψ (that is, the free occurrences of variables in ψ are exactly the corresponding occurrences in $\neg\psi$).

3. If φ is of the form $(\psi \wedge \chi)$, then an occurrence of x in $(\psi \wedge \chi)$ is free iff the corresponding occurrence of x is free in ψ or in χ.

4. If φ is of the form $\exists x\, \psi$, then no occurrence of x in φ is free; if it is of the form $\exists y\, \psi$ where y is a different variable than x, then an occurrence of x in $\exists y\, \psi$ is free iff the corresponding occurrence of x is free in ψ.

Once we have a precise definition of free and bound occurrences of variables, we can simply say: a sentence is any formula without free occurrences of variables.

---

content/first-order-logic/introduction/semantic-notions.tex

14.6 Semantic Notions

fol:int:sem:sec

We mentioned above that when we consider whether $\mathfrak{M}, s \vDash \varphi$ holds, we (for convenience) let s assign values to all variables, but only the values it assigns to variables in φ are used. In fact, it's only the values of *free* variables in φ that matter. Of course, because we're careful, we are going to prove this fact. Since sentences have no free variables, s doesn't matter at all when it comes to whether or not they are satisfied in a structure. So, when φ is a sentence we can define $\mathfrak{M} \vDash \varphi$ to mean "$\mathfrak{M}, s \vDash \varphi$ for all s," which as it happens is true iff $\mathfrak{M}, s \vDash \varphi$ for at least one s. We need to introduce variable assignments to get a working definition of satisfaction for formulas, but for sentences, satisfaction is independent of the variable assignments.

Once we have a definition of "$\mathfrak{M} \vDash \varphi$," we know what "case" and "true in" mean as far as sentences of first-order logic are concerned. On the basis of the definition of $\mathfrak{M} \vDash \varphi$ for sentences we can then define the basic semantic notions of validity, entailment, and satisfiability. A sentence is valid, $\vDash \varphi$, if every structure satisfies it. It is entailed by a set of sentences, $\Gamma \vDash \varphi$, if every structure that satisfies all the sentences in Γ also satisfies φ. And a set of sentences is satisfiable if some structure satisfies all sentences in it at the same time.

Because formulas are inductively defined, and satisfaction is in turn defined by induction on the structure of formulas, we can use induction to prove properties of our semantics and to relate the semantic notions defined. We'll collect and prove some of these properties, partly because they are individually interesting, but mainly because many of them will come in handy when we go on to investigate the relation between semantics and derivation systems. In order

to do so, we'll also have to define (precisely, i.e., by induction) some syntactic notions and operations we haven't mentioned yet.

---

content/first-order-logic/introduction/substitution.tex

14.7 Substitution

fol:int:sub:sec

We'll discuss an example to illustrate how things hang together, and how the development of syntax and semantics lays the foundation for our more advanced investigations later. Our derivation systems should let us derive $P(a)$ from $\forall v_0\, P(v_0)$. Maybe we even want to state this as a rule of inference. However, to do so, we must be able to state it in the most general terms: not just for P, a, and v_0, but for any formula φ, and term t, and variable x. (Recall that constant symbols are terms, but we'll consider also more complicated terms built from constant symbols and function symbols.) So we want to be able to say something like, "whenever you have derived $\forall x\, \varphi(x)$ you are justified in inferring $\varphi(t)$—the result of removing $\forall x$ and replacing x by t." But what exactly does "replacing x by t" mean? What is the relation between $\varphi(x)$ and $\varphi(t)$? Does this always work?

To make this precise, we define the operation of *substitution*. Substitution is actually tricky, because we can't just replace all x's in φ by t, and not every t can be substituted for any x. We'll deal with this, again, using inductive definitions. But once this is done, specifying an inference rule as "infer $\varphi(t)$ from $\forall x\, \varphi(x)$" becomes a precise definition. Moreover, we'll be able to show that this is a good inference rule in the sense that $\forall x\, \varphi(x)$ entails $\varphi(t)$. But to prove this, we have to again prove something that may at first glance prompt you to ask "why are we doing this?" That $\forall x\, \varphi(x)$ entails $\varphi(t)$ relies on the fact that whether or not $\mathfrak{M} \vDash \varphi(t)$ holds depends only on the value of the term t, i.e., if we let m be whatever element of $|\mathfrak{M}|$ is picked out by t, then $\mathfrak{M}, s \vDash \varphi(t)$ iff $\mathfrak{M}, s[m/x] \vDash \varphi(x)$. This holds even when t contains variables, but we'll have to be careful with how exactly we state the result.

---

content/first-order-logic/introduction/models-theories.tex

14.8 Models and Theories

fol:int:mod:sec

Once we've defined the syntax and semantics of first-order logic, we can get to work investigating the properties of structures and the semantic notions. We can also define derivation systems, and investigate those. For a set of sentences, we can ask: what structures make all the sentences in that set true? Given a set of sentences Γ, a structure $\mathfrak{M}$ that satisfies them is called a *model of* Γ. We might start from Γ and try to find its models—what do they look like? How big or small do they have to be? But we might also start with a single structure or collection of structures and ask: what sentences are true in them? Are there sentences that *characterize* these structures in the sense that they,

and only they, are true in them? These kinds of questions are the domain of *model theory*. They also underlie the *axiomatic method*: describing a collection of structures by a set of sentences, the axioms of a theory. This is made possible by the observation that exactly those sentences entailed in first-order logic by the axioms are true in all models of the axioms.

As a very simple example, consider preorders. A preorder is a relation R on some set A which is both reflexive and transitive. A set A with a two-place relation $R \subseteq A \times A$ on it is exactly what we would need to give a structure for a first-order language with a single two-place relation symbol P: we would set $|\mathfrak{M}| = A$ and $P^{\mathfrak{M}} = R$. Since R is a preorder, it is reflexive and transitive, and we can find a set Γ of sentences of first-order logic that say this:

$$\forall v_0\, P(v_0, v_0)$$
$$\forall v_0\, \forall v_1\, \forall v_2\, ((P(v_0, v_1) \land P(v_1, v_2)) \to P(v_0, v_2))$$

These sentences are just the symbolizations of "for any x, Rxx" (R is reflexive) and "whenever Rxy and Ryz then also Rxz" (R is transitive). We see that a structure $\mathfrak{M}$ is a model of these two sentences Γ iff R (i.e., $P^{\mathfrak{M}}$), is a preorder on A (i.e., $|\mathfrak{M}|$). In other words, the models of Γ are exactly the preorders. Any property of all preorders that can be expressed in the first-order language with just P as predicate symbol (like reflexivity and transitivity above), is entailed by the two sentences in Γ and vice versa. So anything we can prove about models of Γ we have proved about all preorders.

For any particular theory and class of models (such as Γ and all preorders), there will be interesting questions about what can be expressed in the corresponding first-order language, and what cannot be expressed. There are some properties of structures that are interesting for all languages and classes of models, namely those concerning the size of the domain. One can always express, for instance, that the domain contains exactly n elements, for any $n \in \mathbb{Z}^+$. One can also express, using a set of infinitely many sentences, that the domain is infinite. But one cannot express that the domain is finite, or that the domain is non-enumerable. These results about the limitations of first-order languages are consequences of the compactness and Löwenheim–Skolem theorems.

---

content/first-order-logic/introduction/soundness-completeness.tex

14.9 Soundness and Completeness

fol:int:scp:sec

We'll also introduce derivation systems for first-order logic. There are many derivation systems that logicians have developed, but they all define the same derivability relation between sentences. We say that Γ *derives* φ, $\Gamma \vdash \varphi$, if there is a derivation of a certain precisely defined sort. Derivations are always finite arrangements of symbols—perhaps a list of sentences, or some more complicated structure. The purpose of derivation systems is to provide a tool to determine if a sentence is entailed by some set Γ. In order to serve that purpose, it must be true that $\Gamma \vDash \varphi$ if, and only if, $\Gamma \vdash \varphi$.

If $\Gamma \vdash \varphi$ but not $\Gamma \models \varphi$, our derivation system would be too strong, prove too much. The property that if $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$ is called *soundness*, and it is a minimal requirement on any good derivation system. On the other hand, if $\Gamma \models \varphi$ but not $\Gamma \vdash \varphi$, then our derivation system is too weak, it doesn't prove enough. The property that if $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$ is called *completeness*. Soundness is usually relatively easy to prove (by induction on the structure of derivations, which are inductively defined). Completeness is harder to prove.

Soundness and completeness have a number of important consequences. If a set of sentences Γ derives a contradiction (such as $\varphi \wedge \neg\varphi$) it is called *inconsistent*. Inconsistent Γs cannot have any models, they are unsatisfiable. From completeness the converse follows: any Γ that is not inconsistent—or, as we will say, *consistent*—has a model. In fact, this is equivalent to completeness, and is the form of completeness we will actually prove. It is a deep and perhaps surprising result: just because you cannot prove $\varphi \wedge \neg\varphi$ from Γ guarantees that there is a structure that is as Γ describes it. So completeness gives an answer to the question: which sets of sentences have models? Answer: all and only consistent sets do.

The soundness and completeness theorems have two important consequences: the compactness and the Löwenheim–Skolem theorem. These are important results in the theory of models, and can be used to establish many interesting results. We've already mentioned two: first-order logic cannot express that the domain of a structure is finite or that it is non-enumerable.

Historically, all of this—how to define syntax and semantics of first-order logic, how to define good derivation systems, how to prove that they are sound and complete, getting clear about what can and cannot be expressed in first-order languages—took a long time to figure out and get right. We now know how to do it, but going through all the details can still be confusing and tedious. But it's also important, because the methods developed here for the formal language of first-order logic are applied all over the place in logic, computer science, and linguistics. So working through the details pays off in the long run.

Chapter 15

Syntax of First-Order Logic

content/first-order-logic/syntax-and-semantics/intro-syntax.tex

15.1 Introduction

fol:syn:itx:sec

In order to develop the theory and metatheory of first-order logic, we must first define the syntax and semantics of its expressions. The expressions of first-order logic are terms and formulas. Terms are formed from variables, constant symbols, and function symbols. Formulas, in turn, are formed from predicate symbols together with terms (these form the smallest, "atomic" formulas), and then from atomic formulas we can form more complex ones using logical connectives and quantifiers. There are many different ways to set down the formation rules; we give just one possible one. Other systems will chose different symbols, will select different sets of connectives as primitive, will use parentheses differently (or even not at all, as in the case of so-called Polish notation). What all approaches have in common, though, is that the formation rules define the set of terms and formulas *inductively*. If done properly, every expression can result essentially in only one way according to the formation rules. The inductive definition resulting in expressions that are *uniquely readable* means we can give meanings to these expressions using the same method—inductive definition.

---

content/first-order-logic/syntax-and-semantics/first-order-languages.tex

15.2 First-Order Languages

fol:syn:fol:sec

Expressions of first-order logic are built up from a basic vocabulary containing *variables*, *constant symbols*, *predicate symbols* and sometimes *function symbols*. From them, together with logical connectives, quantifiers, and punctuation symbols such as parentheses and commas, *terms* and *formulas* are formed.

Informally, predicate symbols are names for properties and relations, constant symbols are names for individual objects, and function symbols are names for mappings. These, except for the identity predicate $=$, are the *non-logical symbols* and together make up a language. Any first-order language $\mathcal{L}$ is determined by its non-logical symbols. In the most general case, $\mathcal{L}$ contains infinitely many symbols of each kind. explanation

In the general case, we make use of the following symbols in first-order logic:

1. Logical symbols
    a) Logical connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (conditional), $\leftrightarrow$ (biconditional), $\forall$ (universal quantifier), $\exists$ (existential quantifier).
    b) The propositional constant for falsity $\bot$.
    c) The propositional constant for truth $\top$.
    d) The two-place identity predicate $=$.

e) A denumerable set of variables: v_0, v_1, v_2, ...

2. Non-logical symbols, making up the *standard language* of first-order logic

a) A denumerable set of n-place predicate symbols for each $n > 0$: A^n_0, A^n_1, A^n_2, ...

b) A denumerable set of constant symbols: c_0, c_1, c_2, ...

c) A denumerable set of n-place function symbols for each $n > 0$: f^n_0, f^n_1, f^n_2, ...

3. Punctuation marks: (,), and the comma.

Most of our definitions and results will be formulated for the full standard language of first-order logic. However, depending on the application, we may also restrict the language to only a few predicate symbols, constant symbols, and function symbols.

Example 15.1. The language $\mathcal{L}_A$ of arithmetic contains a single two-place predicate symbol $<$, a single constant symbol 0, one one-place function symbol $'$, and two two-place function symbols $+$ and $\times$.

Example 15.2. The language of set theory $\mathcal{L}_Z$ contains only the single two-place predicate symbol $\in$.

Example 15.3. The language of orders $\mathcal{L}_\leq$ contains only the two-place predicate symbol $\leq$.

Again, these are conventions: officially, these are just aliases, e.g., $<$, $\in$, and $\leq$ are aliases for A^2_0, 0 for c_0, $'$ for f^1_0, $+$ for f^2_0, $\times$ for f^2_1.

.

intro

You may be familiar with different terminology and symbols than the ones we use above. Logic texts (and teachers) commonly use $\sim$, $\neg$, or ! for "negation", $\wedge$, $\cdot$, or & for "conjunction". Commonly used symbols for the "conditional" or "implication" are $\rightarrow$, $\Rightarrow$, and $\supset$. Symbols for "biconditional," "bi-implication," or "(material) equivalence" are $\leftrightarrow$, $\Leftrightarrow$, and $\equiv$. The $\bot$ symbol is variously called "falsity," "falsum,", "absurdity," or "bottom." The $\top$ symbol is variously called "truth," "verum," or "top."

It is conventional to use lower case letters (e.g., a, b, c) from the beginning of the Latin alphabet for constant symbols (sometimes called names), and lower case letters from the end (e.g., x, y, z) for variables. Quantifiers combine with variables, e.g., x; notational variations include $\forall x$, $(\forall x)$, (x), Πx, $\bigwedge_x$ for the universal quantifier and $\exists x$, $(\exists x)$, (Ex), Σx, $\bigvee_x$ for the existential quantifier.

explanation

We might treat all the propositional operators and both quantifiers as primitive symbols of the language. We might instead choose a smaller stock of primitive symbols and treat the other logical operators as defined. "Truth functionally complete" sets of Boolean operators include $\{\neg, \vee\}$, $\{\neg, \wedge\}$, and

$\{\neg, \to\}$—these can be combined with either quantifier for an expressively complete first-order language.

You may be familiar with two other logical operators: the Sheffer stroke $|$ (named after Henry Sheffer), and Peirce's arrow $\downarrow$, also known as Quine's dagger. When given their usual readings of "nand" and "nor" (respectively), these operators are truth functionally complete by themselves.

---

content/first-order-logic/syntax-and-semantics/terms-formulas.tex

15.3 Terms and Formulas

fol:syn:frm: sec

Once a first-order language $\mathcal{L}$ is given, we can define expressions built up from the basic vocabulary of $\mathcal{L}$. These include in particular *terms* and *formulas*.

fol:syn:frm: defn:terms

Definition 15.4 (Terms). The set of *terms* $\mathrm{Trm}(\mathcal{L})$ of $\mathcal{L}$ is defined inductively by:

1. Every variable is a term.
2. Every constant symbol of $\mathcal{L}$ is a term.
3. If f is an n-place function symbol and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.
4. Nothing else is a term.

A term containing no variables is a *closed term*.

explanation

The constant symbols appear in our specification of the language and the terms as a separate category of symbols, but they could instead have been included as zero-place function symbols. We could then do without the second clause in the definition of terms. We just have to understand $f(t_1, \ldots, t_n)$ as just f by itself if $n = 0$.

fol:syn:frm: defn:formulas

Definition 15.5 (Formulas). The set of *formulas* $\mathrm{Frm}(\mathcal{L})$ of the language $\mathcal{L}$ is defined inductively as follows:

1. $\bot$ is an atomic formula.
2. $\top$ is an atomic formula.
3. If R is an n-place predicate symbol of $\mathcal{L}$ and $t_1, \ldots, t_n$ are terms of $\mathcal{L}$, then $R(t_1, \ldots, t_n)$ is an atomic formula.
4. If t_1 and t_2 are terms of $\mathcal{L}$, then $=(t_1, t_2)$ is an atomic formula.
5. If φ is a formula, then $\neg\varphi$ is a formula.
6. If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.
7. If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.

8. If φ and ψ are formulas, then $(\varphi \to \psi)$ is a formula.
9. If φ and ψ are formulas, then $(\varphi \leftrightarrow \psi)$ is a formula.
10. If φ is a formula and x is a variable, then $\forall x\, \varphi$ is a formula.
11. If φ is a formula and x is a variable, then $\exists x\, \varphi$ is a formula.
12. Nothing else is a formula.

explanation

The definitions of the set of terms and that of formulas are *inductive definitions*. Essentially, we construct the set of formulas in infinitely many stages. In the initial stage, we pronounce all atomic formulas to be formulas; this corresponds to the first few cases of the definition, i.e., the cases for $\top$, $\bot$, $R(t_1, \ldots, t_n)$ and $=(t_1, t_2)$. "Atomic formula" thus means any formula of this form.

The other cases of the definition give rules for constructing new formulas out of formulas already constructed. At the second stage, we can use them to construct formulas out of atomic formulas. At the third stage, we construct new formulas from the atomic formulas and those obtained in the second stage, and so on. A formula is anything that is eventually constructed at such a stage, and nothing else.

By convention, we write $=$ between its arguments and leave out the parentheses: $t_1 = t_2$ is an abbreviation for $=(t_1, t_2)$. Moreover, $\neg{=}(t_1, t_2)$ is abbreviated as $t_1 \neq t_2$. When writing a formula $(\psi * \chi)$ constructed from ψ, χ using a two-place connective $*$, we will often leave out the outermost pair of parentheses and write simply $\psi * \chi$.

intro

Some logic texts require that the variable x must occur in φ in order for $\exists x\, \varphi$ and $\forall x\, \varphi$ to count as formulas. Nothing bad happens if you don't require this, and it makes things easier.

If we work in a language for a specific application, we will often write two-place predicate symbols and function symbols between the respective terms, e.g., $t_1 < t_2$ and $(t_1 + t_2)$ in the language of arithmetic and $t_1 \in t_2$ in the language of set theory. The successor function in the language of arithmetic is even written conventionally *after* its argument: t'. Officially, however, these are just conventional abbreviations for $A^2_0(t_1, t_2)$, $f^2_0(t_1, t_2)$, $A^2_0(t_1, t_2)$ and $f^1_0(t)$, respectively.

Definition 15.6 (Syntactic identity). The symbol $\equiv$ expresses syntactic identity between strings of symbols, i.e., $\varphi \equiv \psi$ iff φ and ψ are strings of symbols of the same length and which contain the same symbol in each place.

The $\equiv$ symbol may be flanked by strings obtained by concatenation, e.g., $\varphi \equiv (\psi \vee \chi)$ means: the string of symbols φ is the same string as the one obtained by concatenating an opening parenthesis, the string ψ, the $\vee$ symbol, the string χ, and a closing parenthesis, in this order. If this is the case, then we know that the first symbol of φ is an opening parenthesis, φ contains ψ as a substring (starting at the second symbol), that substring is followed by $\vee$, etc.

As terms and formulas are built up from basic elements via inductive definitions, we can use the following induction principles to prove things about them.

fol:syn:frm:lem:trmind **Lemma 15.7 (*Principle of induction on terms*).** *Let $\mathcal{L}$ be a first-order language. If some property P is such that*

1. *it holds for every variable v,*
2. *it holds for every constant symbol a of $\mathcal{L}$, and*
3. *it holds for $f(t_1, \ldots, t_n)$ whenever it holds for t_1, …, t_n and f is an n-place function symbol of $\mathcal{L}$*

(assuming t_1, …, t_n are terms of $\mathcal{L}$), then P holds for every term in $\mathrm{Trm}(\mathcal{L})$.

Problem 15.1. Prove Lemma 15.7.

fol:syn:frm:thm:frmind **Lemma 15.8 (*Principle of induction on formulas*).** *Let $\mathcal{L}$ be a first-order language. If some property P holds for all the atomic formulas and is such that*

1. *it holds for $\neg\varphi$ whenever it holds for φ;*
2. *it holds for $(\varphi \wedge \psi)$ whenever it holds for φ and ψ;*
3. *it holds for $(\varphi \vee \psi)$ whenever it holds for φ and ψ;*
4. *it holds for $(\varphi \to \psi)$ whenever it holds for φ and ψ;*
5. *it holds for $(\varphi \leftrightarrow \psi)$ whenever it holds for φ and ψ;*
6. *it holds for $\exists x\, \varphi$ whenever it holds for φ;*
7. *it holds for $\forall x\, \varphi$ whenever it holds for φ;*

(assuming φ and ψ are formulas of $\mathcal{L}$), then P holds for all formulas in $\mathrm{Frm}(\mathcal{L})$.

---

content/first-order-logic/syntax-and-semantics/unique-readability.tex

15.4 Unique Readability

fol:syn:unq:sec The way we defined formulas guarantees that every formula has a *unique reading*, i.e., there is essentially only one way of constructing it according to our formation rules for formulas and only one way of "interpreting" it. If this were not so, we would have ambiguous formulas, i.e., formulas that have more than one reading or intepretation—and that is clearly something we want to avoid. But more importantly, without this property, most of the definitions and proofs we are going to give will not go through. explanation

Perhaps the best way to make this clear is to see what would happen if we had given bad rules for forming formulas that would not guarantee unique readability. For instance, we could have forgotten the parentheses in the formation rules for connectives, e.g., we might have allowed this:

> If φ and ψ are formulas, then so is $\varphi \to \psi$.

Starting from an atomic formula θ, this would allow us to form $\theta \to \theta$. From this, together with θ, we would get $\theta \to \theta \to \theta$. But there are two ways to do this:

1. We take θ to be φ and $\theta \to \theta$ to be ψ.
2. We take φ to be $\theta \to \theta$ and ψ is θ.

Correspondingly, there are two ways to "read" the formula $\theta \to \theta \to \theta$. It is of the form $\psi \to \chi$ where ψ is θ and χ is $\theta \to \theta$, but *it is also* of the form $\psi \to \chi$ with ψ being $\theta \to \theta$ and χ being θ.

If this happens, our definitions will not always work. For instance, when we define the main operator of a formula, we say: in a formula of the form $\psi \to \chi$, the main operator is the indicated occurrence of $\to$. But if we can match the formula $\theta \to \theta \to \theta$ with $\psi \to \chi$ in the two different ways mentioned above, then in one case we get the first occurrence of $\to$ as the main operator, and in the second case the second occurrence. But we intend the main operator to be a *function* of the formula, i.e., every formula must have exactly one main operator occurrence.

Lemma 15.9. *The number of left and right parentheses in a formula φ are equal.*

Proof. We prove this by induction on the way φ is constructed. This requires two things: (a) We have to prove first that all atomic formulas have the property in question (the induction basis). (b) Then we have to prove that when we construct new formulas out of given formulas, the new formulas have the property provided the old ones do.

Let $l(\varphi)$ be the number of left parentheses, and $r(\varphi)$ the number of right parentheses in φ, and $l(t)$ and $r(t)$ similarly the number of left and right parentheses in a term t.

Problem 15.2. Prove that for any term t, $l(t) = r(t)$.

1. $\varphi \equiv \bot$: φ has 0 left and 0 right parentheses.
2. $\varphi \equiv \top$: φ has 0 left and 0 right parentheses.
3. $\varphi \equiv R(t_1, \ldots, t_n)$: $l(\varphi) = 1 + l(t_1) + \cdots + l(t_n) = 1 + r(t_1) + \cdots + r(t_n) = r(\varphi)$. Here we make use of the fact, left as an exercise, that $l(t) = r(t)$ for any term t.
4. $\varphi \equiv t_1 = t_2$: $l(\varphi) = l(t_1) + l(t_2) = r(t_1) + r(t_2) = r(\varphi)$.

5. $\varphi \equiv \neg\psi$: By induction hypothesis, $l(\psi) = r(\psi)$. Thus $l(\varphi) = l(\psi) = r(\psi) = r(\varphi)$.
6. $\varphi \equiv (\psi * \chi)$: By induction hypothesis, $l(\psi) = r(\psi)$ and $l(\chi) = r(\chi)$. Thus $l(\varphi) = 1 + l(\psi) + l(\chi) = 1 + r(\psi) + r(\chi) = r(\varphi)$.
7. $\varphi \equiv \forall x\, \psi$: By induction hypothesis, $l(\psi) = r(\psi)$. Thus, $l(\varphi) = l(\psi) = r(\psi) = r(\varphi)$.
8. $\varphi \equiv \exists x\, \psi$: Similarly. □

Definition 15.10 (Proper prefix). A string of symbols ψ is a *proper prefix* of a string of symbols φ if concatenating ψ and a non-empty string of symbols yields φ.

fol:syn:unq: lem:no-prefix

Lemma 15.11. *If φ is a formula, and ψ is a proper prefix of φ, then ψ is not a formula.*

Proof. Exercise. □

Problem 15.3. Prove Lemma 15.11.

fol:syn:unq: prop:unique-atomic

Proposition 15.12. *If φ is an atomic formula, then it satisfies one, and only one of the following conditions.*

1. $\varphi \equiv \bot$.
2. $\varphi \equiv \top$.
3. $\varphi \equiv R(t_1, \ldots, t_n)$ *where R is an n-place predicate symbol, $t_1, \ldots, t_n$ are terms, and each of R, $t_1, \ldots, t_n$ is uniquely determined.*
4. $\varphi \equiv t_1 = t_2$ *where t_1 and t_2 are uniquely determined terms.*

Proof. Exercise. □

Problem 15.4. Prove Proposition 15.12 (Hint: Formulate and prove a version of Lemma 15.11 for terms.)

Proposition 15.13 (Unique Readability). *Every formula satisfies one, and only one of the following conditions.*

1. *φ is atomic.*
2. *φ is of the form $\neg\psi$.*
3. *φ is of the form $(\psi \wedge \chi)$.*
4. *φ is of the form $(\psi \vee \chi)$.*
5. *φ is of the form $(\psi \to \chi)$.*

6. φ *is of the form* $(\psi \leftrightarrow \chi)$.
7. φ *is of the form* $\forall x\, \psi$.
8. φ *is of the form* $\exists x\, \psi$.

Moreover, in each case ψ, or ψ and χ, are uniquely determined. This means that, e.g., there are no different pairs ψ, χ and ψ', χ' so that φ is both of the form $(\psi \to \chi)$ and $(\psi' \to \chi')$.

Proof. The formation rules require that if a formula is not atomic, it must start with an opening parenthesis (, $\neg$, or a quantifier. On the other hand, every formula that starts with one of the following symbols must be atomic: a predicate symbol, a function symbol, a constant symbol, $\bot$, $\top$.

So we really only have to show that if φ is of the form $(\psi * \chi)$ and also of the form $(\psi' *' \chi')$, then $\psi \equiv \psi'$, $\chi \equiv \chi'$, and $* = *'$.

So suppose both $\varphi \equiv (\psi * \chi)$ and $\varphi \equiv (\psi' *' \chi')$. Then either $\psi \equiv \psi'$ or not. If it is, clearly $* = *'$ and $\chi \equiv \chi'$, since they then are substrings of φ that begin in the same place and are of the same length. The other case is $\psi \not\equiv \psi'$. Since ψ and ψ' are both substrings of φ that begin at the same place, one must be a proper prefix of the other. But this is impossible by Lemma 15.11. □

---

content/first-order-logic/syntax-and-semantics/main-operator.tex

15.5 Main operator of a Formula

explanation

It is often useful to talk about the last operator used in constructing a formula φ. This operator is called the *main operator* of φ. Intuitively, it is the "outermost" operator of φ. For example, the main operator of $\neg\varphi$ is $\neg$, the main operator of $(\varphi \vee \psi)$ is $\vee$, etc.

fol:syn:main:sec

Definition 15.14 (Main operator). The *main operator* of a formula φ is defined as follows:

fol:syn:main:def:main-op

1. φ is atomic: φ has no main operator.
2. $\varphi \equiv \neg\psi$: the main operator of φ is $\neg$.
3. $\varphi \equiv (\psi \wedge \chi)$: the main operator of φ is $\wedge$.
4. $\varphi \equiv (\psi \vee \chi)$: the main operator of φ is $\vee$.
5. $\varphi \equiv (\psi \to \chi)$: the main operator of φ is $\to$.
6. $\varphi \equiv (\psi \leftrightarrow \chi)$: the main operator of φ is $\leftrightarrow$.
7. $\varphi \equiv \forall x\, \psi$: the main operator of φ is $\forall$.
8. $\varphi \equiv \exists x\, \psi$: the main operator of φ is $\exists$.

In each case, we intend the specific indicated *occurrence* of the main operator in the formula. For instance, since the formula $((\theta \to \alpha) \to (\alpha \to \theta))$ is of the form $(\psi \to \chi)$ where ψ is $(\theta \to \alpha)$ and χ is $(\alpha \to \theta)$, the second occurrence of $\to$ is the main operator.

This is a *recursive* definition of a function which maps all non-atomic formulas to their main operator occurrence. Because of the way formulas are defined inductively, every formula φ satisfies one of the cases in Definition 15.14. This guarantees that for each non-atomic formula φ a main operator exists. Because each formula satisfies only one of these conditions, and because the smaller formulas from which φ is constructed are uniquely determined in each case, the main operator occurrence of φ is unique, and so we have defined a function. explanation

We call formulas by the names in Table 15.1 depending on which symbol their main operator is.

Main operator	Type of formula	Example
none	atomic (formula)	$\bot$, $\top$, $R(t_1, \ldots, t_n)$, $t_1 = t_2$
$\neg$	negation	$\neg\varphi$
$\wedge$	conjunction	$(\varphi \wedge \psi)$
$\vee$	disjunction	$(\varphi \vee \psi)$
$\to$	conditional	$(\varphi \to \psi)$
$\leftrightarrow$	biconditional	$(\varphi \leftrightarrow \psi)$
$\forall$	universal (formula)	$\forall x\, \varphi$
$\exists$	existential (formula)	$\exists x\, \varphi$

Table 15.1: Main operator and names of formulas

fol:syn:mai: tab:main-op

content/first-order-logic/syntax-and-semantics/subformulas.tex

15.6 Subformulas

fol:syn:sbf: sec

It is often useful to talk about the formulas that "make up" a given formula. We call these its *subformulas*. Any formula counts as a subformula of itself; a subformula of φ other than φ itself is a *proper subformula*. explanation

Definition 15.15 (Immediate Subformula). If φ is a formula, the *immediate subformulas* of φ are defined inductively as follows:

1. Atomic formulas have no immediate subformulas.
2. $\varphi \equiv \neg\psi$: The only immediate subformula of φ is ψ.
3. $\varphi \equiv (\psi * \chi)$: The immediate subformulas of φ are ψ and χ ($*$ is any one of the two-place connectives).
4. $\varphi \equiv \forall x\, \psi$: The only immediate subformula of φ is ψ.
5. $\varphi \equiv \exists x\, \psi$: The only immediate subformula of φ is ψ.

Definition 15.16 (Proper Subformula). If φ is a formula, the *proper subformulas* of φ are defined recursively as follows:

1. Atomic formulas have no proper subformulas.
2. $\varphi \equiv \neg\psi$: The proper subformulas of φ are ψ together with all proper subformulas of ψ.
3. $\varphi \equiv (\psi * \chi)$: The proper subformulas of φ are ψ, χ, together with all proper subformulas of ψ and those of χ.
4. $\varphi \equiv \forall x\, \psi$: The proper subformulas of φ are ψ together with all proper subformulas of ψ.
5. $\varphi \equiv \exists x\, \psi$: The proper subformulas of φ are ψ together with all proper subformulas of ψ.

Definition 15.17 (Subformula). The subformulas of φ are φ itself together with all its proper subformulas.

explanation

Note the subtle difference in how we have defined immediate subformulas and proper subformulas. In the first case, we have directly defined the immediate subformulas of a formula φ for each possible form of φ. It is an explicit definition by cases, and the cases mirror the inductive definition of the set of formulas. In the second case, we have also mirrored the way the set of all formulas is defined, but in each case we have also included the proper subformulas of the smaller formulas ψ, χ in addition to these formulas themselves. This makes the definition *recursive*. In general, a definition of a function on an inductively defined set (in our case, formulas) is recursive if the cases in the definition of the function make use of the function itself. To be well defined, we must make sure, however, that we only ever use the values of the function for arguments that come "before" the one we are defining—in our case, when defining "proper subformula" for $(\psi * \chi)$ we only use the proper subformulas of the "earlier" formulas ψ and χ.

Proposition 15.18. *Suppose ψ is a subformula of φ and χ is a subformula of ψ. Then χ is a subformula of φ. In other words, the subformula relation is transitive.*

fol:syn:sbf: prop:subfrm-trans

Problem 15.5. Prove Proposition 15.18.

Proposition 15.19. *Suppose φ is a formula with n connectives and quantifiers. Then φ has at most $2n + 1$ subformulas.*

fol:syn:sbf: prop:count-subfrms

Problem 15.6. Prove Proposition 15.19.

---

content/first-order-logic/syntax-and-semantics/formation-sequences.tex

15.7 Formation Sequences

fol:syn:fseq:sec

Defining formulas via an inductive definition, and the complementary technique of proving properties of formulas via induction, is an elegant and efficient approach. However, it can also be useful to consider a more bottom-up, step-by-step approach to the construction of formulas, which we do here using the notion of a *formation sequence*. To show how terms and formulas can be introduced in this way without needing to refer to their inductive definitions, we first introduce the notion of an arbitrary string of symbols drawn from some language $\mathcal{L}$.

fol:syn:fseq:defn:string

Definition 15.20 (Strings). Suppose $\mathcal{L}$ is a first-order language. An $\mathcal{L}$-*string* is a finite sequence of symbols of $\mathcal{L}$. Where the language $\mathcal{L}$ is clearly fixed by the context, we will often refer to a $\mathcal{L}$-string simply as a *string*.

Example 15.21. For any first-order language $\mathcal{L}$, all $\mathcal{L}$-formulas are $\mathcal{L}$-strings, but not conversely. For example,

$$)(v_0 \to \exists$$

is an $\mathcal{L}$-string but not an $\mathcal{L}$-formula.

fol:syn:fseq:defn:fseq-trm

Definition 15.22 (Formation sequences for terms). A finite sequence of $\mathcal{L}$-strings $\langle t_0, \ldots, t_n \rangle$ is a *formation sequence* for a term t if $t \equiv t_n$ and for all $i \leq n$, either t_i is a variable or a constant symbol, or $\mathcal{L}$ contains a k-ary function symbol f and there exist $m_0, \ldots, m_k < i$ such that $t_i \equiv f(t_{m_0}, \ldots, t_{m_k})$. When it is necessary to distinguish, we will refer to formation sequences for terms as *term formation sequences*.

Example 15.23. The sequence

$$\langle c_0, v_0, f_0^2(c_0, v_0), f_0^1(f_0^2(c_0, v_0)) \rangle$$

is a formation sequence for the term $f_0^1(f_0^2(c_0, v_0))$, as is

$$\langle v_0, c_0, f_0^2(c_0, v_0), f_0^1(f_0^2(c_0, v_0)) \rangle.$$

fol:syn:fseq:defn:fseq-frm

Definition 15.24 (Formation sequences for formulas). A finite sequence of $\mathcal{L}$-strings $\langle \varphi_0, \ldots, \varphi_n \rangle$ is a *formation sequence* for φ if $\varphi \equiv \varphi_n$ and for all $i \leq n$, either φ_i is an atomic formula or there exist $j, k < i$ and a variable x such that one of the following holds:

1. $\varphi_i \equiv \neg\varphi_j$.
2. $\varphi_i \equiv (\varphi_j \wedge \varphi_k)$.
3. $\varphi_i \equiv (\varphi_j \vee \varphi_k)$.
4. $\varphi_i \equiv (\varphi_j \to \varphi_k)$.

5. $\varphi_i \equiv (\varphi_j \leftrightarrow \varphi_k)$.
6. $\varphi_i \equiv \forall x\, \varphi_j$.
7. $\varphi_i \equiv \exists x\, \varphi_j$.

When it is necessary to distinguish, we will refer to formation sequences for formulas as *formula formation sequences*.

Example 15.25.

$$\langle A^1_0(v_0), A^1_1(c_1), (A^1_1(c_1) \land A^1_0(v_0)), \exists v_0\, (A^1_1(c_1) \land A^1_0(v_0))\rangle$$

is a formation sequence of $\exists v_0\, (A^1_1(c_1) \land A^1_0(v_0))$, as is

$$\langle A^1_0(v_0), A^1_1(c_1), (A^1_1(c_1) \land A^1_0(v_0)), A^1_1(c_1), \forall v_1\, A^1_0(v_0), \exists v_0\, (A^1_1(c_1) \land A^1_0(v_0))\rangle.$$

As can be seen from the second example, formation sequences may contain "junk": formulas which are redundant or do not contribute to the construction.

Proposition 15.26. *Every formula φ in $\mathrm{Frm}(\mathcal{L})$ has a formation sequence.* *fol:syn:fseq: prop:formed*

Proof. Suppose φ is atomic. Then the sequence $\langle\varphi\rangle$ is a formation sequence for φ. Now suppose that ψ and χ have formation sequences $\langle\psi_0, \ldots, \psi_n\rangle$ and $\langle\chi_0, \ldots, \chi_m\rangle$ respectively.

1. If $\varphi \equiv \neg\psi$, then $\langle\psi_0, \ldots, \psi_n, \neg\psi_n\rangle$ is a formation sequence for φ.
2. If $\varphi \equiv (\psi \land \chi)$, then $\langle\psi_0, \ldots, \psi_n, \chi_0, \ldots, \chi_m, (\psi_n \land \chi_m)\rangle$ is a formation sequence for φ.
3. If $\varphi \equiv (\psi \lor \chi)$, then $\langle\psi_0, \ldots, \psi_n, \chi_0, \ldots, \chi_m, (\psi_n \lor \chi_m)\rangle$ is a formation sequence for φ.
4. If $\varphi \equiv (\psi \to \chi)$, then $\langle\psi_0, \ldots, \psi_n, \chi_0, \ldots, \chi_m, (\psi_n \to \chi_m)\rangle$ is a formation sequence for φ.
5. If $\varphi \equiv (\psi \leftrightarrow \chi)$, then $\langle\psi_0, \ldots, \psi_n, \chi_0, \ldots, \chi_m, (\psi_n \leftrightarrow \chi_m)\rangle$ is a formation sequence for φ.
6. If $\varphi \equiv \forall x\, \psi$, then $\langle\psi_0, \ldots, \psi_n, \forall x\, \psi_n\rangle$ is a formation sequence for φ.
7. If $\varphi \equiv \exists x\, \psi$, then $\langle\psi_0, \ldots, \psi_n, \exists x\, \psi_n\rangle$ is a formation sequence for φ.

By the principle of induction on formulas, every formula has a formation sequence. □

We can also prove the converse. This is important because it shows that our two ways of defining formulas are equivalent: they give the same results. It also means that we can prove theorems about formulas by using ordinary induction on the length of formation sequences.

fol:syn:fseq: lem:fseq-init **Lemma 15.27.** *Suppose that $\langle \varphi_0, \ldots, \varphi_n \rangle$ is a formation sequence for φ_n, and that $k \leq n$. Then $\langle \varphi_0, \ldots, \varphi_k \rangle$ is a formation sequence for φ_k.*

Proof. Exercise. □

Problem 15.7. Prove Lemma 15.27.

fol:syn:fseq: thm:fseq-frm-equiv **Theorem 15.28.** $\mathrm{Frm}(\mathcal{L})$ *is the set of all $\mathcal{L}$-strings φ such that there exists a formula formation sequence for φ.*

Proof. Let F be the set of all strings of symbols in the language $\mathcal{L}$ that have a formation sequence. We have seen in Proposition 15.26 that $\mathrm{Frm}(\mathcal{L}) \subseteq F$, so now we prove the converse.

Suppose φ has a formation sequence $\langle \varphi_0, \ldots, \varphi_n \rangle$. We prove that $\varphi \in \mathrm{Frm}(\mathcal{L})$ by strong induction on n. Our induction hypothesis is that every string of symbols with a formation sequence of length $m < n$ is in $\mathrm{Frm}(\mathcal{L})$. By the definition of a formation sequence, either $\varphi \equiv \varphi_n$ is atomic or there must exist $j, k < n$ such that one of the following is the case:

1. $\varphi \equiv \neg\varphi_j$.
2. $\varphi \equiv (\varphi_j \wedge \varphi_k)$.
3. $\varphi \equiv (\varphi_j \vee \varphi_k)$.
4. $\varphi \equiv (\varphi_j \to \varphi_k)$.
5. $\varphi \equiv (\varphi_j \leftrightarrow \varphi_k)$.
6. $\varphi \equiv \forall x\, \varphi_j$.
7. $\varphi \equiv \exists x\, \varphi_j$.

Now we reason by cases. If φ is atomic then $\varphi_n \in \mathrm{Frm}(\mathcal{L}_0)$. Suppose instead that $\varphi \equiv (\varphi_j \wedge \varphi_k)$. By Lemma 15.27, $\langle \varphi_0, \ldots, \varphi_j \rangle$ and $\langle \varphi_0, \ldots, \varphi_k \rangle$ are formation sequences for φ_j and φ_k, respectively. Since these are proper initial subsequences of the formation sequence for φ, they both have length less than n. Therefore by the induction hypothesis, φ_j and φ_k are in $\mathrm{Frm}(\mathcal{L}_0)$, and by the definition of a formula, so is $(\varphi_j \wedge \varphi_k)$. The other cases follow by parallel reasoning. □

Formation sequences for terms have similar properties to those for formulas.

fol:syn:fseq: prop:fseq-trm-equiv **Proposition 15.29.** $\mathrm{Trm}(\mathcal{L})$ *is the set of all $\mathcal{L}$-strings t such that there exists a term formation sequence for t.*

Proof. Exercise. □

Problem 15.8. Prove Proposition 15.29. Hint: use a similar strategy to that used in the proof of Theorem 15.28.

There are two types of "junk" that can appear in formation sequences: repeated elements, and elements that are irrelevant to the construction of the formation or term. We can eliminate both by looking at minimal formation sequences.

Definition 15.30 (Minimal formation sequences). A formation sequence $\langle \varphi_0, \ldots, \varphi_n \rangle$ for a formula φ is a *minimal formation sequence* for φ if for every other formation sequence s for φ, the length of s is greater than or equal to $n+1$. fol:syn:fseq: defn:minimal-fseq

Similarly, a formation sequence $\langle t_0, \ldots, t_n \rangle$ for a term t is a *minimal formation sequence* for t if for every other formation sequence s for t, the length of s is greater than or equal to $n+1$.

Note that a formula or term can have more than one minimal formation sequence, but they will contain exactly the same strings.

Proposition 15.31. *The following are equivalent:* fol:syn:fseq: prop:subformula-equivs

1. *ψ is a sub-formula of φ.*
2. *ψ occurs in every formation sequence of φ.*
3. *ψ occurs in a minimal formation sequence of φ.*

Proof. Exercise. □

Problem 15.9. Prove Proposition 15.31.

Historical Remarks Formation sequences were introduced by Raymond Smullyan in his textbook *First-Order Logic* (Smullyan, 1968). Additional properties of formation sequences were established by Zuckerman (1973).

---

content/first-order-logic/syntax-and-semantics/free-vars-sentences.tex

15.8 Free Variables and Sentences

Definition 15.32 (Free occurrences of a variable). The *free* occurrences of a variable in a formula are defined inductively as follows: fol:syn:fvs: sec fol:syn:fvs: defn:free-occ

1. φ is atomic: all variable occurrences in φ are free.
2. $\varphi \equiv \neg\psi$: the free variable occurrences of φ are exactly those of ψ.
3. $\varphi \equiv (\psi * \chi)$: the free variable occurrences of φ are those in ψ together with those in χ.
4. $\varphi \equiv \forall x\, \psi$: the free variable occurrences in φ are all of those in ψ except for occurrences of x.

5. $\varphi \equiv \exists x\, \psi$: the free variable occurrences in φ are all of those in ψ except for occurrences of x.

Definition 15.33 (Bound Variables). An occurrence of a variable in a formula φ is *bound* if it is not free.

Problem 15.10. Give an inductive definition of the bound variable occurrences along the lines of Definition 15.32.

Definition 15.34 (Scope). If $\forall x\, \psi$ is an occurrence of a subformula in a formula φ, then the corresponding occurrence of ψ in φ is called the *scope* of the corresponding occurrence of $\forall x$. Similarly for $\exists x$.

If ψ is the scope of a quantifier occurrence $\forall x$ or $\exists x$ in φ, then the free occurrences of x in ψ are bound in $\forall x\, \psi$ and $\exists x\, \psi$. We say that these occurrences are *bound by* the mentioned quantifier occurrence.

Example 15.35. Consider the following formula:

$$\exists v_0\ \underbrace{A^2_0(v_0, v_1)}_{\psi}$$

ψ represents the scope of $\exists v_0$. The quantifier binds the occurrence of v_0 in ψ, but does not bind the occurrence of v_1. So v_1 is a free variable in this case.

We can now see how this might work in a more complicated formula φ:

$$\forall v_0\ \underbrace{(A^1_0(v_0) \to A^2_0(v_0, v_1))}_{\psi} \to \exists v_1\ \underbrace{(A^2_1(v_0, v_1) \vee \forall v_0\ \overbrace{\neg A^1_1(v_0)}^{\theta})}_{\chi}$$

ψ is the scope of the first $\forall v_0$, χ is the scope of $\exists v_1$, and θ is the scope of the second $\forall v_0$. The first $\forall v_0$ binds the occurrences of v_0 in ψ, $\exists v_1$ binds the occurrence of v_1 in χ, and the second $\forall v_0$ binds the occurrence of v_0 in θ. The first occurrence of v_1 and the fourth occurrence of v_0 are free in φ. The last occurrence of v_0 is free in θ, but bound in χ and φ.

Definition 15.36 (Sentence). A formula φ is a *sentence* iff it contains no free occurrences of variables.

---

content/first-order-logic/syntax-and-semantics/substitution.tex

15.9 Substitution

fol:syn:sub:sec

Definition 15.37 (Substitution in a term). We define $s[t/x]$, the result of *substituting* t for every occurrence of x in s, recursively:

1. $s \equiv c$: $s[t/x]$ is just s.
2. $s \equiv y$: $s[t/x]$ is also just s, provided y is a variable and $y \not\equiv x$.
3. $s \equiv x$: $s[t/x]$ is t.
4. $s \equiv f(t_1, \ldots, t_n)$: $s[t/x]$ is $f(t_1[t/x], \ldots, t_n[t/x])$.

Definition 15.38. A term t is *free for* x in φ if none of the free occurrences of x in φ occur in the scope of a quantifier that binds a variable in t.

Example 15.39.

1. v_8 is free for v_1 in $\exists v_3 A_4^2(v_3, v_1)$
2. $f_1^2(v_1, v_2)$ is *not* free for v_0 in $\forall v_2 A_4^2(v_0, v_2)$

Definition 15.40 (Substitution in a formula). If φ is a formula, x is a variable, and t is a term free for x in φ, then $\varphi[t/x]$ is the result of substituting t for all free occurrences of x in φ.

1. $\varphi \equiv \bot$: $\varphi[t/x]$ is $\bot$.
2. $\varphi \equiv \top$: $\varphi[t/x]$ is $\top$.
3. $\varphi \equiv P(t_1, \ldots, t_n)$: $\varphi[t/x]$ is $P(t_1[t/x], \ldots, t_n[t/x])$.
4. $\varphi \equiv t_1 = t_2$: $\varphi[t/x]$ is $t_1[t/x] = t_2[t/x]$.
5. $\varphi \equiv \neg\psi$: $\varphi[t/x]$ is $\neg\psi[t/x]$.
6. $\varphi \equiv (\psi \wedge \chi)$: $\varphi[t/x]$ is $(\psi[t/x] \wedge \chi[t/x])$.
7. $\varphi \equiv (\psi \vee \chi)$: $\varphi[t/x]$ is $(\psi[t/x] \vee \chi[t/x])$.
8. $\varphi \equiv (\psi \to \chi)$: $\varphi[t/x]$ is $(\psi[t/x] \to \chi[t/x])$.
9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\varphi[t/x]$ is $(\psi[t/x] \leftrightarrow \chi[t/x])$.
10. $\varphi \equiv \forall y\, \psi$: $\varphi[t/x]$ is $\forall y\, \psi[t/x]$, provided y is a variable other than x; otherwise $\varphi[t/x]$ is just φ.
11. $\varphi \equiv \exists y\, \psi$: $\varphi[t/x]$ is $\exists y\, \psi[t/x]$, provided y is a variable other than x; otherwise $\varphi[t/x]$ is just φ.

explanation

Note that substitution may be vacuous: If x does not occur in φ at all, then $\varphi[t/x]$ is just φ.

The restriction that t must be free for x in φ is necessary to exclude cases like the following. If $\varphi \equiv \exists y\, x < y$ and $t \equiv y$, then $\varphi[t/x]$ would be $\exists y\, y < y$. In this case the free variable y is "captured" by the quantifier $\exists y$ upon substitution, and that is undesirable. For instance, we would like it to be the case that whenever $\forall x\, \psi$ holds, so does $\psi[t/x]$. But consider $\forall x \exists y\, x < y$ (here ψ is

$\exists y\, x < y$). It is a sentence that is true about, e.g., the natural numbers: for every number x there is a number y greater than it. If we allowed y as a possible substitution for x, we would end up with $\psi[y/x] \equiv \exists y\, y < y$, which is false. We prevent this by requiring that none of the free variables in t would end up being bound by a quantifier in φ.

We often use the following convention to avoid cumbersome notation: If φ is a formula which may contain the variable x free, we also write $\varphi(x)$ to indicate this. When it is clear which φ and x we have in mind, and t is a term (assumed to be free for x in $\varphi(x)$), then we write $\varphi(t)$ as short for $\varphi[t/x]$. So for instance, we might say, "we call $\varphi(t)$ an instance of $\forall x\, \varphi(x)$." By this we mean that if φ is any formula, x a variable, and t a term that's free for x in φ, then $\varphi[t/x]$ is an instance of $\forall x\, \varphi$.

Chapter 16

Semantics of First-Order Logic

---

content/first-order-logic/syntax-and-semantics/intro-semantics.tex

16.1 Introduction

fol:syn:its:sec

Giving the meaning of expressions is the domain of semantics. The central concept in semantics is that of satisfaction in a structure. A structure gives meaning to the building blocks of the language: a domain is a non-empty set of objects. The quantifiers are interpreted as ranging over this domain, constant symbols are assigned elements in the domain, function symbols are assigned functions from the domain to itself, and predicate symbols are assigned relations on the domain. The domain together with assignments to the basic vocabulary constitutes a structure. Variables may appear in formulas, and in order to give a semantics, we also have to assign elements of the domain to them—this is a variable assignment. The satisfaction relation, finally, brings these together. A formula may be satisfied in a structure $\mathfrak{M}$ relative to a variable assignment s, written as $\mathfrak{M}, s \vDash \varphi$. This relation is also defined by induction on the structure of φ, using the truth tables for the logical connectives to define, say, satisfaction of $(\varphi \wedge \psi)$ in terms of satisfaction (or not) of φ and ψ. It then turns out that the variable assignment is irrelevant if

the formula φ is a sentence, i.e., has no free variables, and so we can talk of sentences being simply satisfied (or not) in structures.

On the basis of the satisfaction relation $\mathfrak{M} \vDash \varphi$ for sentences we can then define the basic semantic notions of validity, entailment, and satisfiability. A sentence is valid, $\vDash \varphi$, if every structure satisfies it. It is entailed by a set of sentences, $\Gamma \vDash \varphi$, if every structure that satisfies all the sentences in Γ also satisfies φ. And a set of sentences is satisfiable if some structure satisfies all sentences in it at the same time. Because formulas are inductively defined, and satisfaction is in turn defined by induction on the structure of formulas, we can use induction to prove properties of our semantics and to relate the semantic notions defined.

---

content/first-order-logic/syntax-and-semantics/structures.tex

16.2 Structures for First-order Languages

explanation

First-order languages are, by themselves, *uninterpreted:* the constant symbols, function symbols, and predicate symbols have no specific meaning attached to them. Meanings are given by specifying a *structure*. It specifies the *domain*, i.e., the objects which the constant symbols pick out, the function symbols operate on, and the quantifiers range over. In addition, it specifies which constant symbols pick out which objects, how a function symbol maps objects to objects, and which objects the predicate symbols apply to. Structures are the basis for *semantic* notions in logic, e.g., the notion of consequence, validity, satisfiability. They are variously called "structures," "interpretations," or "models" in the literature.

fol:syn:str:sec

Definition 16.1 (Structures). A *structure* $\mathfrak{M}$, for a language $\mathcal{L}$ of first-order logic consists of the following elements:

1. *Domain:* a non-empty set, $|\mathfrak{M}|$
2. *Interpretation of constant symbols:* for each constant symbol c of $\mathcal{L}$, an element $c^{\mathfrak{M}} \in |\mathfrak{M}|$
3. *Interpretation of predicate symbols:* for each n-place predicate symbol R of $\mathcal{L}$ (other than $=$), an n-place relation $R^{\mathfrak{M}} \subseteq |\mathfrak{M}|^n$
4. *Interpretation of function symbols:* for each n-place function symbol f of $\mathcal{L}$, an n-place function $f^{\mathfrak{M}} \colon |\mathfrak{M}|^n \to |\mathfrak{M}|$

Example 16.2. A structure $\mathfrak{M}$ for the language of arithmetic consists of a set, an element of $|\mathfrak{M}|$, $\mathsf{o}^{\mathfrak{M}}$, as interpretation of the constant symbol o, a one-place function $\mathsf{'}^{\mathfrak{M}} \colon |\mathfrak{M}| \to |\mathfrak{M}|$, two two-place functions $+^{\mathfrak{M}}$ and $\times^{\mathfrak{M}}$, both $|\mathfrak{M}|^2 \to |\mathfrak{M}|$, and a two-place relation $<^{\mathfrak{M}} \subseteq |\mathfrak{M}|^2$.

An obvious example of such a structure is the following:

1. $|\mathfrak{N}| = \mathbb{N}$

2. $\mathsf{o}^{\mathfrak{N}} = 0$
3. $\mathsf{'}^{\mathfrak{N}}(n) = n + 1$ for all $n \in \mathbb{N}$
4. $+^{\mathfrak{N}}(n, m) = n + m$ for all $n, m \in \mathbb{N}$
5. $\times^{\mathfrak{N}}(n, m) = n \cdot m$ for all $n, m \in \mathbb{N}$
6. $<^{\mathfrak{N}} = \{\langle n, m\rangle : n \in \mathbb{N}, m \in \mathbb{N}, n < m\}$

The structure $\mathfrak{N}$ for $\mathcal{L}_A$ so defined is called the *standard model of arithmetic*, because it interprets the non-logical constants of $\mathcal{L}_A$ exactly how you would expect.

However, there are many other possible structures for $\mathcal{L}_A$. For instance, we might take as the domain the set $\mathbb{Z}$ of integers instead of $\mathbb{N}$, and define the interpretations of o, $\mathsf{'}$, $+$, $\times$, $<$ accordingly. But we can also define structures for $\mathcal{L}_A$ which have nothing even remotely to do with numbers.

Example 16.3. A structure $\mathfrak{M}$ for the language $\mathcal{L}_Z$ of set theory requires just a set and a single-two place relation. So technically, e.g., the set of people plus the relation "x is older than y" could be used as a structure for $\mathcal{L}_Z$, as well as $\mathbb{N}$ together with $n \geq m$ for $n, m \in \mathbb{N}$.

A particularly interesting structure for $\mathcal{L}_Z$ in which the elements of the domain are actually sets, and the interpretation of $\in$ actually is the relation "x is an element of y" is the structure $\mathfrak{HF}$ of *hereditarily finite sets*:

1. $|\mathfrak{HF}| = \emptyset \cup \wp(\emptyset) \cup \wp(\wp(\emptyset)) \cup \wp(\wp(\wp(\emptyset))) \cup \ldots$;
2. $\in^{\mathfrak{HF}} = \{\langle x, y\rangle : x, y \in |\mathfrak{HF}|, x \in y\}$.

The stipulations we make as to what counts as a structure impact our logic. For example, the choice to prevent empty domains ensures, given the usual account of satisfaction (or truth) for quantified sentences, that $\exists x\,(\varphi(x) \vee \neg\varphi(x))$ is valid—that is, a logical truth. And the stipulation that all constant symbols must refer to an object in the domain ensures that the existential generalization is a sound pattern of inference: $\varphi(a)$, therefore $\exists x\,\varphi(x)$. If we allowed names to refer outside the domain, or to not refer, then we would be on our way to a *free logic*, in which existential generalization requires an additional premise: $\varphi(a)$ and $\exists x\, x = a$, therefore $\exists x\,\varphi(x)$. digression

content/first-order-logic/syntax-and-semantics/covered-structures.tex

16.3 Covered Structures for First-order Languages

fol:syn:cov:sec

Recall that a term is *closed* if it contains no variables. explanation

Definition 16.4 (Value of closed terms). If t is a closed term of the language $\mathcal{L}$ and $\mathfrak{M}$ is a structure for $\mathcal{L}$, the *value* $\mathrm{Val}^{\mathfrak{M}}(t)$ is defined as follows:

1. If t is just the constant symbol c, then $\mathrm{Val}^{\mathfrak{M}}(c) = c^{\mathfrak{M}}$.
2. If t is of the form $f(t_1, \ldots, t_n)$, then

$$\mathrm{Val}^{\mathfrak{M}}(t) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}(t_n)).$$

Definition 16.5 (Covered structure). A structure is *covered* if every element of the domain is the value of some closed term.

Example 16.6. Let $\mathcal{L}$ be the language with constant symbols zero, one, two, ..., the binary predicate symbol $<$, and the binary function symbols $+$ and $\times$. Then a structure $\mathfrak{M}$ for $\mathcal{L}$ is the one with domain $|\mathfrak{M}| = \{0, 1, 2, \ldots\}$ and assignments $\mathit{zero}^{\mathfrak{M}} = 0$, $\mathit{one}^{\mathfrak{M}} = 1$, $\mathit{two}^{\mathfrak{M}} = 2$, and so forth. For the binary relation symbol $<$, the set $<^{\mathfrak{M}}$ is the set of all pairs $\langle c_1, c_2 \rangle \in |\mathfrak{M}|^2$ such that c_1 is less than c_2: for example, $\langle 1, 3 \rangle \in <^{\mathfrak{M}}$ but $\langle 2, 2 \rangle \notin <^{\mathfrak{M}}$. For the binary function symbol $+$, define $+^{\mathfrak{M}}$ in the usual way—for example, $+^{\mathfrak{M}}(2, 3)$ maps to 5, and similarly for the binary function symbol $\times$. Hence, the value of four is just 4, and the value of $\times(\mathit{two}, +(\mathit{three}, \mathit{zero}))$ (or in infix notation, $\mathit{two} \times (\mathit{three} + \mathit{zero})$) is

$$\begin{aligned}
\mathrm{Val}^{\mathfrak{M}}(\times(\mathit{two}, +(\mathit{three}, \mathit{zero}))) = \\
&= \times^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(\mathit{two}), \mathrm{Val}^{\mathfrak{M}}(+(\mathit{three}, \mathit{zero}))) \\
&= \times^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(\mathit{two}), +^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(\mathit{three}), \mathrm{Val}^{\mathfrak{M}}(\mathit{zero}))) \\
&= \times^{\mathfrak{M}}(\mathit{two}^{\mathfrak{M}}, +^{\mathfrak{M}}(\mathit{three}^{\mathfrak{M}}, \mathit{zero}^{\mathfrak{M}})) \\
&= \times^{\mathfrak{M}}(2, +^{\mathfrak{M}}(3, 0)) \\
&= \times^{\mathfrak{M}}(2, 3) \\
&= 6
\end{aligned}$$

Problem 16.1. Is $\mathfrak{N}$, the standard model of arithmetic, covered? Explain.

---

content/first-order-logic/syntax-and-semantics/satisfaction.tex

16.4 Satisfaction of a Formula in a Structure

explanation

The basic notion that relates expressions such as terms and formulas, on the one hand, and structures on the other, are those of *value* of a term and *satisfaction* of a formula. Informally, the value of a term is an element of a structure—if the term is just a constant, its value is the object assigned to the constant by the structure, and if it is built up using function symbols, the value is computed from the values of constants and the functions assigned to the functions in the term. A formula is *satisfied* in a structure if the interpretation given to the predicates makes the formula true in the domain of the structure. This notion of satisfaction is specified inductively: the specification of the structure directly

fol:syn:sat:sec

states when atomic formulas are satisfied, and we define when a complex formula is satisfied depending on the main connective or quantifier and whether or not the immediate subformulas are satisfied.

The case of the quantifiers here is a bit tricky, as the immediate subformula of a quantified formula has a free variable, and structures don't specify the values of variables. In order to deal with this difficulty, we also introduce *variable assignments* and define satisfaction not with respect to a structure alone, but with respect to a structure plus a variable assignment.

Definition 16.7 (Variable Assignment). A *variable assignment* s for a structure $\mathfrak{M}$ is a function which maps each variable to an element of $|\mathfrak{M}|$, i.e., $s\colon \mathrm{Var} \to |\mathfrak{M}|$.

A structure assigns a value to each constant symbol, and a variable assignment to each variable. But we want to use terms built up from them to also name elements of the domain. For this we define the value of terms inductively. For constant symbols and variables the value is just as the structure or the variable assignment specifies it; for more complex terms it is computed recursively using the functions the structure assigns to the function symbols. explanation

Definition 16.8 (Value of Terms). If t is a term of the language $\mathcal{L}$, $\mathfrak{M}$ is a structure for $\mathcal{L}$, and s is a variable assignment for $\mathfrak{M}$, the *value* $\mathrm{Val}^{\mathfrak{M}}_s(t)$ is defined as follows:

1. $t \equiv c$: $\mathrm{Val}^{\mathfrak{M}}_s(t) = c^{\mathfrak{M}}$.
2. $t \equiv x$: $\mathrm{Val}^{\mathfrak{M}}_s(t) = s(x)$.
3. $t \equiv f(t_1, \ldots, t_n)$:

$$\mathrm{Val}^{\mathfrak{M}}_s(t) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_s(t_n)).$$

Definition 16.9 (x-Variant). If s is a variable assignment for a structure $\mathfrak{M}$, then any variable assignment s' for $\mathfrak{M}$ which differs from s at most in what it assigns to x is called an *x-variant* of s. If s' is an x-variant of s we write $s' \sim_x s$.

Note that an x-variant of an assignment s does not *have* to assign something different to x. In fact, every assignment counts as an x-variant of itself. explanation

Definition 16.10. If s is a variable assignment for a structure $\mathfrak{M}$ and $m \in |\mathfrak{M}|$, then the assignment $s[m/x]$ is the variable assignment defined by

$$s[m/x](y) = \begin{cases} m & \text{if } y \equiv x \\ s(y) & \text{otherwise.} \end{cases}$$

In other words, $s[m/x]$ is the particular x-variant of s which assigns the domain element m to x, and assigns the same things to variables other than x that s does.

Definition 16.11 (Satisfaction). Satisfaction of a formula φ in a structure $\mathfrak{M}$ relative to a variable assignment s, in symbols: $\mathfrak{M}, s \vDash \varphi$, is defined recursively as follows. (We write $\mathfrak{M}, s \nvDash \varphi$ to mean "not $\mathfrak{M}, s \vDash \varphi$.") fol:syn:sat: defn:satisfaction

1. $\varphi \equiv \bot$: $\mathfrak{M}, s \nvDash \varphi$.
2. $\varphi \equiv \top$: $\mathfrak{M}, s \vDash \varphi$.
3. $\varphi \equiv R(t_1, \ldots, t_n)$: $\mathfrak{M}, s \vDash \varphi$ iff $\langle \mathrm{Val}^{\mathfrak{M}}_s(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_s(t_n)\rangle \in R^{\mathfrak{M}}$.
4. $\varphi \equiv t_1 = t_2$: $\mathfrak{M}, s \vDash \varphi$ iff $\mathrm{Val}^{\mathfrak{M}}_s(t_1) = \mathrm{Val}^{\mathfrak{M}}_s(t_2)$.
5. $\varphi \equiv \neg\psi$: $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s \nvDash \psi$.
6. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s \vDash \psi$ and $\mathfrak{M}, s \vDash \chi$.
7. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s \vDash \psi$ or $\mathfrak{M}, s \vDash \chi$ (or both).
8. $\varphi \equiv (\psi \to \chi)$: $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s \nvDash \psi$ or $\mathfrak{M}, s \vDash \chi$ (or both).
9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, s \vDash \varphi$ iff either both $\mathfrak{M}, s \vDash \psi$ and $\mathfrak{M}, s \vDash \chi$, or neither $\mathfrak{M}, s \vDash \psi$ nor $\mathfrak{M}, s \vDash \chi$.
10. $\varphi \equiv \forall x\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff for every element $m \in |\mathfrak{M}|$, $\mathfrak{M}, s[m/x] \vDash \psi$.
11. $\varphi \equiv \exists x\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff for at least one element $m \in |\mathfrak{M}|$, $\mathfrak{M}, s[m/x] \vDash \psi$.

explanation

The variable assignments are important in the last two clauses. We cannot define satisfaction of $\forall x\, \psi(x)$ by "for all $m \in |\mathfrak{M}|$, $\mathfrak{M} \vDash \psi(m)$." We cannot define satisfaction of $\exists x\, \psi(x)$ by "for at least one $m \in |\mathfrak{M}|$, $\mathfrak{M} \vDash \psi(m)$." The reason is that if $m \in |\mathfrak{M}|$, it is not a symbol of the language, and so $\psi(m)$ is not a formula (that is, $\psi[m/x]$ is undefined). We also cannot assume that we have constant symbols or terms available that name every element of $\mathfrak{M}$, since there is nothing in the definition of structures that requires it. In the standard language, the set of constant symbols is denumerable, so if $|\mathfrak{M}|$ is not enumerable there aren't even enough constant symbols to name every object.

We solve this problem by introducing variable assignments, which allow us to link variables directly with elements of the domain. Then instead of saying that, e.g., $\exists x\, \psi(x)$ is satisfied in $\mathfrak{M}$ iff for at least one $m \in |\mathfrak{M}|$, we say it is satisfied in $\mathfrak{M}$ *relative to* s iff $\psi(x)$ is satisfied relative to $s[m/x]$ for at least one $m \in |\mathfrak{M}|$.

Example 16.12. Let $\mathcal{L} = \{a, b, f, R\}$ where a and b are constant symbols, f is a two-place function symbol, and R is a two-place predicate symbol. Consider the structure $\mathfrak{M}$ defined by:

1. $|\mathfrak{M}| = \{1, 2, 3, 4\}$
2. $a^{\mathfrak{M}} = 1$
3. $b^{\mathfrak{M}} = 2$

4. $f^{\mathfrak{M}}(x, y) = x + y$ if $x + y \leq 3$ and $= 3$ otherwise.

5. $R^{\mathfrak{M}} = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \langle 2, 4\rangle\}$

The function $s(x) = 1$ that assigns $1 \in |\mathfrak{M}|$ to every variable is a variable assignment for $\mathfrak{M}$.

Then

$$\mathrm{Val}^{\mathfrak{M}}_s(f(a, b)) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(a), \mathrm{Val}^{\mathfrak{M}}_s(b)).$$

Since a and b are constant symbols, $\mathrm{Val}^{\mathfrak{M}}_s(a) = a^{\mathfrak{M}} = 1$ and $\mathrm{Val}^{\mathfrak{M}}_s(b) = b^{\mathfrak{M}} = 2$. So

$$\mathrm{Val}^{\mathfrak{M}}_s(f(a, b)) = f^{\mathfrak{M}}(1, 2) = 1 + 2 = 3.$$

To compute the value of $f(f(a, b), a)$ we have to consider

$$\mathrm{Val}^{\mathfrak{M}}_s(f(f(a, b), a)) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(f(a, b)), \mathrm{Val}^{\mathfrak{M}}_s(a)) = f^{\mathfrak{M}}(3, 1) = 3,$$

since $3 + 1 > 3$. Since $s(x) = 1$ and $\mathrm{Val}^{\mathfrak{M}}_s(x) = s(x)$, we also have

$$\mathrm{Val}^{\mathfrak{M}}_s(f(f(a, b), x)) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(f(a, b)), \mathrm{Val}^{\mathfrak{M}}_s(x)) = f^{\mathfrak{M}}(3, 1) = 3,$$

An atomic formula $R(t_1, t_2)$ is satisfied if the tuple of values of its arguments, i.e., $\langle \mathrm{Val}^{\mathfrak{M}}_s(t_1), \mathrm{Val}^{\mathfrak{M}}_s(t_2)\rangle$, is an element of $R^{\mathfrak{M}}$. So, e.g., we have $\mathfrak{M}, s \vDash R(b, f(a, b))$ since $\langle \mathrm{Val}^{\mathfrak{M}}(b), \mathrm{Val}^{\mathfrak{M}}(f(a, b))\rangle = \langle 2, 3\rangle \in R^{\mathfrak{M}}$, but $\mathfrak{M}, s \nvDash R(x, f(a, b))$ since $\langle 1, 3\rangle \notin R^{\mathfrak{M}}[s]$.

To determine if a non-atomic formula φ is satisfied, you apply the clauses in the inductive definition that applies to the main connective. For instance, the main connective in $R(a, a) \to (R(b, x) \vee R(x, b))$ is the $\to$, and

$$\begin{aligned}&\mathfrak{M}, s \vDash R(a, a) \to (R(b, x) \vee R(x, b)) \text{ iff}\\ &\quad \mathfrak{M}, s \nvDash R(a, a) \text{ or } \mathfrak{M}, s \vDash R(b, x) \vee R(x, b)\end{aligned}$$

Since $\mathfrak{M}, s \vDash R(a, a)$ (because $\langle 1, 1\rangle \in R^{\mathfrak{M}}$) we can't yet determine the answer and must first figure out if $\mathfrak{M}, s \vDash R(b, x) \vee R(x, b)$:

$$\begin{aligned}&\mathfrak{M}, s \vDash R(b, x) \vee R(x, b) \text{ iff}\\ &\quad \mathfrak{M}, s \vDash R(b, x) \text{ or } \mathfrak{M}, s \vDash R(x, b)\end{aligned}$$

And this is the case, since $\mathfrak{M}, s \vDash R(x, b)$ (because $\langle 1, 2\rangle \in R^{\mathfrak{M}}$).

Recall that an x-variant of s is a variable assignment that differs from s at most in what it assigns to x. For every element of $|\mathfrak{M}|$, there is an x-variant of s:

$$s_1 = s[1/x], \qquad s_2 = s[2/x],$$
$$s_3 = s[3/x], \qquad s_4 = s[4/x].$$

So, e.g., $s_2(x) = 2$ and $s_2(y) = s(y) = 1$ for all variables y other than x. These are all the x-variants of s for the structure $\mathfrak{M}$, since $|\mathfrak{M}| = \{1, 2, 3, 4\}$. Note, in particular, that $s_1 = s$ (s is always an x-variant of itself).

To determine if an existentially quantified formula $\exists x\, \varphi(x)$ is satisfied, we have to determine if $\mathfrak{M}, s[m/x] \vDash \varphi(x)$ for at least one $m \in |\mathfrak{M}|$. So,

$$\mathfrak{M}, s \vDash \exists x\, (R(b, x) \vee R(x, b)),$$

since $\mathfrak{M}, s[1/x] \vDash R(b, x) \vee R(x, b)$ ($s[3/x]$ would also fit the bill). But,

$$\mathfrak{M}, s \nvDash \exists x\, (R(b, x) \wedge R(x, b))$$

since, whichever $m \in |\mathfrak{M}|$ we pick, $\mathfrak{M}, s[m/x] \nvDash R(b, x) \wedge R(x, b)$.

To determine if a universally quantified formula $\forall x\, \varphi(x)$ is satisfied, we have to determine if $\mathfrak{M}, s[m/x] \vDash \varphi(x)$ for all $m \in |\mathfrak{M}|$. So,

$$\mathfrak{M}, s \vDash \forall x\, (R(x, a) \to R(a, x)),$$

since $\mathfrak{M}, s[m/x] \vDash R(x, a) \to R(a, x)$ for all $m \in |\mathfrak{M}|$. For $m = 1$, we have $\mathfrak{M}, s[1/x] \vDash R(a, x)$ so the consequent is true; for $m = 2$, 3, and 4, we have $\mathfrak{M}, s[m/x] \nvDash R(x, a)$, so the antecedent is false. But,

$$\mathfrak{M}, s \nvDash \forall x\, (R(a, x) \to R(x, a))$$

since $\mathfrak{M}, s[2/x] \nvDash R(a, x) \to R(x, a)$ (because $\mathfrak{M}, s[2/x] \vDash R(a, x)$ and $\mathfrak{M}, s[2/x] \nvDash R(x, a)$).

For a more complicated case, consider

$$\forall x\, (R(a, x) \to \exists y\, R(x, y)).$$

Since $\mathfrak{M}, s[3/x] \nvDash R(a, x)$ and $\mathfrak{M}, s[4/x] \nvDash R(a, x)$, the interesting cases where we have to worry about the consequent of the conditional are only $m = 1$ and $= 2$. Does $\mathfrak{M}, s[1/x] \vDash \exists y\, R(x, y)$ hold? It does if there is at least one $n \in |\mathfrak{M}|$ so that $\mathfrak{M}, s[1/x][n/y] \vDash R(x, y)$. In fact, if we take $n = 1$, we have $s[1/x][n/y] = s[1/y] = s$. Since $s(x) = 1$, $s(y) = 1$, and $\langle 1, 1\rangle \in R^{\mathfrak{M}}$, the answer is yes.

To determine if $\mathfrak{M}, s[2/x] \vDash \exists y\, R(x, y)$, we have to look at the variable assignments $s[2/x][n/y]$. Here, for $n = 1$, this assignment is $s_2 = s[2/x]$, which does not satisfy $R(x, y)$ ($s_2(x) = 2$, $s_2(y) = 1$, and $\langle 2, 1\rangle \notin R^{\mathfrak{M}}$). However, consider $s[2/x][3/y] = s_2[3/y]$. $\mathfrak{M}, s_2[3/y] \vDash R(x, y)$ since $\langle 2, 3\rangle \in R^{\mathfrak{M}}$, and so $\mathfrak{M}, s_2 \vDash \exists y\, R(x, y)$.

So, for all $n \in |\mathfrak{M}|$, either $\mathfrak{M}, s[m/x] \nvDash R(a,x)$ (if $m = 3, 4$) or $\mathfrak{M}, s[m/x] \vDash \exists y\, R(x,y)$ (if $m = 1, 2$), and so

$$\mathfrak{M}, s \vDash \forall x\, (R(a,x) \to \exists y\, R(x,y)).$$

On the other hand,

$$\mathfrak{M}, s \nvDash \exists x\, (R(a,x) \land \forall y\, R(x,y)).$$

We have $\mathfrak{M}, s[m/x] \vDash R(a,x)$ only for $m = 1$ and $m = 2$. But for both of these values of m, there is in turn an $n \in |\mathfrak{M}|$, namely $n = 4$, so that $\mathfrak{M}, s[m/x][n/y] \nvDash R(x,y)$ and so $\mathfrak{M}, s[m/x] \nvDash \forall y\, R(x,y)$ for $m = 1$ and $m = 2$. In sum, there is no $m \in |\mathfrak{M}|$ such that $\mathfrak{M}, s[m/x] \vDash R(a,x) \land \forall y\, R(x,y)$.

Problem 16.2. Let $\mathcal{L} = \{c, f, A\}$ with one constant symbol, one one-place function symbol and one two-place predicate symbol, and let the structure $\mathfrak{M}$ be given by

1. $|\mathfrak{M}| = \{1, 2, 3\}$
2. $c^{\mathfrak{M}} = 3$
3. $f^{\mathfrak{M}}(1) = 2, f^{\mathfrak{M}}(2) = 3, f^{\mathfrak{M}}(3) = 2$
4. $A^{\mathfrak{M}} = \{\langle 1,2\rangle, \langle 2,3\rangle, \langle 3,3\rangle\}$

(a) Let $s(v) = 1$ for all variables v. Find out whether

$$\mathfrak{M}, s \vDash \exists x\, (A(f(z), c) \to \forall y\, (A(y,x) \lor A(f(y), x)))$$

Explain why or why not.

(b) Give a different structure and variable assignment in which the formula is not satisfied.

---

content/first-order-logic/syntax-and-semantics/assignments.tex

16.5 Variable Assignments

fol:syn:ass:sec

A variable assignment s provides a value for *every* variable—and there are infinitely many of them. This is of course not necessary. We require variable assignments to assign values to all variables simply because it makes things a lot easier. The value of a term t, and whether or not a formula φ is satisfied in a structure with respect to s, only depend on the assignments s makes to the variables in t and the free variables of φ. This is the content of the next two propositions. To make the idea of "depends on" precise, we show that any two variable assignments that agree on all the variables in t give the same value, and that φ is satisfied relative to one iff it is satisfied relative to the other if two variable assignments agree on all free variables of φ.

explanation

Proposition 16.13. *If the variables in a term t are among $x_1, \ldots, x_n$, and $s_1(x_i) = s_2(x_i)$ for $i = 1, \ldots, n$, then $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t) = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t)$.* fol:syn:ass: prop:valindep

Proof. By induction on the complexity of t. For the base case, t can be a constant symbol or one of the variables $x_1, \ldots, x_n$. If $t = c$, then $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t) = c^{\mathfrak{M}} = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t)$. If $t = x_i$, $s_1(x_i) = s_2(x_i)$ by the hypothesis of the proposition, and so $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t) = s_1(x_i) = s_2(x_i) = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t)$.

For the inductive step, assume that $t = f(t_1, \ldots, t_k)$ and that the claim holds for $t_1, \ldots, t_k$. Then

$$\begin{aligned}\mathrm{Val}^{\mathfrak{M}}_{s_1}(t) &= \mathrm{Val}^{\mathfrak{M}}_{s_1}(f(t_1, \ldots, t_k)) \\ &= f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_{s_1}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_k)).\end{aligned}$$

For $j = 1, \ldots, k$, the variables of t_j are among $x_1, \ldots, x_n$. By the induction hypothesis, $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t_j) = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_j)$. So,

$$\begin{aligned}\mathrm{Val}^{\mathfrak{M}}_{s_1}(t) &= \mathrm{Val}^{\mathfrak{M}}_{s_1}(f(t_1, \ldots, t_k)) \\ &= f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_{s_1}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_k)) \\ &= f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_{s_2}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_k)) \\ &= \mathrm{Val}^{\mathfrak{M}}_{s_2}(f(t_1, \ldots, t_k)) = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t).\end{aligned}$$

□

Proposition 16.14. *If the free variables in φ are among $x_1, \ldots, x_n$, and $s_1(x_i) = s_2(x_i)$ for $i = 1, \ldots, n$, then $\mathfrak{M}, s_1 \vDash \varphi$ iff $\mathfrak{M}, s_2 \vDash \varphi$.* fol:syn:ass: prop:satindep

Proof. We use induction on the complexity of φ. For the base case, where φ is atomic, φ can be: $\top$, $\bot$, $R(t_1, \ldots, t_k)$ for a k-place predicate R and terms $t_1, \ldots, t_k$, or $t_1 = t_2$ for terms t_1 and t_2. In the latter two cases, we only demonstrate the forward direction of the biconditional, since the proof of the reverse is symmetrical.

1. $\varphi \equiv \top$: both $\mathfrak{M}, s_1 \vDash \varphi$ and $\mathfrak{M}, s_2 \vDash \varphi$.
2. $\varphi \equiv \bot$: both $\mathfrak{M}, s_1 \nvDash \varphi$ and $\mathfrak{M}, s_2 \nvDash \varphi$.
3. $\varphi \equiv R(t_1, \ldots, t_k)$: let $\mathfrak{M}, s_1 \vDash \varphi$. Then
$$\langle \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_k) \rangle \in R^{\mathfrak{M}}.$$
For $i = 1, \ldots, k$, $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t_i) = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_i)$ by Proposition 16.13. So we also have $\langle \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_i), \ldots, \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_k) \rangle \in R^{\mathfrak{M}}$, and hence $\mathfrak{M}, s_2 \vDash \varphi$.
4. $\varphi \equiv t_1 = t_2$: suppose $\mathfrak{M}, s_1 \vDash \varphi$. Then $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t_1) = \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_2)$. So,
$$\begin{aligned}\mathrm{Val}^{\mathfrak{M}}_{s_2}(t_1) &= \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_1) && \text{(by Proposition 16.13)} \\ &= \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_2) && \text{(since } \mathfrak{M}, s_1 \vDash t_1 = t_2\text{)} \\ &= \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_2) && \text{(by Proposition 16.13),}\end{aligned}$$
so $\mathfrak{M}, s_2 \vDash t_1 = t_2$.

Now assume $\mathfrak{M}, s_1 \vDash \psi$ iff $\mathfrak{M}, s_2 \vDash \psi$ for all formulas ψ less complex than φ. The induction step proceeds by cases determined by the main operator of φ. In each case, we only demonstrate the forward direction of the biconditional; the proof of the reverse direction is symmetrical. In all cases except those for the quantifiers, we apply the induction hypothesis to sub-formulas ψ of φ. The free variables of ψ are among those of φ. Thus, if s_1 and s_2 agree on the free variables of φ, they also agree on those of ψ, and the induction hypothesis applies to ψ.

1. $\varphi \equiv \neg\psi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then $\mathfrak{M}, s_1 \nvDash \psi$, so by the induction hypothesis, $\mathfrak{M}, s_2 \nvDash \psi$, hence $\mathfrak{M}, s_2 \vDash \varphi$.

2. $\varphi \equiv \psi \wedge \chi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then $\mathfrak{M}, s_1 \vDash \psi$ and $\mathfrak{M}, s_1 \vDash \chi$, so by induction hypothesis, $\mathfrak{M}, s_2 \vDash \psi$ and $\mathfrak{M}, s_2 \vDash \chi$. Hence, $\mathfrak{M}, s_2 \vDash \varphi$.

3. $\varphi \equiv \psi \vee \chi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then $\mathfrak{M}, s_1 \vDash \psi$ or $\mathfrak{M}, s_1 \vDash \chi$. By induction hypothesis, $\mathfrak{M}, s_2 \vDash \psi$ or $\mathfrak{M}, s_2 \vDash \chi$, so $\mathfrak{M}, s_2 \vDash \varphi$.

4. $\varphi \equiv \psi \to \chi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then $\mathfrak{M}, s_1 \nvDash \psi$ or $\mathfrak{M}, s_1 \vDash \chi$. By the induction hypothesis, $\mathfrak{M}, s_2 \nvDash \psi$ or $\mathfrak{M}, s_2 \vDash \chi$, so $\mathfrak{M}, s_2 \vDash \varphi$.

5. $\varphi \equiv \psi \leftrightarrow \chi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then either $\mathfrak{M}, s_1 \vDash \psi$ and $\mathfrak{M}, s_1 \vDash \chi$, or $\mathfrak{M}, s_1 \nvDash \psi$ and $\mathfrak{M}, s_1 \nvDash \chi$. By the induction hypothesis, either $\mathfrak{M}, s_2 \vDash \psi$ and $\mathfrak{M}, s_2 \vDash \chi$ or $\mathfrak{M}, s_2 \nvDash \psi$ and $\mathfrak{M}, s_2 \nvDash \chi$. In either case, $\mathfrak{M}, s_2 \vDash \varphi$.

6. $\varphi \equiv \exists x\, \psi$: if $\mathfrak{M}, s_1 \vDash \varphi$, there is an $m \in |\mathfrak{M}|$ so that $\mathfrak{M}, s_1[m/x] \vDash \psi$. Let $s_1' = s_1[m/x]$ and $s_2' = s_2[m/x]$. The free variables of ψ are among x_1, $\ldots$, x_n, and x. $s_1'(x_i) = s_2'(x_i)$, since s_1' and s_2' are x-variants of s_1 and s_2, respectively, and by hypothesis $s_1(x_i) = s_2(x_i)$. $s_1'(x) = s_2'(x) = m$ by the way we have defined s_1' and s_2'. Then the induction hypothesis applies to ψ and s_1', s_2', so $\mathfrak{M}, s_2' \vDash \psi$. Hence, since $s_2' = s_2[m/x]$, there is an $m \in |\mathfrak{M}|$ such that $\mathfrak{M}, s_2[m/x] \vDash \psi$, and so $\mathfrak{M}, s_2 \vDash \varphi$.

7. $\varphi \equiv \forall x\, \psi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then for every $m \in |\mathfrak{M}|$, $\mathfrak{M}, s_1[m/x] \vDash \psi$. We want to show that also, for every $m \in |\mathfrak{M}|$, $\mathfrak{M}, s_2[m/x] \vDash \psi$. So let $m \in |\mathfrak{M}|$ be arbitrary, and consider $s_1' = s[m/x]$ and $s_2' = s[m/x]$. We have that $\mathfrak{M}, s_1' \vDash \psi$. The free variables of ψ are among x_1, $\ldots$, x_n, and x. $s_1'(x_i) = s_2'(x_i)$, since s_1' and s_2' are x-variants of s_1 and s_2, respectively, and by hypothesis $s_1(x_i) = s_2(x_i)$. $s_1'(x) = s_2'(x) = m$ by the way we have defined s_1' and s_2'. Then the induction hypothesis applies to ψ and s_1', s_2', and we have $\mathfrak{M}, s_2' \vDash \psi$. This applies to every $m \in |\mathfrak{M}|$, i.e., $\mathfrak{M}, s_2[m/x] \vDash \psi$ for all $m \in |\mathfrak{M}|$, so $\mathfrak{M}, s_2 \vDash \varphi$.

By induction, we get that $\mathfrak{M}, s_1 \vDash \varphi$ iff $\mathfrak{M}, s_2 \vDash \varphi$ whenever the free variables in φ are among $x_1, \ldots, x_n$ and $s_1(x_i) = s_2(x_i)$ for $i = 1, \ldots, n$. □

Problem 16.3. Complete the proof of Proposition 16.14.

explanation

Sentences have no free variables, so any two variable assignments assign the same things to all the (zero) free variables of any sentence. The proposition just proved then means that whether or not a sentence is satisfied in a structure relative to a variable assignment is completely independent of the assignment. We'll record this fact. It justifies the definition of satisfaction of a sentence in a structure (without mentioning a variable assignment) that follows.

Corollary 16.15. *If φ is a sentence and s a variable assignment, then $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s' \vDash \varphi$ for every variable assignment s'.* fol:syn:ass: cor:sat-sentence

Proof. Let s' be any variable assignment. Since φ is a sentence, it has no free variables, and so every variable assignment s' trivially assigns the same things to all free variables of φ as does s. So the condition of Proposition 16.14 is satisfied, and we have $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s' \vDash \varphi$. □

Definition 16.16. If φ is a sentence, we say that a structure $\mathfrak{M}$ *satisfies* φ, $\mathfrak{M} \vDash \varphi$, iff $\mathfrak{M}, s \vDash \varphi$ for all variable assignments s. fol:syn:ass: defn:satisfaction

If $\mathfrak{M} \vDash \varphi$, we also simply say that φ *is true in* $\mathfrak{M}$. The notion of satisfaction naturally extends from individual sentences to sets of sentences.

Definition 16.17. If Γ is a set of sentences Γ, we say that a structure $\mathfrak{M}$ *satisfies* Γ, $\mathfrak{M} \vDash \Gamma$, iff $\mathfrak{M} \vDash \varphi$ for all $\varphi \in \Gamma$. fol:syn:ass: defn:sat

Proposition 16.18. *Let $\mathfrak{M}$ be a structure, φ be a sentence, and s a variable assignment. $\mathfrak{M} \vDash \varphi$ iff $\mathfrak{M}, s \vDash \varphi$.* fol:syn:ass: prop:sentence-sat-true

Proof. Exercise. □

Problem 16.4. Prove Proposition 16.18

Proposition 16.19. *Suppose $\varphi(x)$ only contains x free, and $\mathfrak{M}$ is a structure. Then:* fol:syn:ass: prop:sat-quant

1. *$\mathfrak{M} \vDash \exists x\, \varphi(x)$ iff $\mathfrak{M}, s \vDash \varphi(x)$ for at least one variable assignment s.*
2. *$\mathfrak{M} \vDash \forall x\, \varphi(x)$ iff $\mathfrak{M}, s \vDash \varphi(x)$ for all variable assignments s.*

Proof. Exercise. □

Problem 16.5. Prove Proposition 16.19.

Problem 16.6. Suppose $\mathcal{L}$ is a language without function symbols. Given a structure $\mathfrak{M}$, c a constant symbol and $a \in |\mathfrak{M}|$, define $\mathfrak{M}[a/c]$ to be the structure that is just like $\mathfrak{M}$, except that $c^{\mathfrak{M}[a/c]} = a$. Define $\mathfrak{M} \mid\!\models \varphi$ for sentences φ by:

1. $\varphi \equiv \bot$: not $\mathfrak{M} \mid\!\models \varphi$.
2. $\varphi \equiv \top$: $\mathfrak{M} \mid\!\models \varphi$.

3. $\varphi \equiv R(d_1, \ldots, d_n)$: $\mathfrak{M} \mid\!\models \varphi$ iff $\langle d_1^{\mathfrak{M}}, \ldots, d_n^{\mathfrak{M}} \rangle \in R^{\mathfrak{M}}$.
4. $\varphi \equiv d_1 = d_2$: $\mathfrak{M} \mid\!\models \varphi$ iff $d_1^{\mathfrak{M}} = d_2^{\mathfrak{M}}$.
5. $\varphi \equiv \neg\psi$: $\mathfrak{M} \mid\!\models \varphi$ iff not $\mathfrak{M} \mid\!\models \psi$.
6. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M} \mid\!\models \varphi$ iff $\mathfrak{M} \mid\!\models \psi$ and $\mathfrak{M} \mid\!\models \chi$.
7. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M} \mid\!\models \varphi$ iff $\mathfrak{M} \mid\!\models \psi$ or $\mathfrak{M} \mid\!\models \chi$ (or both).
8. $\varphi \equiv (\psi \to \chi)$: $\mathfrak{M} \mid\!\models \varphi$ iff not $\mathfrak{M} \mid\!\models \psi$ or $\mathfrak{M} \mid\!\models \chi$ (or both).
9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M} \mid\!\models \varphi$ iff either both $\mathfrak{M} \mid\!\models \psi$ and $\mathfrak{M} \mid\!\models \chi$, or neither $\mathfrak{M} \mid\!\models \psi$ nor $\mathfrak{M} \mid\!\models \chi$.
10. $\varphi \equiv \forall x\, \psi$: $\mathfrak{M} \mid\!\models \varphi$ iff for all $a \in |\mathfrak{M}|$, $\mathfrak{M}[a/c] \mid\!\models \psi[c/x]$, if c does not occur in ψ.
11. $\varphi \equiv \exists x\, \psi$: $\mathfrak{M} \mid\!\models \varphi$ iff there is an $a \in |\mathfrak{M}|$ such that $\mathfrak{M}[a/c] \mid\!\models \psi[c/x]$, if c does not occur in ψ.

Let $x_1, \ldots, x_n$ be all free variables in φ, $c_1, \ldots, c_n$ constant symbols not in φ, $a_1, \ldots, a_n \in |\mathfrak{M}|$, and $s(x_i) = a_i$.

Show that $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}[a_1/c_1, \ldots, a_n/c_n] \mid\!\models \varphi[c_1/x_1] \ldots [c_n/x_n]$.

(This problem shows that it is possible to give a semantics for first-order logic that makes do without variable assignments.)

Problem 16.7. Suppose that f is a function symbol not in $\varphi(x, y)$. Show that there is a structure $\mathfrak{M}$ such that $\mathfrak{M} \vDash \forall x\, \exists y\, \varphi(x, y)$ iff there is an $\mathfrak{M}'$ such that $\mathfrak{M}' \vDash \forall x\, \varphi(x, f(x))$.

(This problem is a special case of what's known as Skolem's Theorem; $\forall x\, \varphi(x, f(x))$ is called a *Skolem normal form* of $\forall x\, \exists y\, \varphi(x, y)$.)

---

content/first-order-logic/syntax-and-semantics/extensionality.tex

16.6 Extensionality

fol:syn:ext:sec

Extensionality, sometimes called relevance, can be expressed informally as follows: the only factors that bear upon the satisfaction of formula φ in a structure $\mathfrak{M}$ relative to a variable assignment s, are the size of the domain and the assignments made by $\mathfrak{M}$ and s to the elements of the language that actually appear in φ. explanation

One immediate consequence of extensionality is that where two structures $\mathfrak{M}$ and $\mathfrak{M}'$ agree on all the elements of the language appearing in a sentence φ and have the same domain, $\mathfrak{M}$ and $\mathfrak{M}'$ must also agree on whether or not φ itself is true.

Proposition 16.20 (Extensionality). *Let φ be a formula, and $\mathfrak{M}_1$ and $\mathfrak{M}_2$ be structures with $|\mathfrak{M}_1| = |\mathfrak{M}_2|$, and s a variable assignment on $|\mathfrak{M}_1| = |\mathfrak{M}_2|$. If $c^{\mathfrak{M}_1} = c^{\mathfrak{M}_2}$, $R^{\mathfrak{M}_1} = R^{\mathfrak{M}_2}$, and $f^{\mathfrak{M}_1} = f^{\mathfrak{M}_2}$ for every constant symbol c, relation symbol R, and function symbol f occurring in φ, then $\mathfrak{M}_1, s \vDash \varphi$ iff $\mathfrak{M}_2, s \vDash \varphi$.* fol:syn:ext: prop:extensionality

Proof. First prove (by induction on t) that for every term, $\mathrm{Val}^{\mathfrak{M}_1}_s(t) = \mathrm{Val}^{\mathfrak{M}_2}_s(t)$. Then prove the proposition by induction on φ, making use of the claim just proved for the induction basis (where φ is atomic). □

Problem 16.8. Carry out the proof of Proposition 16.20 in detail.

Corollary 16.21 (Extensionality for Sentences). *Let φ be a sentence and $\mathfrak{M}_1$, $\mathfrak{M}_2$ as in Proposition 16.20. Then $\mathfrak{M}_1 \vDash \varphi$ iff $\mathfrak{M}_2 \vDash \varphi$.* fol:syn:ext: cor:extensionality-sent

Proof. Follows from Proposition 16.20 by Corollary 16.15. □

Moreover, the value of a term, and whether or not a structure satisfies a formula, only depend on the values of its subterms.

Proposition 16.22. *Let $\mathfrak{M}$ be a structure, t and t' terms, and s a variable assignment. Then $\mathrm{Val}^{\mathfrak{M}}_s(t[t'/x]) = \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(t)$.* fol:syn:ext: prop:ext-terms

Proof. By induction on t.

1. If t is a constant, say, $t \equiv c$, then $t[t'/x] = c$, and $\mathrm{Val}^{\mathfrak{M}}_s(c) = c^{\mathfrak{M}} = \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(c)$.

2. If t is a variable other than x, say, $t \equiv y$, then $t[t'/x] = y$, and $\mathrm{Val}^{\mathfrak{M}}_s(y) = \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(y)$ since $s \sim_x s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]$.

3. If $t \equiv x$, then $t[t'/x] = t'$. But $\mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(x) = \mathrm{Val}^{\mathfrak{M}}_s(t')$ by definition of $s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]$.

4. If $t \equiv f(t_1, \ldots, t_n)$ then we have:

$$\begin{aligned}
&\mathrm{Val}^{\mathfrak{M}}_s(t[t'/x]) = \\
&\quad = \mathrm{Val}^{\mathfrak{M}}_s(f(t_1[t'/x], \ldots, t_n[t'/x])) \\
&\qquad \text{by definition of } t[t'/x] \\
&\quad = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(t_1[t'/x]), \ldots, \mathrm{Val}^{\mathfrak{M}}_s(t_n[t'/x])) \\
&\qquad \text{by definition of } \mathrm{Val}^{\mathfrak{M}}_s(f(\ldots)) \\
&\quad = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(t_n)) \\
&\qquad \text{by induction hypothesis} \\
&\quad = \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(t) \text{ by definition of } \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(f(\ldots))
\end{aligned}$$ □

fol:syn:ext: prop:ext-formulas **Proposition 16.23.** *Let $\mathfrak{M}$ be a structure, φ a formula, t' a term, and s a variable assignment. Then $\mathfrak{M}, s \vDash \varphi[t'/x]$ iff $\mathfrak{M}, s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x] \vDash \varphi$.*

Proof. Exercise. □

Problem 16.9. Prove Proposition 16.23

The point of Propositions 16.22 and 16.23 is the following. Suppose we have a term t or a formula φ and some term t', and we want to know the value of $t[t'/x]$ or whether or not $\varphi[t'/x]$ is satisfied in a structure $\mathfrak{M}$ relative to a variable assignment s. Then we can either perform the substitution first and then consider the value or satisfaction relative to $\mathfrak{M}$ and s, or we can first determine the value $m = \mathrm{Val}^{\mathfrak{M}}_s(t')$ of t' in $\mathfrak{M}$ relative to s, change the variable assignment to $s[m/x]$ and then consider the value of t in $\mathfrak{M}$ and $s[m/x]$, or whether $\mathfrak{M}, s[m/x] \vDash \varphi$. Propositions 16.22 and 16.23 guarantee that the answer will be the same, whichever way we do it. explanation

---

content/first-order-logic/syntax-and-semantics/semantic-notions.tex

16.7 Semantic Notions

fol:syn:sem: sec Given the definition of structures for first-order languages, we can define some basic semantic properties of and relationships between sentences. The simplest of these is the notion of *validity* of a sentence. A sentence is valid if it is satisfied in every structure. Valid sentences are those that are satisfied regardless of how the non-logical symbols in it are interpreted. Valid sentences are therefore also called *logical truths*—they are true, i.e., satisfied, in any structure and hence their truth depends only on the logical symbols occurring in them and their syntactic structure, but not on the non-logical symbols or their interpretation. explanation

Definition 16.24 (Validity). A sentence φ is *valid*, $\vDash \varphi$, iff $\mathfrak{M} \vDash \varphi$ for every structure $\mathfrak{M}$.

Definition 16.25 (Entailment). A set of sentences Γ *entails* a sentence φ, $\Gamma \vDash \varphi$, iff for every structure $\mathfrak{M}$ with $\mathfrak{M} \vDash \Gamma$, $\mathfrak{M} \vDash \varphi$.

Definition 16.26 (Satisfiability). A set of sentences Γ is *satisfiable* if $\mathfrak{M} \vDash \Gamma$ for some structure $\mathfrak{M}$. If Γ is not satisfiable it is called *unsatisfiable*.

Proposition 16.27. *A sentence φ is valid iff $\Gamma \vDash \varphi$ for every set of sentences Γ.*

Proof. For the forward direction, let φ be valid, and let Γ be a set of sentences. Let $\mathfrak{M}$ be a structure so that $\mathfrak{M} \vDash \Gamma$. Since φ is valid, $\mathfrak{M} \vDash \varphi$, hence $\Gamma \vDash \varphi$.

For the contrapositive of the reverse direction, let φ be invalid, so there is a structure $\mathfrak{M}$ with $\mathfrak{M} \nvDash \varphi$. When $\Gamma = \{\top\}$, since $\top$ is valid, $\mathfrak{M} \vDash \Gamma$. Hence, there is a structure $\mathfrak{M}$ so that $\mathfrak{M} \vDash \Gamma$ but $\mathfrak{M} \nvDash \varphi$, hence Γ does not entail φ. □

Proposition 16.28. *$\Gamma \vDash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable.*

fol:syn:sem: prop:entails-unsat

Proof. For the forward direction, suppose $\Gamma \vDash \varphi$ and suppose to the contrary that there is a structure $\mathfrak{M}$ so that $\mathfrak{M} \vDash \Gamma \cup \{\neg\varphi\}$. Since $\mathfrak{M} \vDash \Gamma$ and $\Gamma \vDash \varphi$, $\mathfrak{M} \vDash \varphi$. Also, since $\mathfrak{M} \vDash \Gamma \cup \{\neg\varphi\}$, $\mathfrak{M} \vDash \neg\varphi$, so we have both $\mathfrak{M} \vDash \varphi$ and $\mathfrak{M} \nvDash \varphi$, a contradiction. Hence, there can be no such structure $\mathfrak{M}$, so $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable.

For the reverse direction, suppose $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable. So for every structure $\mathfrak{M}$, either $\mathfrak{M} \nvDash \Gamma$ or $\mathfrak{M} \vDash \varphi$. Hence, for every structure $\mathfrak{M}$ with $\mathfrak{M} \vDash \Gamma$, $\mathfrak{M} \vDash \varphi$, so $\Gamma \vDash \varphi$. □

Problem 16.10. 1. Show that $\Gamma \vDash \bot$ iff Γ is unsatisfiable.

2. Show that $\Gamma \cup \{\varphi\} \vDash \bot$ iff $\Gamma \vDash \neg\varphi$.

3. Suppose c does not occur in φ or Γ. Show that $\Gamma \vDash \forall x\, \varphi$ iff $\Gamma \vDash \varphi[c/x]$.

Proposition 16.29. *If $\Gamma \subseteq \Gamma'$ and $\Gamma \vDash \varphi$, then $\Gamma' \vDash \varphi$.*

Proof. Suppose that $\Gamma \subseteq \Gamma'$ and $\Gamma \vDash \varphi$. Let $\mathfrak{M}$ be a structure such that $\mathfrak{M} \vDash \Gamma'$; then $\mathfrak{M} \vDash \Gamma$, and since $\Gamma \vDash \varphi$, we get that $\mathfrak{M} \vDash \varphi$. Hence, whenever $\mathfrak{M} \vDash \Gamma'$, $\mathfrak{M} \vDash \varphi$, so $\Gamma' \vDash \varphi$. □

Theorem 16.30 (Semantic Deduction Theorem). *$\Gamma \cup \{\varphi\} \vDash \psi$ iff $\Gamma \vDash \varphi \to \psi$.*

fol:syn:sem: thm:sem-deduction

Proof. For the forward direction, let $\Gamma \cup \{\varphi\} \vDash \psi$ and let $\mathfrak{M}$ be a structure so that $\mathfrak{M} \vDash \Gamma$. If $\mathfrak{M} \vDash \varphi$, then $\mathfrak{M} \vDash \Gamma \cup \{\varphi\}$, so since $\Gamma \cup \{\varphi\}$ entails ψ, we get $\mathfrak{M} \vDash \psi$. Therefore, $\mathfrak{M} \vDash \varphi \to \psi$, so $\Gamma \vDash \varphi \to \psi$.

For the reverse direction, let $\Gamma \vDash \varphi \to \psi$ and $\mathfrak{M}$ be a structure so that $\mathfrak{M} \vDash \Gamma \cup \{\varphi\}$. Then $\mathfrak{M} \vDash \Gamma$, so $\mathfrak{M} \vDash \varphi \to \psi$, and since $\mathfrak{M} \vDash \varphi$, $\mathfrak{M} \vDash \psi$. Hence, whenever $\mathfrak{M} \vDash \Gamma \cup \{\varphi\}$, $\mathfrak{M} \vDash \psi$, so $\Gamma \cup \{\varphi\} \vDash \psi$. □

Proposition 16.31. *Let $\mathfrak{M}$ be a structure, and $\varphi(x)$ a formula with one free variable x, and t a closed term. Then:*

fol:syn:sem: prop:quant-terms

1. *$\varphi(t) \vDash \exists x\, \varphi(x)$*
2. *$\forall x\, \varphi(x) \vDash \varphi(t)$*

Proof. 1. Suppose $\mathfrak{M} \vDash \varphi(t)$. Let s be a variable assignment with $s(x) = \mathrm{Val}^{\mathfrak{M}}(t)$. Then $\mathfrak{M}, s \vDash \varphi(t)$ since $\varphi(t)$ is a sentence. By Proposition 16.23, $\mathfrak{M}, s \vDash \varphi(x)$. By Proposition 16.19, $\mathfrak{M} \vDash \exists x\, \varphi(x)$.

2. Suppose $\mathfrak{M} \vDash \forall x\, \varphi(x)$. Let s be a variable assignment with $s(x) = \mathrm{Val}^{\mathfrak{M}}(t)$. By Proposition 16.19, $\mathfrak{M}, s \vDash \varphi(x)$. By Proposition 16.23, $\mathfrak{M}, s \vDash \varphi(t)$. By Proposition 16.18, $\mathfrak{M} \vDash \varphi(t)$ since $\varphi(t)$ is a sentence. □

Problem 16.11. Complete the proof of Proposition 16.31.

Chapter 17

Theories and Their Models

---

content/first-order-logic/models-theories/introduction.tex

17.1 Introduction

fol:mat:int:sec

explanation

The development of the axiomatic method is a significant achievement in the history of science, and is of special importance in the history of mathematics. An axiomatic development of a field involves the clarification of many questions: What is the field about? What are the most fundamental concepts? How are they related? Can all the concepts of the field be defined in terms of these fundamental concepts? What laws do, and must, these concepts obey?

The axiomatic method and logic were made for each other. Formal logic provides the tools for formulating axiomatic theories, for proving theorems from the axioms of the theory in a precisely specified way, for studying the properties of all systems satisfying the axioms in a systematic way.

Definition 17.1. A set of sentences Γ is *closed* iff, whenever $\Gamma \vDash \varphi$ then $\varphi \in \Gamma$. The *closure* of a set of sentences Γ is $\{\varphi : \Gamma \vDash \varphi\}$.

We say that Γ is *axiomatized by* a set of sentences Δ if Γ is the closure of Δ.

explanation

We can think of an axiomatic theory as the set of sentences that is axiomatized by its set of axioms Δ. In other words, when we have a first-order language which contains non-logical symbols for the primitives of the axiomatically developed science we wish to study, together with a set of sentences that express the fundamental laws of the science, we can think of the theory as represented by all the sentences in this language that are entailed by the axioms. This ranges from simple examples with only a single primitive and simple axioms, such as the theory of partial orders, to complex theories such as Newtonian mechanics.

The important logical facts that make this formal approach to the axiomatic method so important are the following. Suppose Γ is an axiom system for a theory, i.e., a set of sentences.

1. We can state precisely when an axiom system captures an intended class of structures. That is, if we are interested in a certain class of structures, we will successfully capture that class by an axiom system Γ iff the structures are exactly those $\mathfrak{M}$ such that $\mathfrak{M} \models \Gamma$.

2. We may fail in this respect because there are $\mathfrak{M}$ such that $\mathfrak{M} \models \Gamma$, but $\mathfrak{M}$ is not one of the structures we intend. This may lead us to add axioms which are not true in $\mathfrak{M}$.

3. If we are successful at least in the respect that Γ is true in all the intended structures, then a sentence φ is true in all intended structures whenever $\Gamma \models \varphi$. Thus we can use logical tools (such as derivation methods) to show that sentences are true in all intended structures simply by showing that they are entailed by the axioms.

4. Sometimes we don't have intended structures in mind, but instead start from the axioms themselves: we begin with some primitives that we want to satisfy certain laws which we codify in an axiom system. One thing that we would like to verify right away is that the axioms do not contradict each other: if they do, there can be no concepts that obey these laws, and we have tried to set up an incoherent theory. We can verify that this doesn't happen by finding a model of Γ. And if there are models of our theory, we can use logical methods to investigate them, and we can also use logical methods to construct models.

5. The independence of the axioms is likewise an important question. It may happen that one of the axioms is actually a consequence of the others, and so is redundant. We can prove that an axiom φ in Γ is redundant by proving $\Gamma \setminus \{\varphi\} \models \varphi$. We can also prove that an axiom is not redundant by showing that $(\Gamma \setminus \{\varphi\}) \cup \{\neg\varphi\}$ is satisfiable. For instance, this is how it was shown that the parallel postulate is independent of the other axioms of geometry.

6. Another important question is that of definability of concepts in a theory: The choice of the language determines what the models of a theory consist of. But not every aspect of a theory must be represented separately in its models. For instance, every ordering $\leq$ determines a corresponding strict ordering $<$—given one, we can define the other. So it is not necessary that a model of a theory involving such an order must *also* contain the corresponding strict ordering. When is it the case, in general, that one relation can be defined in terms of others? When is it impossible to define a relation in terms of others (and hence must add it to the primitives of the language)?

content/first-order-logic/models-theories/expressing-props-of-structures.tex

17.2 Expressing Properties of Structures

fol:mat:exs:sec

It is often useful and important to express conditions on functions and relations, or more generally, that the functions and relations in a structure satisfy these conditions. For instance, we would like to have ways of distinguishing those structures for a language which "capture" what we want the predicate symbols to "mean" from those that do not. Of course we're completely free to specify which structures we "intend," e.g., we can specify that the interpretation of the predicate symbol $\leq$ must be an ordering, or that we are only interested in interpretations of $\mathcal{L}$ in which the domain consists of sets and $\in$ is interpreted by the "is an element of" relation. But can we do this with sentences of the language? In other words, which conditions on a structure $\mathfrak{M}$ can we express by a sentence (or perhaps a set of sentences) in the language of $\mathfrak{M}$? There are some conditions that we will not be able to express. For instance, there is no sentence of $\mathcal{L}_A$ which is only true in a structure $\mathfrak{M}$ if $|\mathfrak{M}| = \mathbb{N}$. We cannot express "the domain contains only natural numbers." But there are "structural properties" of structures that we perhaps can express. Which properties of structures can we express by sentences? Or, to put it another way, which collections of structures can we describe as those making a sentence (or set of sentences) true?

explanation

Definition 17.2 (Model of a set). Let Γ be a set of sentences in a language $\mathcal{L}$. We say that a structure $\mathfrak{M}$ *is a model of* Γ if $\mathfrak{M} \vDash \varphi$ for all $\varphi \in \Gamma$.

Example 17.3. The sentence $\forall x\, x \leq x$ is true in $\mathfrak{M}$ iff $\leq^{\mathfrak{M}}$ is a reflexive relation. The sentence $\forall x \forall y\, ((x \leq y \wedge y \leq x) \to x = y)$ is true in $\mathfrak{M}$ iff $\leq^{\mathfrak{M}}$ is anti-symmetric. The sentence $\forall x \forall y \forall z\, ((x \leq y \wedge y \leq z) \to x \leq z)$ is true in $\mathfrak{M}$ iff $\leq^{\mathfrak{M}}$ is transitive. Thus, the models of

$$\begin{aligned} \{ \quad & \forall x\, x \leq x, \\ & \forall x \forall y\, ((x \leq y \wedge y \leq x) \to x = y), \\ & \forall x \forall y \forall z\, ((x \leq y \wedge y \leq z) \to x \leq z) \quad \} \end{aligned}$$

are exactly those structures in which $\leq^{\mathfrak{M}}$ is reflexive, anti-symmetric, and transitive, i.e., a partial order. Hence, we can take them as axioms for the *first-order theory of partial orders*.

content/first-order-logic/models-theories/theories.tex

17.3 Examples of First-Order Theories

fol:mat:the:sec

Example 17.4. The theory of strict linear orders in the language $\mathcal{L}_<$ is axiomatized by the set

$$\begin{aligned}\{\quad & \forall x \neg x < x,\\ & \forall x \forall y \left((x < y \vee y < x) \vee x = y\right),\\ & \forall x \forall y \forall z \left((x < y \wedge y < z) \rightarrow x < z\right) \quad \}\end{aligned}$$

It completely captures the intended structures: every strict linear order is a model of this axiom system, and vice versa, if R is a linear order on a set X, then the structure $\mathfrak{M}$ with $|\mathfrak{M}| = X$ and $<^{\mathfrak{M}} = R$ is a model of this theory.

Example 17.5. The theory of groups in the language $\mathtt{1}$ (constant symbol), $\cdot$ (two-place function symbol) is axiomatized by

$$\begin{aligned}&\forall x \left(x \cdot \mathtt{1}\right) = x\\ &\forall x \forall y \forall z \left(x \cdot (y \cdot z)\right) = ((x \cdot y) \cdot z)\\ &\forall x \exists y \left(x \cdot y\right) = \mathtt{1}\end{aligned}$$

Example 17.6. The theory of Peano arithmetic is axiomatized by the following sentences in the language of arithmetic $\mathcal{L}_A$.

$$\begin{aligned}&\forall x \forall y \left(x' = y' \rightarrow x = y\right)\\ &\forall x\, \mathtt{0} \neq x'\\ &\forall x \left(x + \mathtt{0}\right) = x\\ &\forall x \forall y \left(x + y'\right) = \left(x + y\right)'\\ &\forall x \left(x \times \mathtt{0}\right) = \mathtt{0}\\ &\forall x \forall y \left(x \times y'\right) = \left(\left(x \times y\right) + x\right)\\ &\forall x \forall y \left(x < y \leftrightarrow \exists z \left(z' + x\right) = y\right)\end{aligned}$$

plus all sentences of the form

$$\left(\varphi(\mathtt{0}) \wedge \forall x \left(\varphi(x) \rightarrow \varphi(x')\right)\right) \rightarrow \forall x\, \varphi(x)$$

Since there are infinitely many sentences of the latter form, this axiom system is infinite. The latter form is called the *induction schema*. (Actually, the induction schema is a bit more complicated than we let on here.)

The last axiom is an *explicit definition* of $<$.

Example 17.7. The theory of pure sets plays an important role in the foundations (and in the philosophy) of mathematics. A set is pure if all its elements are also pure sets. The empty set counts therefore as pure, but a set that has something as an element that is not a set would not be pure. So the pure sets are those that are formed just from the empty set and no "urelements," i.e., objects that are not themselves sets.

The following might be considered as an axiom system for a theory of pure sets:

$$\exists x\, \neg \exists y\, y \in x$$
$$\forall x\, \forall y\, (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y)$$
$$\forall x\, \forall y\, \exists z\, \forall u\, (u \in z \leftrightarrow (u = x \vee u = y))$$
$$\forall x\, \exists y\, \forall z\, (z \in y \leftrightarrow \exists u\, (z \in u \wedge u \in x))$$

plus all sentences of the form

$$\exists x\, \forall y\, (y \in x \leftrightarrow \varphi(y))$$

The first axiom says that there is a set with no elements (i.e., $\emptyset$ exists); the second says that sets are extensional; the third that for any sets X and Y, the set $\{X, Y\}$ exists; the fourth that for any set X, the set $\cup X$ exists, where $\cup X$ is the union of all the elements of X.

The sentences mentioned last are collectively called the *naive comprehension scheme*. It essentially says that for every $\varphi(x)$, the set $\{x : \varphi(x)\}$ exists—so at first glance a true, useful, and perhaps even necessary axiom. It is called "naive" because, as it turns out, it makes this theory unsatisfiable: if you take $\varphi(y)$ to be $\neg y \in y$, you get the sentence

$$\exists x\, \forall y\, (y \in x \leftrightarrow \neg y \in y)$$

and this sentence is not satisfied in any structure.

Example 17.8. In the area of *mereology*, the relation of *parthood* is a fundamental relation. Just like theories of sets, there are theories of parthood that axiomatize various conceptions (sometimes conflicting) of this relation.

The language of mereology contains a single two-place predicate symbol P, and $P(x, y)$ "means" that x is a part of y. When we have this interpretation in mind, a structure for this language is called a *parthood structure*. Of course, not every structure for a single two-place predicate will really deserve this name. To have a chance of capturing "parthood," $P^{\mathfrak{M}}$ must satisfy some conditions, which we can lay down as axioms for a theory of parthood. For instance, parthood is a partial order on objects: every object is a part (albeit an *improper* part) of itself; no two different objects can be parts of each other; a part of a part of an object is itself part of that object. Note that in this sense "is a part of" resembles "is a subset of," but does not resemble "is an element of" which is neither reflexive nor transitive.

$$\forall x\, P(x, x)$$
$$\forall x\, \forall y\, ((P(x, y) \wedge P(y, x)) \rightarrow x = y)$$
$$\forall x\, \forall y\, \forall z\, ((P(x, y) \wedge P(y, z)) \rightarrow P(x, z))$$

Moreover, any two objects have a mereological sum (an object that has these two objects as parts, and is minimal in this respect).

$$\forall x \, \forall y \, \exists z \, \forall u \, (P(z, u) \leftrightarrow (P(x, u) \land P(y, u)))$$

These are only some of the basic principles of parthood considered by metaphysicians. Further principles, however, quickly become hard to formulate or write down without first introducing some defined relations. For instance, most metaphysicians interested in mereology also view the following as a valid principle: whenever an object x has a proper part y, it also has a part z that has no parts in common with y, and so that the fusion of y and z is x.

---

content/first-order-logic/models-theories/expressing-relations.tex

17.4 Expressing Relations in a Structure

explanation

One main use formulas can be put to is to express properties and relations in a structure $\mathfrak{M}$ in terms of the primitives of the language $\mathcal{L}$ of $\mathfrak{M}$. By this we mean the following: the domain of $\mathfrak{M}$ is a set of objects. The constant symbols, function symbols, and predicate symbols are interpreted in $\mathfrak{M}$ by some objects in $|\mathfrak{M}|$, functions on $|\mathfrak{M}|$, and relations on $|\mathfrak{M}|$. For instance, if A^2_0 is in $\mathcal{L}$, then $\mathfrak{M}$ assigns to it a relation $R = A^{2\mathfrak{M}}_0$. Then the formula $A^2_0(v_1, v_2)$ *expresses* that very relation, in the following sense: if a variable assignment s maps v_1 to $a \in |\mathfrak{M}|$ and v_2 to $b \in |\mathfrak{M}|$, then

fol:mat:exr:sec

$$Rab \quad \text{iff} \quad \mathfrak{M}, s \vDash A^2_0(v_1, v_2).$$

Note that we have to involve variable assignments here: we can't just say "Rab iff $\mathfrak{M} \vDash A^2_0(a, b)$" because a and b are not symbols of our language: they are elements of $|\mathfrak{M}|$.

Since we don't just have atomic formulas, but can combine them using the logical connectives and the quantifiers, more complex formulas can define other relations which aren't directly built into $\mathfrak{M}$. We're interested in how to do that, and specifically, which relations we can define in a structure.

Definition 17.9. Let $\varphi(v_1, \ldots, v_n)$ be a formula of $\mathcal{L}$ in which only $v_1, \ldots, v_n$ occur free, and let $\mathfrak{M}$ be a structure for $\mathcal{L}$. $\varphi(v_1, \ldots, v_n)$ *expresses the relation* $R \subseteq |\mathfrak{M}|^n$ iff

$$Ra_1 \ldots a_n \quad \text{iff} \quad \mathfrak{M}, s \vDash \varphi(v_1, \ldots, v_n)$$

for any variable assignment s with $s(v_i) = a_i$ $(i = 1, \ldots, n)$.

Example 17.10. In the standard model of arithmetic $\mathfrak{N}$, the formula $v_1 < v_2 \lor v_1 = v_2$ expresses the $\leq$ relation on $\mathbb{N}$. The formula $v_2 = v_1'$ expresses the successor relation, i.e., the relation $R \subseteq \mathbb{N}^2$ where Rnm holds if m is the

successor of n. The formula $v_1 = v_2'$ expresses the predecessor relation. The formulas $\exists v_3\,(v_3 \neq 0 \wedge v_2 = (v_1 + v_3))$ and $\exists v_3\,(v_1 + v_3') = v_2$ both express the $<$ relation. This means that the predicate symbol $<$ is actually superfluous in the language of arithmetic; it can be defined.

This idea is not just interesting in specific structures, but generally whenever we use a language to describe an intended model or models, i.e., when we consider theories. These theories often only contain a few predicate symbols as basic symbols, but in the domain they are used to describe often many other relations play an important role. If these other relations can be systematically expressed by the relations that interpret the basic predicate symbols of the language, we say we can *define* them in the language. explanation

Problem 17.1. Find formulas in $\mathcal{L}_A$ which define the following relations:

1. n is between i and j;
2. n evenly divides m (i.e., m is a multiple of n);
3. n is a prime number (i.e., no number other than 1 and n evenly divides n).

Problem 17.2. Suppose the formula $\varphi(v_1, v_2)$ expresses the relation $R \subseteq |\mathfrak{M}|^2$ in a structure $\mathfrak{M}$. Find formulas that express the following relations:

1. the inverse R^{-1} of R;
2. the relative product $R \mid R$;

Can you find a way to express R^+, the transitive closure of R?

Problem 17.3. Let $\mathcal{L}$ be the language containing a 2-place predicate symbol $<$ only (no other constant symbols, function symbols or predicate symbols—except of course $=$). Let $\mathfrak{N}$ be the structure such that $|\mathfrak{N}| = \mathbb{N}$, and $<^{\mathfrak{N}} = \{\langle n, m\rangle : n < m\}$. Prove the following:

1. $\{0\}$ is definable in $\mathfrak{N}$;
2. $\{1\}$ is definable in $\mathfrak{N}$;
3. $\{2\}$ is definable in $\mathfrak{N}$;
4. for each $n \in \mathbb{N}$, the set $\{n\}$ is definable in $\mathfrak{N}$;
5. every finite subset of $|\mathfrak{N}|$ is definable in $\mathfrak{N}$;
6. every co-finite subset of $|\mathfrak{N}|$ is definable in $\mathfrak{N}$ (where $X \subseteq \mathbb{N}$ is co-finite iff $\mathbb{N} \setminus X$ is finite).

17.5 The Theory of Sets

fol:mat:set:sec

Almost all of mathematics can be developed in the theory of sets. Developing mathematics in this theory involves a number of things. First, it requires a set of axioms for the relation $\in$. A number of different axiom systems have been developed, sometimes with conflicting properties of $\in$. The axiom system known as **ZFC**, Zermelo–Fraenkel set theory with the axiom of choice stands out: it is by far the most widely used and studied, because it turns out that its axioms suffice to prove almost all the things mathematicians expect to be able to prove. But before that can be established, it first is necessary to make clear how we can even *express* all the things mathematicians would like to express. For starters, the language contains no constant symbols or function symbols, so it seems at first glance unclear that we can talk about particular sets (such as $\emptyset$ or $\mathbb{N}$), can talk about operations on sets (such as $X \cup Y$ and $\wp(X)$), let alone other constructions which involve things other than sets, such as relations and functions.

To begin with, "is an element of" is not the only relation we are interested in: "is a subset of" seems almost as important. But we can *define* "is a subset of" in terms of "is an element of." To do this, we have to find a formula $\varphi(x, y)$ in the language of set theory which is satisfied by a pair of sets $\langle X, Y \rangle$ iff $X \subseteq Y$. But X is a subset of Y just in case all elements of X are also elements of Y. So we can define $\subseteq$ by the formula

$$\forall z\,(z \in x \to z \in y)$$

Now, whenever we want to use the relation $\subseteq$ in a formula, we could instead use that formula (with x and y suitably replaced, and the bound variable z renamed if necessary). For instance, extensionality of sets means that if any sets x and y are contained in each other, then x and y must be the same set. This can be expressed by $\forall x \forall y\,((x \subseteq y \wedge y \subseteq x) \to x = y)$, or, if we replace $\subseteq$ by the above definition, by

$$\forall x \forall y\,((\forall z\,(z \in x \to z \in y) \wedge \forall z\,(z \in y \to z \in x)) \to x = y).$$

This is in fact one of the axioms of **ZFC**, the "axiom of extensionality."

There is no constant symbol for $\emptyset$, but we can express "x is empty" by $\neg\exists y\, y \in x$. Then "$\emptyset$ exists" becomes the sentence $\exists x \neg\exists y\, y \in x$. This is another axiom of **ZFC**. (Note that the axiom of extensionality implies that there is only one empty set.) Whenever we want to talk about $\emptyset$ in the language of set theory, we would write this as "there is a set that's empty and ..." As an example, to express the fact that $\emptyset$ is a subset of every set, we could write

$$\exists x\,(\neg\exists y\, y \in x \wedge \forall z\, x \subseteq z)$$

where, of course, $x \subseteq z$ would in turn have to be replaced by its definition.

To talk about operations on sets, such as $X \cup Y$ and $\wp(X)$, we have to use a similar trick. There are no function symbols in the language of set theory,

but we can express the functional relations $X \cup Y = Z$ and $\wp(X) = Y$ by

$$\forall u\,((u \in x \vee u \in y) \leftrightarrow u \in z)$$
$$\forall u\,(u \subseteq x \leftrightarrow u \in y)$$

since the elements of $X \cup Y$ are exactly the sets that are either elements of X or elements of Y, and the elements of $\wp(X)$ are exactly the subsets of X. However, this doesn't allow us to use $x \cup y$ or $\wp(x)$ as if they were terms: we can only use the entire formulas that define the relations $X \cup Y = Z$ and $\wp(X) = Y$. In fact, we do not know that these relations are ever satisfied, i.e., we do not know that unions and power sets always exist. For instance, the sentence $\forall x\,\exists y\,\wp(x) = y$ is another axiom of **ZFC** (the power set axiom).

Now what about talk of ordered pairs or functions? Here we have to explain how we can think of ordered pairs and functions as special kinds of sets. One way to define the ordered pair $\langle x, y\rangle$ is as the set $\{\{x\}, \{x, y\}\}$. But like before, we cannot introduce a function symbol that names this set; we can only define the relation $\langle x, y\rangle = z$, i.e., $\{\{x\}, \{x, y\}\} = z$:

$$\forall u\,(u \in z \leftrightarrow (\forall v\,(v \in u \leftrightarrow v = x) \vee \forall v\,(v \in u \leftrightarrow (v = x \vee v = y))))$$

This says that the elements u of z are exactly those sets which either have x as its only element or have x and y as its only elements (in other words, those sets that are either identical to $\{x\}$ or identical to $\{x, y\}$). Once we have this, we can say further things, e.g., that $X \times Y = Z$:

$$\forall z\,(z \in Z \leftrightarrow \exists x\,\exists y\,(x \in X \wedge y \in Y \wedge \langle x, y\rangle = z))$$

A function $f\colon X \to Y$ can be thought of as the relation $f(x) = y$, i.e., as the set of pairs $\{\langle x, y\rangle : f(x) = y\}$. We can then say that a set f is a function from X to Y if (a) it is a relation $\subseteq X \times Y$, (b) it is total, i.e., for all $x \in X$ there is some $y \in Y$ such that $\langle x, y\rangle \in f$ and (c) it is functional, i.e., whenever $\langle x, y\rangle, \langle x, y'\rangle \in f$, $y = y'$ (because values of functions must be unique). So "f is a function from X to Y" can be written as:

$$\begin{aligned}&\forall u\,(u \in f \to \exists x\,\exists y\,(x \in X \wedge y \in Y \wedge \langle x, y\rangle = u)) \wedge\\ &\forall x\,(x \in X \to (\exists y\,(y \in Y \wedge \mathrm{maps}(f, x, y)) \wedge\\ &\qquad\qquad (\forall y\,\forall y'\,((\mathrm{maps}(f, x, y) \wedge \mathrm{maps}(f, x, y')) \to y = y')))\end{aligned}$$

where $\mathrm{maps}(f, x, y)$ abbreviates $\exists v\,(v \in f \wedge \langle x, y\rangle = v)$ (this formula expresses "$f(x) = y$").

It is now also not hard to express that $f\colon X \to Y$ is injective, for instance:

$$\begin{aligned}f\colon X \to Y \wedge \forall x\,\forall x'\,((x \in X \wedge x' \in X \wedge\\ \exists y\,(\mathrm{maps}(f, x, y) \wedge \mathrm{maps}(f, x', y))) \to x = x')\end{aligned}$$

A function $f\colon X \to Y$ is injective iff, whenever f maps $x, x' \in X$ to a single y, $x = x'$. If we abbreviate this formula as $\mathrm{inj}(f, X, Y)$, we're already in a position

to state in the language of set theory something as non-trivial as Cantor's theorem: there is no injective function from $\wp(X)$ to X:

$$\forall X\, \forall Y\, (\wp(X) = Y \to \neg\exists f\, \mathrm{inj}(f, Y, X))$$

One might think that set theory requires another axiom that guarantees the existence of a set for every defining property. If $\varphi(x)$ is a formula of set theory with the variable x free, we can consider the sentence

$$\exists y\, \forall x\, (x \in y \leftrightarrow \varphi(x)).$$

This sentence states that there is a set y whose elements are all and only those x that satisfy $\varphi(x)$. This schema is called the "comprehension principle." It looks very useful; unfortunately it is inconsistent. Take $\varphi(x) \equiv \neg x \in x$, then the comprehension principle states

$$\exists y\, \forall x\, (x \in y \leftrightarrow x \notin x),$$

i.e., it states the existence of a set of all sets that are not elements of themselves. No such set can exist—this is Russell's Paradox. **ZFC**, in fact, contains a restricted—and consistent—version of this principle, the separation principle:

$$\forall z\, \exists y\, \forall x\, (x \in y \leftrightarrow (x \in z \wedge \varphi(x)).$$

Problem 17.4. Show that the comprehension principle is inconsistent by giving a derivation that shows

$$\exists y\, \forall x\, (x \in y \leftrightarrow x \notin x) \vdash \bot.$$

It may help to first show $(A \to \neg A) \wedge (\neg A \to A) \vdash \bot$.

---

content/first-order-logic/models-theories/size-of-structures.tex

17.6 Expressing the Size of Structures

explanation

There are some properties of structures we can express even without using the non-logical symbols of a language. For instance, there are sentences which are true in a structure iff the domain of the structure has at least, at most, or exactly a certain number n of elements.

fol:mat:siz:sec

Proposition 17.11. *The sentence*

$$\begin{aligned}\varphi_{\geq n} \equiv \exists x_1\, \exists x_2\, \ldots \exists x_n\ & \\ & (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge \cdots \wedge x_1 \neq x_n \wedge \\ & \qquad x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge \cdots \wedge x_2 \neq x_n \wedge \\ & \qquad\qquad \vdots \\ & \qquad x_{n-1} \neq x_n)\end{aligned}$$

is true in a structure $\mathfrak{M}$ *iff* $|\mathfrak{M}|$ *contains at least* n *elements. Consequently,* $\mathfrak{M} \vDash \neg\varphi_{\geq n+1}$ *iff* $|\mathfrak{M}|$ *contains at most* n *elements.*

Proposition 17.12. *The sentence*

$$\begin{aligned}\varphi_{=n} \equiv \exists x_1\, \exists x_2\, \dots \exists x_n \\ (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge \dots \wedge x_1 \neq x_n \wedge \\ x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge \dots \wedge x_2 \neq x_n \wedge \\ \vdots \\ x_{n-1} \neq x_n \wedge \\ \forall y\, (y = x_1 \vee \dots \vee y = x_n))\end{aligned}$$

is true in a structure $\mathfrak{M}$ *iff* $|\mathfrak{M}|$ *contains exactly* n *elements.*

Proposition 17.13. *A structure is infinite iff it is a model of*

$$\{\varphi_{\geq 1}, \varphi_{\geq 2}, \varphi_{\geq 3}, \dots\}.$$

There is no single purely logical sentence which is true in $\mathfrak{M}$ iff $|\mathfrak{M}|$ is infinite. However, one can give sentences with non-logical predicate symbols which only have infinite models (although not every infinite structure is a model of them). The property of being a finite structure, and the property of being a non-enumerable structure cannot even be expressed with an infinite set of sentences. These facts follow from the compactness and Löwenheim–Skolem theorems.

Chapter 18

Derivation Systems

This chapter collects general material on derivation systems. A textbook using a specific system can insert the introduction section plus the relevant survey section at the beginning of the chapter introducing that system.

content/first-order-logic/proof-systems/introduction.tex

18.1 Introduction

fol:prf:int:sec

Logics commonly have both a semantics and a derivation system. The semantics concerns concepts such as truth, satisfiability, validity, and entailment. The purpose of derivation systems is to provide a purely syntactic method of establishing entailment and validity. They are purely syntactic in the sense that a derivation in such a system is a finite syntactic object, usually a sequence (or other finite arrangement) of sentences or formulas. Good derivation systems have the property that any given sequence or arrangement of sentences or formulas can be verified mechanically to be "correct."

The simplest (and historically first) derivation systems for first-order logic were *axiomatic*. A sequence of formulas counts as a derivation in such a system if each individual formula in it is either among a fixed set of "axioms" or follows from formulas coming before it in the sequence by one of a fixed number of "inference rules"—and it can be mechanically verified if a formula is an axiom and whether it follows correctly from other formulas by one of the inference rules. Axiomatic derivation systems are easy to describe—and also easy to handle meta-theoretically—but derivations in them are hard to read and understand, and are also hard to produce.

Other derivation systems have been developed with the aim of making it easier to construct derivations or easier to understand derivations once they are complete. Examples are natural deduction, truth trees, also known as tableaux proofs, and the sequent calculus. Some derivation systems are designed especially with mechanization in mind, e.g., the resolution method is easy to implement in software (but its derivations are essentially impossible to understand). Most of these other derivation systems represent derivations as trees of formulas rather than sequences. This makes it easier to see which parts of a derivation depend on which other parts.

So for a given logic, such as first-order logic, the different derivation systems will give different explications of what it is for a sentence to be a *theorem* and what it means for a sentence to be derivable from some others. However that is done (via axiomatic derivations, natural deductions, sequent derivations, truth trees, resolution refutations), we want these relations to match the semantic notions of validity and entailment. Let's write $\vdash \varphi$ for "φ is a theorem" and "$\Gamma \vdash \varphi$" for "φ is derivable from Γ." However $\vdash$ is defined, we want it to match up with $\models$, that is:

1. $\vdash \varphi$ if and only if $\models \varphi$
2. $\Gamma \vdash \varphi$ if and only if $\Gamma \models \varphi$

The "only if" direction of the above is called *soundness*. A derivation system is sound if derivability guarantees entailment (or validity). Every decent derivation system has to be sound; unsound derivation systems are not useful at all.

After all, the entire purpose of a derivation is to provide a syntactic guarantee of validity or entailment. We'll prove soundness for the derivation systems we present.

The converse "if" direction is also important: it is called *completeness*. A complete derivation system is strong enough to show that φ is a theorem whenever φ is valid, and that $\Gamma \vdash \varphi$ whenever $\Gamma \models \varphi$. Completeness is harder to establish, and some logics have no complete derivation systems. First-order logic does. Kurt Gödel was the first one to prove completeness for a derivation system of first-order logic in his 1929 dissertation.

Another concept that is connected to derivation systems is that of *consistency*. A set of sentences is called inconsistent if anything whatsoever can be derived from it, and consistent otherwise. Inconsistency is the syntactic counterpart to unsatisfiablity: like unsatisfiable sets, inconsistent sets of sentences do not make good theories, they are defective in a fundamental way. Consistent sets of sentences may not be true or useful, but at least they pass that minimal threshold of logical usefulness. For different derivation systems the specific definition of consistency of sets of sentences might differ, but like $\vdash$, we want consistency to coincide with its semantic counterpart, satisfiability. We want it to always be the case that Γ is consistent if and only if it is satisfiable. Here, the "only if" direction amounts to completeness (consistency guarantees satisfiability), and the "if" direction amounts to soundness (satisfiability guarantees consistency). In fact, for classical first-order logic, the two versions of soundness and completeness are equivalent.

---

content/first-order-logic/proof-systems/sequent-calculus.tex

18.2 The Sequent Calculus

fol:prf:seq:sec

While many derivation systems operate with arrangements of sentences, the sequent calculus operates with *sequents*. A sequent is an expression of the form

$$\varphi_1, \ldots, \varphi_m \Rightarrow \psi_1, \ldots, \psi_m,$$

that is a pair of sequences of sentences, separated by the sequent symbol $\Rightarrow$. Either sequence may be empty. A derivation in the sequent calculus is a tree of sequents, where the topmost sequents are of a special form (they are called "initial sequents" or "axioms") and every other sequent follows from the sequents immediately above it by one of the rules of inference. The rules of inference either manipulate the sentences in the sequents (adding, removing, or rearranging them on either the left or the right), or they introduce a complex formula in the conclusion of the rule. For instance, the $\wedge$L rule allows the inference from $\varphi, \Gamma \Rightarrow \Delta$ to $\varphi \wedge \psi, \Gamma \Rightarrow \Delta$, and the $\rightarrow$R allows the inference from $\varphi, \Gamma \Rightarrow \Delta, \psi$ to $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$, for any Γ, Δ, φ, and ψ. (In particular, Γ and Δ may be empty.)

The $\vdash$ relation based on the sequent calculus is defined as follows: $\Gamma \vdash \varphi$ iff there is some sequence Γ_0 such that every φ in Γ_0 is in Γ and there is a

derivation with the sequent $\Gamma_0 \Rightarrow \varphi$ at its root. φ is a theorem in the sequent calculus if the sequent $\Rightarrow \varphi$ has a derivation. For instance, here is a derivation that shows that $\vdash (\varphi \wedge \psi) \to \varphi$:

$$\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi}\wedge\text{L}}{\Rightarrow (\varphi \wedge \psi) \to \varphi}\to\text{R}$$

A set Γ is inconsistent in the sequent calculus if there is a derivation of $\Gamma_0 \Rightarrow$ (where every $\varphi \in \Gamma_0$ is in Γ and the right side of the sequent is empty). Using the rule WR, any sentence can be derived from an inconsistent set.

The sequent calculus was invented in the 1930s by Gerhard Gentzen. Because of its systematic and symmetric design, it is a very useful formalism for developing a theory of derivations. It is relatively easy to find derivations in the sequent calculus, but these derivations are often hard to read and their connection to proofs are sometimes not easy to see. It has proved to be a very elegant approach to derivation systems, however, and many logics have sequent calculus systems.

---

content/first-order-logic/proof-systems/natural-deduction.tex

18.3 Natural Deduction

fol:prf:ntd:sec

Natural deduction is a derivation system intended to mirror actual reasoning (especially the kind of regimented reasoning employed by mathematicians). Actual reasoning proceeds by a number of "natural" patterns. For instance, proof by cases allows us to establish a conclusion on the basis of a disjunctive premise, by establishing that the conclusion follows from either of the disjuncts. Indirect proof allows us to establish a conclusion by showing that its negation leads to a contradiction. Conditional proof establishes a conditional claim "if ... then ..." by showing that the consequent follows from the antecedent. Natural deduction is a formalization of some of these natural inferences. Each of the logical connectives and quantifiers comes with two rules, an introduction and an elimination rule, and they each correspond to one such natural inference pattern. For instance, $\to$Intro corresponds to conditional proof, and $\vee$Elim to proof by cases. A particularly simple rule is $\wedge$Elim which allows the inference from $\varphi \wedge \psi$ to φ (or ψ).

One feature that distinguishes natural deduction from other derivation systems is its use of assumptions. A derivation in natural deduction is a tree of formulas. A single formula stands at the root of the tree of formulas, and the "leaves" of the tree are formulas from which the conclusion is derived. In natural deduction, some leaf formulas play a role inside the derivation but are "used up" by the time the derivation reaches the conclusion. This corresponds to the practice, in actual reasoning, of introducing hypotheses which only remain in effect for a short while. For instance, in a proof by cases, we assume the truth of each of the disjuncts; in conditional proof, we assume the truth

of the antecedent; in indirect proof, we assume the truth of the negation of the conclusion. This way of introducing hypothetical assumptions and then doing away with them in the service of establishing an intermediate step is a hallmark of natural deduction. The formulas at the leaves of a natural deduction derivation are called assumptions, and some of the rules of inference may "discharge" them. For instance, if we have a derivation of ψ from some assumptions which include φ, then the $\to$Intro rule allows us to infer $\varphi \to \psi$ and discharge any assumption of the form φ. (To keep track of which assumptions are discharged at which inferences, we label the inference and the assumptions it discharges with a number.) The assumptions that remain undischarged at the end of the derivation are together sufficient for the truth of the conclusion, and so a derivation establishes that its undischarged assumptions entail its conclusion.

The relation $\Gamma \vdash \varphi$ based on natural deduction holds iff there is a derivation in which φ is the last sentence in the tree, and every leaf which is undischarged is in Γ. φ is a theorem in natural deduction iff there is a derivation in which φ is the last sentence and all assumptions are discharged. For instance, here is a derivation that shows that $\vdash (\varphi \wedge \psi) \to \varphi$:

$$\dfrac{\dfrac{[\varphi \wedge \psi]^1}{\varphi}\,\wedge\text{Elim}}{(\varphi \wedge \psi) \to \varphi}\,{}^{1}\,\to\text{Intro}$$

The label 1 indicates that the assumption $\varphi \wedge \psi$ is discharged at the $\to$Intro inference.

A set Γ is inconsistent iff $\Gamma \vdash \bot$ in natural deduction. The rule $\bot_I$ makes it so that from an inconsistent set, any sentence can be derived.

Natural deduction systems were developed by Gerhard Gentzen and Stanisław Jaśkowski in the 1930s, and later developed by Dag Prawitz and Frederic Fitch. Because its inferences mirror natural methods of proof, it is favored by philosophers. The versions developed by Fitch are often used in introductory logic textbooks. In the philosophy of logic, the rules of natural deduction have sometimes been taken to give the meanings of the logical operators ("proof-theoretic semantics").

---

content/first-order-logic/proof-systems/tableaux.tex

18.4 Tableaux

fol:prf:tab:sec

While many derivation systems operate with arrangements of sentences, tableaux operate with signed formulas. A signed formula is a pair consisting of a truth value sign ($\mathbb{T}$ or $\mathbb{F}$) and a sentence

$$\mathbb{T}\,\varphi \text{ or } \mathbb{F}\,\varphi.$$

A tableau consists of signed formulas arranged in a downward-branching tree. It begins with a number of *assumptions* and continues with signed formulas

which result from one of the signed formulas above it by applying one of the rules of inference. Each rule allows us to add one or more signed formulas to the end of a branch, or two signed formulas side by side—in this case a branch splits into two, with the two added signed formulas forming the ends of the two branches.

A rule applied to a complex signed formula results in the addition of signed formulas which are immediate sub-formulas. They come in pairs, one rule for each of the two signs. For instance, the $\wedge\mathbb{T}$ rule applies to $\mathbb{T}\,\varphi \wedge \psi$, and allows the addition of both the two signed formulas $\mathbb{T}\,\varphi$ and $\mathbb{T}\,\psi$ to the end of any branch containing $\mathbb{T}\,\varphi \wedge \psi$, and the rule $\varphi \wedge \psi\mathbb{F}$ allows a branch to be split by adding $\mathbb{F}\,\varphi$ and $\mathbb{F}\,\psi$ side-by-side. A tableau is closed if every one of its branches contains a matching pair of signed formulas $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$.

The $\vdash$ relation based on tableaux is defined as follows: $\Gamma \vdash \varphi$ iff there is some finite set $\Gamma_0 = \{\psi_1, \ldots, \psi_n\} \subseteq \Gamma$ such that there is a closed tableau for the assumptions

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$

For instance, here is a closed tableau that shows that $\vdash (\varphi \wedge \psi) \to \varphi$:

1.	$\mathbb{F}\,(\varphi \wedge \psi) \to \varphi$	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\varphi$	$\to\mathbb{T}\,2$
5.	$\mathbb{T}\,\psi$	$\to\mathbb{T}\,2$
	$\otimes$	

A set Γ is inconsistent in the tableau calculus if there is a closed tableau for assumptions

$$\{\mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$

for some $\psi_i \in \Gamma$.

Tableaux were invented in the 1950s independently by Evert Beth and Jaakko Hintikka, and simplified and popularized by Raymond Smullyan. They are very easy to use, since constructing a tableau is a very systematic procedure. Because of the systematic nature of tableaux, they also lend themselves to implementation by computer. However, a tableau is often hard to read and their connection to proofs are sometimes not easy to see. The approach is also quite general, and many different logics have tableau systems. Tableaux also help us to find structures that satisfy given (sets of) sentences: if the set is satisfiable, it won't have a closed tableau, i.e., any tableau will have an open branch. The satisfying structure can be "read off" an open branch, provided every rule it is possible to apply has been applied on that branch. There is also a very close connection to the sequent calculus: essentially, a closed tableau is a condensed derivation in the sequent calculus, written upside-down.

---

content/first-order-logic/proof-systems/axiomatic-deduction.tex

18.5 Axiomatic Derivations

fol:prf:axd:sec

Axiomatic derivations are the oldest and simplest logical derivation systems. Its derivations are simply sequences of sentences. A sequence of sentences counts as a correct derivation if every sentence φ in it satisfies one of the following conditions:

1. φ is an axiom, or
2. φ is an element of a given set Γ of sentences, or
3. φ is justified by a rule of inference.

To be an axiom, φ has to have the form of one of a number of fixed sentence schemas. There are many sets of axiom schemas that provide a satisfactory (sound and complete) derivation system for first-order logic. Some are organized according to the connectives they govern, e.g., the schemas

$$\varphi \to (\psi \to \varphi) \qquad \psi \to (\psi \vee \chi) \qquad (\psi \wedge \chi) \to \psi$$

are common axioms that govern $\to$, $\vee$ and $\wedge$. Some axiom systems aim at a minimal number of axioms. Depending on the connectives that are taken as primitives, it is even possible to find axiom systems that consist of a single axiom.

A rule of inference is a conditional statement that gives a sufficient condition for a sentence in a derivation to be justified. Modus ponens is one very common such rule: it says that if φ and $\varphi \to \psi$ are already justified, then ψ is justified. This means that a line in a derivation containing the sentence ψ is justified, provided that both φ and $\varphi \to \psi$ (for some sentence φ) appear in the derivation before ψ.

The $\vdash$ relation based on axiomatic derivations is defined as follows: $\Gamma \vdash \varphi$ iff there is a derivation with the sentence φ as its last formula (and Γ is taken as the set of sentences in that derivation which are justified by (2) above). φ is a theorem if φ has a derivation where Γ is empty, i.e., every sentence in the derivation is justified either by (1) or (3). For instance, here is a derivation that shows that $\vdash \varphi \to (\psi \to (\psi \vee \varphi))$:

1. $\psi \to (\psi \vee \varphi)$
2. $(\psi \to (\psi \vee \varphi)) \to (\varphi \to (\psi \to (\psi \vee \varphi)))$
3. $\varphi \to (\psi \to (\psi \vee \varphi))$

The sentence on line 1 is of the form of the axiom $\varphi \to (\varphi \vee \psi)$ (with the roles of φ and ψ reversed). The sentence on line 2 is of the form of the axiom $\varphi \to (\psi \to \varphi)$. Thus, both lines are justified. Line 3 is justified by modus ponens: if we abbreviate it as θ, then line 2 has the form $\chi \to \theta$, where χ is $\psi \to (\psi \vee \varphi)$, i.e., line 1.

A set Γ is inconsistent if $\Gamma \vdash \bot$. A complete axiom system will also prove that $\bot \to \varphi$ for any φ, and so if Γ is inconsistent, then $\Gamma \vdash \varphi$ for any φ.

Systems of axiomatic derivations for logic were first given by Gottlob Frege in his 1879 *Begriffsschrift*, which for this reason is often considered the first work of modern logic. They were perfected in Alfred North Whitehead and Bertrand Russell's *Principia Mathematica* and by David Hilbert and his students in the 1920s. They are thus often called "Frege systems" or "Hilbert systems." They are very versatile in that it is often easy to find an axiomatic system for a logic. Because derivations have a very simple structure and only one or two inference rules, it is also relatively easy to prove things *about* them. However, they are very hard to use in practice, i.e., it is difficult to find and write proofs.

Chapter 19

The Sequent Calculus

This chapter presents Gentzen's standard sequent calculus LK for classical first-order logic. It could use more examples and exercises. To include or exclude material relevant to the sequent calculus as a proof system, use the "prfLK" tag.

---

content/first-order-logic/sequent-calculus/rules-and-proofs.tex

19.1 Rules and Derivations

fol:seq:rul:sec

For the following, let $\Gamma, \Delta, \Pi, \Lambda$ represent finite sequences of sentences.

Definition 19.1 (Sequent). A *sequent* is an expression of the form

$$\Gamma \Rightarrow \Delta$$

where Γ and Δ are finite (possibly empty) sequences of sentences of the language $\mathcal{L}$. Γ is called the *antecedent*, while Δ is the *succedent*.

explanation

The intuitive idea behind a sequent is: if all of the sentences in the antecedent hold, then at least one of the sentences in the succedent holds. That is, if $\Gamma = \langle \varphi_1, \ldots, \varphi_m \rangle$ and $\Delta = \langle \psi_1, \ldots, \psi_n \rangle$, then $\Gamma \Rightarrow \Delta$ holds iff

$$(\varphi_1 \wedge \cdots \wedge \varphi_m) \to (\psi_1 \vee \cdots \vee \psi_n)$$

holds. There are two special cases: where Γ is empty and when Δ is empty. When Γ is empty, i.e., $m = 0$, $\Rightarrow \Delta$ holds iff $\psi_1 \vee \cdots \vee \psi_n$ holds. When Δ is empty, i.e., $n = 0$, $\Gamma \Rightarrow$ holds iff $\neg(\varphi_1 \wedge \cdots \wedge \varphi_m)$ does. We say a sequent is valid iff the corresponding sentence is valid.

If Γ is a sequence of sentences, we write Γ, φ for the result of appending φ to the right end of Γ (and φ, Γ for the result of appending φ to the left end of Γ). If Δ is a sequence of sentences also, then Γ, Δ is the concatenation of the two sequences.

Definition 19.2 (Initial Sequent). An *initial sequent* is a sequent of one of the following forms:

1. $\varphi \Rightarrow \varphi$
2. $\Rightarrow \top$
3. $\bot \Rightarrow$

for any sentence φ in the language.

Derivations in the sequent calculus are certain trees of sequents, where the topmost sequents are initial sequents, and if a sequent stands below one or two other sequents, it must follow correctly by a rule of inference. The rules for **LK** are divided into two main types: *logical* rules and *structural* rules. The logical rules are named for the main operator of the sentence containing φ and/or ψ in the lower sequent. Each one comes in two versions, one for inferring a sequent with the sentence containing the logical operator on the left, and one with the sentence on the right.

---

content/first-order-logic/sequent-calculus/propositional-rules.tex

19.2 Propositional Rules

fol:seq:prl:sec

Rules for $\neg$

$$\frac{\Gamma \Rightarrow \Delta, \varphi}{\neg\varphi, \Gamma \Rightarrow \Delta}\,\neg\mathrm{L} \qquad \frac{\varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg\varphi}\,\neg\mathrm{R}$$

Rules for $\wedge$

$$\frac{\varphi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta} \wedge\text{L} \qquad \frac{\psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta} \wedge\text{L} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi \qquad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \wedge\text{R}$$

Rules for $\vee$

$$\frac{\varphi, \Gamma \Rightarrow \Delta \qquad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta} \vee\text{L} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee\text{R} \qquad \frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee\text{R}$$

Rules for $\rightarrow$

$$\frac{\Gamma \Rightarrow \Delta, \varphi \qquad \psi, \Pi \Rightarrow \Lambda}{\varphi \rightarrow \psi, \Gamma, \Pi \Rightarrow \Delta, \Lambda} \rightarrow\text{L} \qquad \frac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi} \rightarrow\text{R}$$

---

content/first-order-logic/sequent-calculus/quantifier-rules.tex

19.3 Quantifier Rules

Rules for $\forall$

fol:seq:qrl:sec

$$\frac{\varphi(t), \Gamma \Rightarrow \Delta}{\forall x\, \varphi(x), \Gamma \Rightarrow \Delta} \forall\text{L} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi(a)}{\Gamma \Rightarrow \Delta, \forall x\, \varphi(x)} \forall\text{R}$$

In $\forall$L, t is a closed term (i.e., one without variables). In $\forall$R, a is a constant symbol which must not occur anywhere in the lower sequent of the $\forall$R rule. We call a the *eigenvariable* of the $\forall$R inference.[1]

[1] We use the term "eigenvariable" even though a in the above rule is a constant symbol. This has historical reasons.

Rules for ∃

$$\frac{\varphi(a), \Gamma \Rightarrow \Delta}{\exists x\, \varphi(x), \Gamma \Rightarrow \Delta}\ \exists\text{L} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi(t)}{\Gamma \Rightarrow \Delta, \exists x\, \varphi(x)}\ \exists\text{R}$$

Again, t is a closed term, and a is a constant symbol which does not occur in the lower sequent of the ∃L rule. We call a the *eigenvariable* of the ∃L inference.

The condition that an eigenvariable not occur in the lower sequent of the ∀R or ∃L inference is called the *eigenvariable condition*.

Recall the convention that when φ is a formula with the variable x free, we indicate this by writing $\varphi(x)$. In the same context, $\varphi(t)$ then is short for $\varphi[t/x]$. So we could also write the ∃R rule as: explanation

$$\frac{\Gamma \Rightarrow \Delta, \varphi[t/x]}{\Gamma \Rightarrow \Delta, \exists x\, \varphi}\ \exists\text{R}$$

Note that t may already occur in φ, e.g., φ might be $P(t, x)$. Thus, inferring $\Gamma \Rightarrow \Delta, \exists x\, P(t, x)$ from $\Gamma \Rightarrow \Delta, P(t, t)$ is a correct application of ∃R—you may "replace" one or more, and not necessarily all, occurrences of t in the premise by the bound variable x. However, the eigenvariable conditions in ∀R and ∃L require that the constant symbol a does not occur in φ. So, you cannot correctly infer $\Gamma \Rightarrow \Delta, \forall x\, P(a, x)$ from $\Gamma \Rightarrow \Delta, P(a, a)$ using ∀R.

In ∃R and ∀L there are no restrictions on the term t. On the other hand, in the ∃L and ∀R rules, the eigenvariable condition requires that the constant symbol a does not occur anywhere outside of $\varphi(a)$ in the upper sequent. It is necessary to ensure that the system is sound, i.e., only derives sequents that are valid. Without this condition, the following would be allowed: explanation

$$\dfrac{\dfrac{\varphi(a) \Rightarrow \varphi(a)}{\exists x\, \varphi(x) \Rightarrow \varphi(a)}\ {*}\exists\text{L}}{\exists x\, \varphi(x) \Rightarrow \forall x\, \varphi(x)}\ \forall\text{R} \qquad \dfrac{\dfrac{\varphi(a) \Rightarrow \varphi(a)}{\varphi(a) \Rightarrow \forall x\, \varphi(x)}\ {*}\forall\text{R}}{\exists x\, \varphi(x) \Rightarrow \forall x\, \varphi(x)}\ \exists\text{L}$$

However, $\exists x\, \varphi(x) \Rightarrow \forall x\, \varphi(x)$ is not valid.

---

content/first-order-logic/sequent-calculus/structural-rules.tex

19.4 Structural Rules

fol:seq:srl:sec

We also need a few rules that allow us to rearrange sentences in the left and right side of a sequent. Since the logical rules require that the sentences in the premise which the rule acts upon stand either to the far left or to the far right, we need an "exchange" rule that allows us to move sentences to the right position. It's also important sometimes to be able to combine two identical sentences into one, and to add a sentence on either side.

Weakening

$$\frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\,\text{WL} \qquad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi}\,\text{WR}$$

Contraction

$$\frac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\,\text{CL} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi}\,\text{CR}$$

Exchange

$$\frac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta}\,\text{XL} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi, \psi, \Lambda}{\Gamma \Rightarrow \Delta, \psi, \varphi, \Lambda}\,\text{XR}$$

A series of weakening, contraction, and exchange inferences will often be indicated by double inference lines.

The following rule, called "cut," is not strictly speaking necessary, but makes it a lot easier to reuse and combine derivations.

$$\frac{\Gamma \Rightarrow \Delta, \varphi \qquad \varphi, \Pi \Rightarrow \Lambda}{\Gamma, \Pi \Rightarrow \Delta, \Lambda}\,\text{Cut}$$

---

content/first-order-logic/sequent-calculus/derivations.tex

19.5 Derivations

explanation

We've said what an initial sequent looks like, and we've given the rules of inference. Derivations in the sequent calculus are inductively generated from these: each derivation either is an initial sequent on its own, or consists of one or two derivations followed by an inference.

fol:seq:der:sec

Definition 19.3 (LK derivation). An **LK**-*derivation* of a sequent S is a finite tree of sequents satisfying the following conditions:

1. The topmost sequents of the tree are initial sequents.
2. The bottommost sequent of the tree is S.

3. Every sequent in the tree except S is a premise of a correct application of an inference rule whose conclusion stands directly below that sequent in the tree.

We then say that S is the *end-sequent* of the derivation and that S is *derivable in* **LK** (or **LK**-derivable).

Example 19.4. Every initial sequent, e.g., $\chi \Rightarrow \chi$ is a derivation. We can obtain a new derivation from this by applying, say, the WL rule,

$$\dfrac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\,\mathrm{WL}$$

The rule, however, is meant to be general: we can replace the φ in the rule with any sentence, e.g., also with θ. If the premise matches our initial sequent $\chi \Rightarrow \chi$, that means that both Γ and Δ are just χ, and the conclusion would then be $\theta, \chi \Rightarrow \chi$. So, the following is a derivation:

$$\dfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\,\mathrm{WL}$$

We can now apply another rule, say XL, which allows us to switch two sentences on the left. So, the following is also a correct derivation:

$$\dfrac{\dfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\,\mathrm{WL}}{\chi, \theta \Rightarrow \chi}\,\mathrm{XL}$$

In this application of the rule, which was given as

$$\dfrac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta,}\,\mathrm{XL}$$

both Γ and Π were empty, Δ is χ, and the roles of φ and ψ are played by θ and χ, respectively. In much the same way, we also see that

$$\dfrac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta}\,\mathrm{WL}$$

is a derivation. Now we can take these two derivations, and combine them using $\wedge$R. That rule was

$$\dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\,\wedge\mathrm{R}$$

In our case, the premises must match the last sequents of the derivations ending in the premises. That means that Γ is χ, θ, Δ is empty, φ is χ and ψ is θ. So the conclusion, if the inference should be correct, is $\chi, \theta \Rightarrow \chi \wedge \theta$.

$$\dfrac{\dfrac{\dfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\,\mathrm{WL}}{\chi, \theta \Rightarrow \chi}\,\mathrm{XL} \qquad \dfrac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta}\,\mathrm{WL}}{\chi, \theta \Rightarrow \chi \wedge \theta}\,\wedge\mathrm{R}$$

Of course, we can also reverse the premises, then φ would be θ and ψ would be χ.

$$\dfrac{\dfrac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta}\,\text{WL} \qquad \dfrac{\dfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\,\text{WL}}{\chi, \theta \Rightarrow \chi}\,\text{XL}}{\chi, \theta \Rightarrow \theta \wedge \chi}\,\wedge\text{R}$$

---

content/first-order-logic/sequent-calculus/proving-things.tex

19.6 Examples of Derivations

fol:seq:pro:sec

Example 19.5. Give an **LK**-derivation for the sequent $\varphi \wedge \psi \Rightarrow \varphi$.

We begin by writing the desired end-sequent at the bottom of the derivation.

$$\dfrac{}{\varphi \wedge \psi \Rightarrow \varphi}$$

Next, we need to figure out what kind of inference could have a lower sequent of this form. This could be a structural rule, but it is a good idea to start by looking for a logical rule. The only logical connective occurring in the lower sequent is $\wedge$, so we're looking for an $\wedge$ rule, and since the $\wedge$ symbol occurs in the antecedent, we're looking at the $\wedge$L rule.

$$\dfrac{}{\varphi \wedge \psi \Rightarrow \varphi}\,\wedge\text{L}$$

There are two options for what could have been the upper sequent of the $\wedge$L inference: we could have an upper sequent of $\varphi \Rightarrow \varphi$, or of $\psi \Rightarrow \varphi$. Clearly, $\varphi \Rightarrow \varphi$ is an initial sequent (which is a good thing), while $\psi \Rightarrow \varphi$ is not derivable in general. We fill in the upper sequent:

$$\dfrac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi}\,\wedge\text{L}$$

We now have a correct **LK**-derivation of the sequent $\varphi \wedge \psi \Rightarrow \varphi$.

Example 19.6. Give an **LK**-derivation for the sequent $\neg\varphi \vee \psi \Rightarrow \varphi \to \psi$.

Begin by writing the desired end-sequent at the bottom of the derivation.

$$\dfrac{}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}$$

To find a logical rule that could give us this end-sequent, we look at the logical connectives in the end-sequent: $\neg$, $\vee$, and $\to$. We only care at the moment about $\vee$ and $\to$ because they are main operators of sentences in the end-sequent, while $\neg$ is inside the scope of another connective, so we will take care of it later. Our options for logical rules for the final inference are therefore the $\vee$L rule and the $\to$R rule. We could pick either rule, really, but let's pick the $\to$R rule (if for no reason other than it allows us to put off splitting into two branches).

According to the form of →R inferences which can yield the lower sequent, this must look like:

$$\dfrac{\dfrac{}{\varphi, \neg\varphi \lor \psi \Rightarrow \psi}}{\neg\varphi \lor \psi \Rightarrow \varphi \to \psi} \to\text{R}$$

If we move $\neg\varphi \lor \psi$ to the outside of the antecedent, we can apply the ∨L rule. According to the schema, this must split into two upper sequents as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\varphi, \varphi \Rightarrow \psi} \quad \dfrac{}{\psi, \varphi \Rightarrow \psi}}{\neg\varphi \lor \psi, \varphi \Rightarrow \psi} \lor\text{L}}{\varphi, \neg\varphi \lor \psi \Rightarrow \psi} \text{XR}}{\neg\varphi \lor \psi \Rightarrow \varphi \to \psi} \to\text{R}$$

Remember that we are trying to wind our way up to initial sequents; we seem to be pretty close! The right branch is just one weakening and one exchange away from an initial sequent and then it is done:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\varphi, \varphi \Rightarrow \psi} \quad \dfrac{\dfrac{\dfrac{}{\psi \Rightarrow \psi}}{\varphi, \psi \Rightarrow \psi} \text{WL}}{\psi, \varphi \Rightarrow \psi} \text{XL}}{\neg\varphi \lor \psi, \varphi \Rightarrow \psi} \lor\text{L}}{\varphi, \neg\varphi \lor \psi \Rightarrow \psi} \text{XR}}{\neg\varphi \lor \psi \Rightarrow \varphi \to \psi} \to\text{R}$$

Now looking at the left branch, the only logical connective in any sentence is the ¬ symbol in the antecedent sentences, so we're looking at an instance of the ¬L rule.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \Rightarrow \psi, \varphi}}{\neg\varphi, \varphi \Rightarrow \psi} \neg\text{L} \quad \dfrac{\dfrac{\dfrac{}{\psi \Rightarrow \psi}}{\varphi, \psi \Rightarrow \psi} \text{WL}}{\psi, \varphi \Rightarrow \psi} \text{XL}}{\neg\varphi \lor \psi, \varphi \Rightarrow \psi} \lor\text{L}}{\varphi, \neg\varphi \lor \psi \Rightarrow \psi} \text{XR}}{\neg\varphi \lor \psi \Rightarrow \varphi \to \psi} \to\text{R}$$

Similarly to how we finished off the right branch, we are just one weakening and one exchange away from finishing off this left branch as well.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \Rightarrow \varphi}}{\varphi \Rightarrow \varphi, \psi} \text{WR}}{\varphi \Rightarrow \psi, \varphi} \text{XR}}{\neg\varphi, \varphi \Rightarrow \psi} \neg\text{L} \quad \dfrac{\dfrac{\dfrac{}{\psi \Rightarrow \psi}}{\varphi, \psi \Rightarrow \psi} \text{WL}}{\psi, \varphi \Rightarrow \psi} \text{XL}}{\neg\varphi \lor \psi, \varphi \Rightarrow \psi} \lor\text{L}}{\varphi, \neg\varphi \lor \psi \Rightarrow \psi} \text{XR}}{\neg\varphi \lor \psi \Rightarrow \varphi \to \psi} \to\text{R}$$

Example 19.7. Give an **LK**-derivation of the sequent $\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)$

Using the techniques from above, we start by writing the desired end-sequent at the bottom.

$$\dfrac{}{\neg\varphi \vee \neg\psi \;\Rightarrow\; \neg(\varphi \wedge \psi)}$$

The available main connectives of sentences in the end-sequent are the $\vee$ symbol and the $\neg$ symbol. It would work to apply either the $\vee$L or the $\neg$R rule here, but we start with the $\neg$R rule because it avoids splitting up into two branches for a moment:

$$\dfrac{\dfrac{}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \;\Rightarrow}}{\neg\varphi \vee \neg\psi \;\Rightarrow\; \neg(\varphi \wedge \psi)}\,\neg\text{R}$$

Now we have a choice of whether to look at the $\wedge$L or the $\vee$L rule. Let's see what happens when we apply the $\wedge$L rule: we have a choice to start with either the sequent $\varphi, \neg\varphi \vee \psi \Rightarrow$ or the sequent $\psi, \neg\varphi \vee \psi \Rightarrow$. Since the derivation is symmetric with regards to φ and ψ, let's go with the former:

$$\dfrac{\dfrac{\dfrac{}{\varphi, \neg\varphi \vee \neg\psi \;\Rightarrow}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \;\Rightarrow}\,\wedge\text{L}}{\neg\varphi \vee \neg\psi \;\Rightarrow\; \neg(\varphi \wedge \psi)}\,\neg\text{R}$$

Continuing to fill in the derivation, we see that we run into a problem:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \;\Rightarrow\; \varphi}}{\neg\varphi, \varphi \;\Rightarrow}\,\neg\text{L} \quad \dfrac{\dfrac{}{\varphi \;\Rightarrow\; \psi}\,?}{\neg\psi, \varphi \;\Rightarrow}\,\neg\text{L}}{\neg\varphi \vee \neg\psi, \varphi \;\Rightarrow}\,\vee\text{L}}{\varphi, \neg\varphi \vee \neg\psi \;\Rightarrow}\,\text{XL}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \;\Rightarrow}\,\wedge\text{L}}{\neg\varphi \vee \neg\psi \;\Rightarrow\; \neg(\varphi \wedge \psi)}\,\neg\text{R}$$

The top of the right branch cannot be reduced any further, and it cannot be brought by way of structural inferences to an initial sequent, so this is not the right path to take. So clearly, it was a mistake to apply the $\wedge$L rule above. Going back to what we had before and carrying out the $\vee$L rule instead, we get

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\varphi, \varphi \wedge \psi \;\Rightarrow} \quad \dfrac{}{\neg\psi, \varphi \wedge \psi \;\Rightarrow}}{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \;\Rightarrow}\,\vee\text{L}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \;\Rightarrow}\,\text{XL}}{\neg\varphi \vee \neg\psi \;\Rightarrow\; \neg(\varphi \wedge \psi)}\,\neg\text{R}$$

Completing each branch as we've done before, we get

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi}\wedge\mathrm{L}}{\neg\varphi, \varphi \wedge \psi \Rightarrow}\neg\mathrm{L} \qquad \dfrac{\dfrac{\psi \Rightarrow \psi}{\varphi \wedge \psi \Rightarrow \psi}\wedge\mathrm{L}}{\neg\psi, \varphi \wedge \psi \Rightarrow}\neg\mathrm{L}}{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \Rightarrow}\vee\mathrm{L}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow}\mathrm{XL}}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)}\neg\mathrm{R}$$

(We could have carried out the $\wedge$ rules lower than the $\neg$ rules in these steps and still obtained a correct derivation).

Example 19.8. So far we haven't used the contraction rule, but it is sometimes required. Here's an example where that happens. Suppose we want to prove $\Rightarrow \varphi \vee \neg\varphi$. Applying $\vee$R backwards would give us one of these two derivations:

$$\dfrac{\dfrac{}{\Rightarrow \varphi}}{\Rightarrow \varphi \vee \neg\varphi}\vee\mathrm{R} \qquad \dfrac{\dfrac{\dfrac{}{\varphi \Rightarrow}}{\Rightarrow \neg\varphi}\neg\mathrm{R}}{\Rightarrow \varphi \vee \neg\varphi}\vee\mathrm{R}$$

Neither of these of course ends in an initial sequent. The trick is to realize that the contraction rule allows us to combine two copies of a sentence into one—and when we're searching for a proof, i.e., going from bottom to top, we can keep a copy of $\varphi \vee \neg\varphi$ in the premise, e.g.,

$$\dfrac{\dfrac{\dfrac{}{\Rightarrow \varphi \vee \neg\varphi, \varphi}}{\Rightarrow \varphi \vee \neg\varphi, \varphi \vee \neg\varphi}\vee\mathrm{R}}{\Rightarrow \varphi \vee \neg\varphi}\mathrm{CR}$$

Now we can apply $\vee$R a second time, and also get $\neg\varphi$, which leads to a complete derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg\varphi}\neg\mathrm{R}}{\Rightarrow \varphi, \varphi \vee \neg\varphi}\vee\mathrm{R}}{\Rightarrow \varphi \vee \neg\varphi, \varphi}\mathrm{XR}}{\Rightarrow \varphi \vee \neg\varphi, \varphi \vee \neg\varphi}\vee\mathrm{R}}{\Rightarrow \varphi \vee \neg\varphi}\mathrm{CR}$$

Problem 19.1. Give derivations of the following sequents:

1. $\varphi \wedge (\psi \wedge \chi) \Rightarrow (\varphi \wedge \psi) \wedge \chi$.
2. $\varphi \vee (\psi \vee \chi) \Rightarrow (\varphi \vee \psi) \vee \chi$.
3. $\varphi \to (\psi \to \chi) \Rightarrow \psi \to (\varphi \to \chi)$.
4. $\varphi \Rightarrow \neg\neg\varphi$.

Problem 19.2. Give derivations of the following sequents:

1. $(\varphi \vee \psi) \to \chi \Rightarrow \varphi \to \chi$.

2. $(\varphi \to \chi) \land (\psi \to \chi) \Rightarrow (\varphi \lor \psi) \to \chi$.
3. $\Rightarrow \neg(\varphi \land \neg\varphi)$.
4. $\psi \to \varphi \Rightarrow \neg\varphi \to \neg\psi$.
5. $\Rightarrow (\varphi \to \neg\varphi) \to \neg\varphi$.
6. $\Rightarrow \neg(\varphi \to \psi) \to \neg\psi$.
7. $\varphi \to \chi \Rightarrow \neg(\varphi \land \neg\chi)$.
8. $\varphi \land \neg\chi \Rightarrow \neg(\varphi \to \chi)$.
9. $\varphi \lor \psi, \neg\psi \Rightarrow \varphi$.
10. $\neg\varphi \lor \neg\psi \Rightarrow \neg(\varphi \land \psi)$.
11. $\Rightarrow (\neg\varphi \land \neg\psi) \to \neg(\varphi \lor \psi)$.
12. $\Rightarrow \neg(\varphi \lor \psi) \to (\neg\varphi \land \neg\psi)$.

Problem 19.3. Give derivations of the following sequents:

1. $\neg(\varphi \to \psi) \Rightarrow \varphi$.
2. $\neg(\varphi \land \psi) \Rightarrow \neg\varphi \lor \neg\psi$.
3. $\varphi \to \psi \Rightarrow \neg\varphi \lor \psi$.
4. $\Rightarrow \neg\neg\varphi \to \varphi$.
5. $\varphi \to \psi, \neg\varphi \to \psi \Rightarrow \psi$.
6. $(\varphi \land \psi) \to \chi \Rightarrow (\varphi \to \chi) \lor (\psi \to \chi)$.
7. $(\varphi \to \psi) \to \varphi \Rightarrow \varphi$.
8. $\Rightarrow (\varphi \to \psi) \lor (\psi \to \chi)$.

(These all require the CR rule.)

---

content/first-order-logic/sequent-calculus/proving-things-quant.tex

19.7 Derivations with Quantifiers

fol:seq:prq:sec

Example 19.9. Give an **LK**-derivation of the sequent $\exists x\, \neg\varphi(x) \Rightarrow \neg\forall x\, \varphi(x)$.

When dealing with quantifiers, we have to make sure not to violate the eigenvariable condition, and sometimes this requires us to play around with the order of carrying out certain inferences. In general, it helps to try and take care of rules subject to the eigenvariable condition first (they will be lower down in the finished proof). Also, it is a good idea to try and look ahead and try to guess what the initial sequent might look like. In our case, it will have to be something like $\varphi(a) \Rightarrow \varphi(a)$. That means that when we are "reversing" the quantifier rules, we will have to pick the same term—what we will call a—for both the $\forall$ and the $\exists$ rule. If we picked different terms for each rule, we would end up with something like $\varphi(a) \Rightarrow \varphi(b)$, which, of course, is not derivable.

Starting as usual, we write

$$\dfrac{}{\exists x\, \neg\varphi(x) \;\Rightarrow\; \neg\forall x\, \varphi(x)}$$

We could either carry out the $\exists$L rule or the $\neg$R rule. Since the $\exists$L rule is subject to the eigenvariable condition, it's a good idea to take care of it sooner rather than later, so we'll do that one first.

$$\dfrac{\dfrac{}{\neg\varphi(a) \;\Rightarrow\; \neg\forall x\, \varphi(x)}}{\exists x\, \neg\varphi(x) \;\Rightarrow\; \neg\forall x\, \varphi(x)}\,\exists\text{L}$$

Applying the $\neg$L and $\neg$R rules backwards, we get

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x\, \varphi(x) \;\Rightarrow\; \varphi(a)}}{\neg\varphi(a), \forall x\, \varphi(x) \;\Rightarrow}\,\neg\text{L}}{\forall x\, \varphi(x), \neg\varphi(a) \;\Rightarrow}\,\text{XL}}{\neg\varphi(a) \;\Rightarrow\; \neg\forall x\varphi(x)}\,\neg\text{R}}{\exists x\neg\varphi(x) \;\Rightarrow\; \neg\forall x\varphi(x)}\,\exists\text{L}$$

At this point, our only option is to carry out the $\forall$L rule. Since this rule is not subject to the eigenvariable restriction, we're in the clear. Remember, we want to try and obtain an initial sequent (of the form $\varphi(a) \Rightarrow \varphi(a)$), so we should choose a as our argument for φ when we apply the rule.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi(a) \;\Rightarrow\; \varphi(a)}}{\forall x\, \varphi(x) \;\Rightarrow\; \varphi(a)}\,\forall\text{L}}{\neg\varphi(a), \forall x\, \varphi(x) \;\Rightarrow}\,\neg\text{L}}{\forall x\, \varphi(x), \neg\varphi(a) \;\Rightarrow}\,\text{XL}}{\neg\varphi(a) \;\Rightarrow\; \neg\forall x\, \varphi(x)}\,\neg\text{R}}{\exists x\, \neg\varphi(x) \;\Rightarrow\; \neg\forall x\, \varphi(x)}\,\exists\text{L}$$

It is important, especially when dealing with quantifiers, to double check at this point that the eigenvariable condition has not been violated. Since the only rule we applied that is subject to the eigenvariable condition was ∃L, and the eigenvariable a does not occur in its lower sequent (the end-sequent), this is a correct derivation.

Problem 19.4. Give derivations of the following sequents:

1. $\Rightarrow (\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \forall z\, (\varphi(z) \land \psi(z))$.
2. $\Rightarrow (\exists x\, \varphi(x) \lor \exists y\, \psi(y)) \to \exists z\, (\varphi(z) \lor \psi(z))$.
3. $\forall x\, (\varphi(x) \to \psi) \Rightarrow \exists y\, \varphi(y) \to \psi$.
4. $\forall x\, \neg\varphi(x) \Rightarrow \neg\exists x\, \varphi(x)$.
5. $\Rightarrow \neg\exists x\, \varphi(x) \to \forall x\, \neg\varphi(x)$.
6. $\Rightarrow \neg\exists x\, \forall y\, ((\varphi(x, y) \to \neg\varphi(y, y)) \land (\neg\varphi(y, y) \to \varphi(x, y)))$.

Problem 19.5. Give derivations of the following sequents:

1. $\Rightarrow \neg\forall x\, \varphi(x) \to \exists x\, \neg\varphi(x)$.
2. $(\forall x\, \varphi(x) \to \psi) \Rightarrow \exists y\, (\varphi(y) \to \psi)$.
3. $\Rightarrow \exists x\, (\varphi(x) \to \forall y\, \varphi(y))$.

(These all require the CR rule.)

This section collects the definitions of the provability relation and consistency for natural deduction.

---

content/first-order-logic/sequent-calculus/proof-theoretic-notions.tex

19.8 Proof-Theoretic Notions

explanation

Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain sequents. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorem*.

fol:seq:ptn:sec

Definition 19.10 (Theorems). A sentence φ is a *theorem* if there is a derivation in **LK** of the sequent $\Rightarrow \varphi$. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 19.11 (Derivability). A sentence φ is *derivable from* a set of sentences Γ, $\Gamma \vdash \varphi$, iff there is a finite subset $\Gamma_0 \subseteq \Gamma$ and a sequence Γ_0' of the sentences in Γ_0 such that **LK** derives $\Gamma_0' \Rightarrow \varphi$. If φ is not derivable from Γ we write $\Gamma \nvdash \varphi$.

Because of the contraction, weakening, and exchange rules, the order and number of sentences in Γ_0' does not matter: if a sequent $\Gamma_0' \Rightarrow \varphi$ is derivable, then so is $\Gamma_0'' \Rightarrow \varphi$ for any Γ_0'' that contains the same sentences as Γ_0'. For instance, if $\Gamma_0 = \{\psi, \chi\}$ then both $\Gamma_0' = \langle \psi, \psi, \chi \rangle$ and $\Gamma_0'' = \langle \chi, \chi, \psi \rangle$ are sequences containing just the sentences in Γ_0. If a sequent containing one is derivable, so is the other, e.g.:

$$\dfrac{\dfrac{\dfrac{\vdots}{\psi, \psi, \chi \Rightarrow \varphi}\,\text{CL}}{\psi, \chi \Rightarrow \varphi}\,\text{XL}}{\chi, \psi \Rightarrow \varphi}\,\text{WL}$$

From now on we'll say that if Γ_0 is a finite set of sentences then $\Gamma_0 \Rightarrow \varphi$ is any sequent where the antecedent is a sequence of sentences in Γ_0 and tacitly include contractions, exchanges, and weakenings if necessary.

Definition 19.12 (Consistency). A set of sentences Γ is *inconsistent* iff there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow$. If Γ is not inconsistent, i.e., if for every finite $\Gamma_0 \subseteq \Gamma$, **LK** does not derive $\Gamma_0 \Rightarrow$, we say it is *consistent*.

fol:seq:ptn: prop:reflexivity

Proposition 19.13 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.*

Proof. The initial sequent $\varphi \Rightarrow \varphi$ is derivable, and $\{\varphi\} \subseteq \Gamma$. □

fol:seq:ptn: prop:monotonicity

Proposition 19.14 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.*

Proof. Suppose $\Gamma \vdash \varphi$, i.e., there is a finite $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \Rightarrow \varphi$ is derivable. Since $\Gamma \subseteq \Delta$, then Γ_0 is also a finite subset of Δ. The derivation of $\Gamma_0 \Rightarrow \varphi$ thus also shows $\Delta \vdash \varphi$. □

fol:seq:ptn: prop:transitivity

Proposition 19.15 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.*

Proof. If $\Gamma \vdash \varphi$, there is a finite $\Gamma_0 \subseteq \Gamma$ and a derivation π_0 of $\Gamma_0 \Rightarrow \varphi$. If $\{\varphi\} \cup \Delta \vdash \psi$, then for some finite subset $\Delta_0 \subseteq \Delta$, there is a derivation π_1 of $\varphi, \Delta_0 \Rightarrow \psi$. Consider the following derivation:

$$\dfrac{\begin{array}{cc}\vdots\,\pi_0 & \vdots\,\pi_1\\ \Gamma_0 \Rightarrow \varphi & \varphi, \Delta_0 \Rightarrow \psi\end{array}}{\Gamma_0, \Delta_0 \Rightarrow \psi}\,\text{Cut}$$

Since $\Gamma_0 \cup \Delta_0 \subseteq \Gamma \cup \Delta$, this shows $\Gamma \cup \Delta \vdash \psi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \ldots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

Proposition 19.16. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every sentence φ.* *fol:seq:ptn: prop:incons*

Proof. Exercise. □

Problem 19.6. Prove Proposition 19.16

Proposition 19.17 (Compactness). *fol:seq:ptn: prop:proves-compact*

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof.

1. If $\Gamma \vdash \varphi$, then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that the sequent $\Gamma_0 \Rightarrow \varphi$ has a derivation. Consequently, $\Gamma_0 \vdash \varphi$.
2. If Γ is inconsistent, there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow$. But then Γ_0 is a finite subset of Γ that is inconsistent. □

`content/first-order-logic/sequent-calculus/provability-consistency.tex`

19.9 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem. *fol:seq:prv: sec*

Proposition 19.18. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.* *fol:seq:prv: prop:provability-contr*

Proof. There are finite Γ_0 and $\Gamma_1 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow \varphi$ and $\varphi, \Gamma_1 \Rightarrow$. Let the **LK**-derivation of $\Gamma_0 \Rightarrow \varphi$ be π_0 and the **LK**-derivation of $\Gamma_1, \varphi \Rightarrow$ be π_1. We can then derive

$$\dfrac{\begin{array}{cc}\vdots\,\pi_0 & \vdots\,\pi_1\\ \Gamma_0 \Rightarrow \varphi & \varphi, \Gamma_1 \Rightarrow\end{array}}{\Gamma_0, \Gamma_1 \Rightarrow}\,\text{Cut}$$

Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$, hence Γ is inconsistent. □

fol:seq:prv: prop:prov-incons

Proposition 19.19. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.*

Proof. First suppose $\Gamma \vdash \varphi$, i.e., there is a derivation π_0 of $\Gamma \Rightarrow \varphi$. By adding a $\neg$L rule, we obtain a derivation of $\neg\varphi, \Gamma \Rightarrow$, i.e., $\Gamma \cup \{\neg\varphi\}$ is inconsistent.

If $\Gamma \cup \{\neg\varphi\}$ is inconsistent, there is a derivation π_1 of $\neg\varphi, \Gamma \Rightarrow$. The following is a derivation of $\Gamma \Rightarrow \varphi$:

$$\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg\varphi}\,\neg\mathrm{R} \qquad \begin{array}{c}\vdots\\ \pi_1\end{array} \quad \neg\varphi, \Gamma \Rightarrow}{\Gamma \Rightarrow \varphi}\,\mathrm{Cut}$$

□

Problem 19.7. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

fol:seq:prv: prop:explicit-inc

Proposition 19.20. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

Proof. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there is a derivation π of a sequent $\Gamma_0 \Rightarrow \varphi$. The sequent $\neg\varphi, \Gamma_0 \Rightarrow$ is also derivable:

$$\dfrac{\begin{array}{c}\vdots\\ \pi\end{array} \qquad \dfrac{\dfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow}\,\neg\mathrm{L}}{\varphi, \neg\varphi \Rightarrow}\,\mathrm{XL}}{\Gamma_0, \neg\varphi \Rightarrow}\,\mathrm{Cut}$$

Since $\neg\varphi \in \Gamma$ and $\Gamma_0 \subseteq \Gamma$, this shows that Γ is inconsistent. □

fol:seq:prv: prop:provability-exhaustive

Proposition 19.21. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

Proof. There are finite sets $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$ and **LK**-derivations π_0 and π_1 of $\varphi, \Gamma_0 \Rightarrow$ and $\neg\varphi, \Gamma_1 \Rightarrow$, respectively. We can then derive

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \pi_0\end{array}}{\varphi, \Gamma_0 \Rightarrow}}{\Gamma_0 \Rightarrow \neg\varphi}\,\neg\mathrm{R} \qquad \begin{array}{c}\vdots\\ \pi_1\end{array} \quad \neg\varphi, \Gamma_1 \Rightarrow}{\Gamma_0, \Gamma_1 \Rightarrow}\,\mathrm{Cut}$$

Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$. Hence Γ is inconsistent. □

19.10 Derivability and the Propositional Connectives

explanation We establish that the derivability relation $\vdash$ of the sequent calculus is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. fol:seq:ppr:sec

Proposition 19.22. fol:seq:ppr:prop:provability-land fol:seq:ppr:prop:provability-land-left fol:seq:ppr:prop:provability-land-right

1. *Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$.*
2. *$\varphi, \psi \vdash \varphi \wedge \psi$.*

Proof. 1. Both sequents $\varphi \wedge \psi \Rightarrow \varphi$ and $\varphi \wedge \psi \Rightarrow \psi$ are derivable:

$$\dfrac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi} \wedge\mathrm{L} \qquad \dfrac{\psi \Rightarrow \psi}{\varphi \wedge \psi \Rightarrow \psi} \wedge\mathrm{L}$$

2. Here is a derivation of the sequent $\varphi, \psi \Rightarrow \varphi \wedge \psi$:

$$\dfrac{\varphi \Rightarrow \varphi \qquad \psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \varphi \wedge \psi} \wedge\mathrm{R}$$

□

Proposition 19.23. fol:seq:ppr:prop:provability-lor

1. *$\varphi \vee \psi, \neg\varphi, \neg\psi$ is inconsistent.*
2. *Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.*

Proof. 1. We give a derivation of the sequent $\varphi \vee \psi, \neg\varphi, \neg\psi \Rightarrow$:

$$\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow}\neg\mathrm{L}}{\varphi, \neg\varphi, \neg\psi \Rightarrow} \qquad \dfrac{\dfrac{\psi \Rightarrow \psi}{\neg\psi, \psi \Rightarrow}\neg\mathrm{L}}{\psi, \neg\varphi, \neg\psi \Rightarrow}}{\varphi \vee \psi, \neg\varphi, \neg\psi \Rightarrow} \vee\mathrm{L}$$

(Recall that double inference lines indicate several weakening, contraction, and exchange inferences.)

2. Both sequents $\varphi \Rightarrow \varphi \vee \psi$ and $\psi \Rightarrow \varphi \vee \psi$ have derivations:

$$\dfrac{\varphi \Rightarrow \varphi}{\varphi \Rightarrow \varphi \vee \psi} \vee\mathrm{R} \qquad \dfrac{\psi \Rightarrow \psi}{\psi \Rightarrow \varphi \vee \psi} \vee\mathrm{R}$$

□

Proposition 19.24. fol:seq:ppr:prop:provability-lif fol:seq:ppr:prop:provability-lif-left

1. *$\varphi, \varphi \to \psi \vdash \psi$.*

fol:seq:ppr: prop:provability-lif-right

2. *Both* $\neg\varphi \vdash \varphi \to \psi$ *and* $\psi \vdash \varphi \to \psi$.

Proof. 1. The sequent $\varphi \to \psi, \varphi \Rightarrow \psi$ is derivable:

$$\dfrac{\varphi \Rightarrow \varphi \qquad \psi \Rightarrow \psi}{\varphi \to \psi, \varphi \Rightarrow \psi} \to\text{L}$$

2. Both sequents $\neg\varphi \Rightarrow \varphi \to \psi$ and $\psi \Rightarrow \varphi \to \psi$ are derivable:

$$\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow}\neg\text{L}}{\varphi, \neg\varphi \Rightarrow}\text{XL}}{\varphi, \neg\varphi \Rightarrow \psi}\text{WR}}{\neg\varphi \Rightarrow \varphi \to \psi} \to\text{R} \qquad \dfrac{\dfrac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi}\text{WL}}{\psi \Rightarrow \varphi \to \psi} \to\text{R}$$

□

---

content/first-order-logic/sequent-calculus/provability-quantifiers.tex

19.11 Derivability and the Quantifiers

fol:seq:qpr: sec

The completeness theorem also requires that the sequent calculus rules yield the facts about $\vdash$ established in this section. explanation

fol:seq:qpr: thm:strong-generalization

Theorem 19.25. *If c is a constant not occurring in Γ or $\varphi(x)$ and $\Gamma \vdash \varphi(c)$, then $\Gamma \vdash \forall x\, \varphi(x)$.*

Proof. Let π_0 be an **LK**-derivation of $\Gamma_0 \Rightarrow \varphi(c)$ for some finite $\Gamma_0 \subseteq \Gamma$. By adding a $\forall$R inference, we obtain a derivation of $\Gamma_0 \Rightarrow \forall x\, \varphi(x)$, since c does not occur in Γ or $\varphi(x)$ and thus the eigenvariable condition is satisfied. □

fol:seq:qpr: prop:provability-quantifiers

Proposition 19.26.

1. $\varphi(t) \vdash \exists x\, \varphi(x)$.
2. $\forall x\, \varphi(x) \vdash \varphi(t)$.

Proof. 1. The sequent $\varphi(t) \Rightarrow \exists x\, \varphi(x)$ is derivable:

$$\dfrac{\varphi(t) \Rightarrow \varphi(t)}{\varphi(t) \Rightarrow \exists x\, \varphi(x)} \exists\text{R}$$

2. The sequent $\forall x\, \varphi(x) \Rightarrow \varphi(t)$ is derivable:

$$\dfrac{\varphi(t) \Rightarrow \varphi(t)}{\forall x\, \varphi(x) \Rightarrow \varphi(t)} \forall\text{L}$$

□

---

content/first-order-logic/sequent-calculus/soundness.tex

19.12 Soundness

explanation

A derivation system, such as the sequent calculus, is *sound* if it cannot derive things that do not actually hold. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that

fol:seq:soundsec

1. every derivable φ is valid;
2. if a sentence is derivable from some others, it is also a consequence of them;
3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Because all these proof-theoretic properties are defined via derivability in the sequent calculus of certain sequents, proving (1)–(3) above requires proving something about the semantic properties of derivable sequents. We will first define what it means for a sequent to be *valid*, and then show that every derivable sequent is valid. (1)–(3) then follow as corollaries from this result.

Definition 19.27. A structure $\mathfrak{M}$ *satisfies* a sequent $\Gamma \Rightarrow \Delta$ iff either $\mathfrak{M} \nvDash \varphi$ for some $\varphi \in \Gamma$ or $\mathfrak{M} \vDash \varphi$ for some $\varphi \in \Delta$.

A sequent is *valid* iff every structure $\mathfrak{M}$ satisfies it.

Theorem 19.28 (Soundness). *If* **LK** *derives* $\Theta \Rightarrow \Xi$*, then* $\Theta \Rightarrow \Xi$ *is valid.*

fol:seq:soundness:thm:sequent-soundness

Proof. Let π be a derivation of $\Theta \Rightarrow \Xi$. We proceed by induction on the number of inferences n in π.

If the number of inferences is 0, then π consists only of an initial sequent. Every initial sequent $\varphi \Rightarrow \varphi$ is obviously valid, since for every $\mathfrak{M}$, either $\mathfrak{M} \nvDash \varphi$ or $\mathfrak{M} \vDash \varphi$.

If the number of inferences is greater than 0, we distinguish cases according to the type of the lowermost inference. By induction hypothesis, we can assume that the premises of that inference are valid, since the number of inferences in the derivation of any premise is smaller than n.

First, we consider the possible inferences with only one premise.

1. The last inference is a weakening. Then $\Theta \Rightarrow \Xi$ is either $\varphi, \Gamma \Rightarrow \Delta$ (if the last inference is WL) or $\Gamma \Rightarrow \Delta, \varphi$ (if it's WR), and the derivation ends in one of

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta\end{array}}{\varphi, \Gamma \Rightarrow \Delta}\,\mathrm{WL} \qquad \dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta\end{array}}{\Gamma \Rightarrow \Delta, \varphi}\,\mathrm{WR}$$

By induction hypothesis, $\Gamma \Rightarrow \Delta$ is valid, i.e., for every structure $\mathfrak{M}$, either there is some $\chi \in \Gamma$ such that $\mathfrak{M} \nvDash \chi$ or there is some $\chi \in \Delta$ such that $\mathfrak{M} \vDash \chi$.

If $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, then $\chi \in \Theta$ as well since $\Theta = \varphi, \Gamma$, and so $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Theta$. Similarly, if $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$, as $\chi \in \Xi$, $\mathfrak{M} \vDash \chi$ for some $\chi \in \Xi$. Consequently, $\Theta \Rightarrow \Xi$ is valid.

2. The last inference is ¬L: Then the premise of the last inference is $\Gamma \Rightarrow \Delta, \varphi$ and the conclusion is $\neg\varphi, \Gamma \Rightarrow \Delta$, i.e., the derivation ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \varphi\end{array}}{\neg\varphi, \Gamma \Rightarrow \Delta}\,\neg\mathrm{L}$$

and $\Theta = \neg\varphi, \Gamma$ while $\Xi = \Delta$.

The induction hypothesis tells us that $\Gamma \Rightarrow \Delta, \varphi$ is valid, i.e., for every $\mathfrak{M}$, either (a) for some $\chi \in \Gamma$, $\mathfrak{M} \nvDash \chi$, or (b) for some $\chi \in \Delta$, $\mathfrak{M} \vDash \chi$, or (c) $\mathfrak{M} \vDash \varphi$. We want to show that $\Theta \Rightarrow \Xi$ is also valid. Let $\mathfrak{M}$ be a structure. If (a) holds, then there is $\chi \in \Gamma$ so that $\mathfrak{M} \nvDash \chi$, but $\chi \in \Theta$ as well. If (b) holds, there is $\chi \in \Delta$ such that $\mathfrak{M} \vDash \chi$, but $\chi \in \Xi$ as well. Finally, if $\mathfrak{M} \vDash \varphi$, then $\mathfrak{M} \nvDash \neg\varphi$. Since $\neg\varphi \in \Theta$, there is $\chi \in \Theta$ such that $\mathfrak{M} \nvDash \chi$. Consequently, $\Theta \Rightarrow \Xi$ is valid.

3. The last inference is ¬R: Exercise.

4. The last inference is ∧L: There are two variants: $\varphi \wedge \psi$ may be inferred on the left from φ or from ψ on the left side of the premise. In the first case, the π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \varphi, \Gamma \Rightarrow \Delta\end{array}}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\,\wedge\mathrm{L}$$

and $\Theta = \varphi \wedge \psi, \Gamma$ while $\Xi = \Delta$. Consider a structure $\mathfrak{M}$. Since by induction hypothesis, $\varphi, \Gamma \Rightarrow \Delta$ is valid, (a) $\mathfrak{M} \nvDash \varphi$, (b) $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, or (c) $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$. In case (a), $\mathfrak{M} \nvDash \varphi \wedge \psi$, so there is $\chi \in \Theta$ (namely, $\varphi \wedge \psi$) such that $\mathfrak{M} \nvDash \chi$. In case (b), there is $\chi \in \Gamma$ such that $\mathfrak{M} \nvDash \chi$, and $\chi \in \Theta$ as well. In case (c), there is $\chi \in \Delta$ such that $\mathfrak{M} \vDash \chi$, and $\chi \in \Xi$ as well since $\Xi = \Delta$. So in each case, $\mathfrak{M}$ satisfies $\varphi \wedge \psi, \Gamma \Rightarrow \Delta$. Since $\mathfrak{M}$ was arbitrary, $\Gamma \Rightarrow \Delta$ is valid. The case where $\varphi \wedge \psi$ is inferred from ψ is handled the same, changing φ to ψ.

5. The last inference is $\vee$R: There are two variants: $\varphi \vee \psi$ may be inferred on the right from φ or from ψ on the right side of the premise. In the first case, π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \varphi\end{array}}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee\text{R}$$

Now $\Theta = \Gamma$ and $\Xi = \Delta, \varphi \vee \psi$. Consider a structure $\mathfrak{M}$. Since $\Gamma \Rightarrow \Delta, \varphi$ is valid, (a) $\mathfrak{M} \vDash \varphi$, (b) $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, or (c) $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$. In case (a), $\mathfrak{M} \vDash \varphi \vee \psi$. In case (b), there is $\chi \in \Gamma$ such that $\mathfrak{M} \nvDash \chi$. In case (c), there is $\chi \in \Delta$ such that $\mathfrak{M} \vDash \chi$. So in each case, $\mathfrak{M}$ satisfies $\Gamma \Rightarrow \Delta, \varphi \vee \psi$, i.e., $\Theta \Rightarrow \Xi$. Since $\mathfrak{M}$ was arbitrary, $\Theta \Rightarrow \Xi$ is valid. The case where $\varphi \vee \psi$ is inferred from ψ is handled the same, changing φ to ψ.

6. The last inference is $\to$R: Then π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \varphi, \Gamma \Rightarrow \Delta, \psi\end{array}}{\Gamma \Rightarrow \Delta, \varphi \to \psi} \to\text{R}$$

Again, the induction hypothesis says that the premise is valid; we want to show that the conclusion is valid as well. Let $\mathfrak{M}$ be arbitrary. Since $\varphi, \Gamma \Rightarrow \Delta, \psi$ is valid, at least one of the following cases obtains: (a) $\mathfrak{M} \nvDash \varphi$, (b) $\mathfrak{M} \vDash \psi$, (c) $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, or (d) $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$. In cases (a) and (b), $\mathfrak{M} \vDash \varphi \to \psi$ and so there is a $\chi \in \Delta, \varphi \to \psi$ such that $\mathfrak{M} \vDash \chi$. In case (c), for some $\chi \in \Gamma$, $\mathfrak{M} \nvDash \chi$. In case (d), for some $\chi \in \Delta$, $\mathfrak{M} \vDash \chi$. In each case, $\mathfrak{M}$ satisfies $\Gamma \Rightarrow \Delta, \varphi \to \psi$. Since $\mathfrak{M}$ was arbitrary, $\Gamma \Rightarrow \Delta, \varphi \to \psi$ is valid.

7. The last inference is $\forall$L: Then there is a formula $\varphi(x)$ and a closed term t such that π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \varphi(t), \Gamma \Rightarrow \Delta\end{array}}{\forall x\, \varphi(x), \Gamma \Rightarrow \Delta} \forall\text{L}$$

We want to show that the conclusion $\forall x\, \varphi(x), \Gamma \Rightarrow \Delta$ is valid. Consider a structure $\mathfrak{M}$. Since the premise $\varphi(t), \Gamma \Rightarrow \Delta$ is valid, (a) $\mathfrak{M} \nvDash \varphi(t)$, (b) $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, or (c) $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$. In case (a), by Proposition 16.31, if $\mathfrak{M} \vDash \forall x\, \varphi(x)$, then $\mathfrak{M} \vDash \varphi(t)$. Since $\mathfrak{M} \nvDash \varphi(t)$,

$\mathfrak{M} \nvDash \forall x\, \varphi(x)$. In case (b) and (c), $\mathfrak{M}$ also satisfies $\forall x\, \varphi(x), \Gamma \Rightarrow \Delta$. Since $\mathfrak{M}$ was arbitrary, $\forall x\, \varphi(x), \Gamma \Rightarrow \Delta$ is valid.

8. The last inference is ∃R: Exercise.

9. The last inference is ∀R: Then there is a formula $\varphi(x)$ and a constant symbol a such that π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \varphi(a)\end{array}}{\Gamma \Rightarrow \Delta, \forall x\, \varphi(x)}\,\forall\mathrm{R}$$

where the eigenvariable condition is satisfied, i.e., a does not occur in $\varphi(x)$, Γ, or Δ. By induction hypothesis, the premise of the last inference is valid. We have to show that the conclusion is valid as well, i.e., that for any structure $\mathfrak{M}$, (a) $\mathfrak{M} \vDash \forall x\, \varphi(x)$, (b) $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, or (c) $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$.

Suppose $\mathfrak{M}$ is an arbitrary structure. If (b) or (c) holds, we are done, so suppose neither holds: for all $\chi \in \Gamma$, $\mathfrak{M} \vDash \chi$, and for all $\chi \in \Delta$, $\mathfrak{M} \nvDash \chi$. We have to show that (a) holds, i.e., $\mathfrak{M} \vDash \forall x\, \varphi(x)$. By Proposition 16.19, if suffices to show that $\mathfrak{M}, s \vDash \varphi(x)$ for all variable assignments s. So let s be an arbitrary variable assignment. Consider the structure $\mathfrak{M}'$ which is just like $\mathfrak{M}$ except $a^{\mathfrak{M}'} = s(x)$. By Corollary 16.21, for any $\chi \in \Gamma$, $\mathfrak{M}' \vDash \chi$ since a does not occur in Γ, and for any $\chi \in \Delta$, $\mathfrak{M}' \nvDash \chi$. But the premise is valid, so $\mathfrak{M}' \vDash \varphi(a)$. By Proposition 16.18, $\mathfrak{M}', s \vDash \varphi(a)$, since $\varphi(a)$ is a sentence. Now $s \sim_x s$ with $s(x) = \mathrm{Val}_s^{\mathfrak{M}'}(a)$, since we've defined $\mathfrak{M}'$ in just this way. So Proposition 16.23 applies, and we get $\mathfrak{M}', s \vDash \varphi(x)$. Since a does not occur in $\varphi(x)$, by Proposition 16.20, $\mathfrak{M}, s \vDash \varphi(x)$. Since s was arbitrary, we've completed the proof that $\mathfrak{M}, s \vDash \varphi(x)$ for all variable assignments.

10. The last inference is ∃L: Exercise.

Now let's consider the possible inferences with two premises.

1. The last inference is a cut: then π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \varphi\end{array} \qquad \begin{array}{c}\vdots\\ \varphi, \Pi \Rightarrow \Lambda\end{array}}{\Gamma, \Pi \Rightarrow \Delta, \Lambda}\,\mathrm{Cut}$$

Let $\mathfrak{M}$ be a structure. By induction hypothesis, the premises are valid, so $\mathfrak{M}$ satisfies both premises. We distinguish two cases: (a) $\mathfrak{M} \nvDash \varphi$ and (b) $\mathfrak{M} \vDash \varphi$. In case (a), in order for $\mathfrak{M}$ to satisfy the left premise, it must

satisfy $\Gamma \Rightarrow \Delta$. But then it also satisfies the conclusion. In case (b), in order for $\mathfrak{M}$ to satisfy the right premise, it must satisfy $\Pi \setminus \Lambda$. Again, $\mathfrak{M}$ satisfies the conclusion.

2. The last inference is $\wedge$R. Then π ends in

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma \Rightarrow \Delta, \varphi & \Gamma \Rightarrow \Delta, \psi\end{array}}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \wedge\text{R}$$

Consider a structure $\mathfrak{M}$. If $\mathfrak{M}$ satisfies $\Gamma \Rightarrow \Delta$, we are done. So suppose it doesn't. Since $\Gamma \Rightarrow \Delta, \varphi$ is valid by induction hypothesis, $\mathfrak{M} \vDash \varphi$. Similarly, since $\Gamma \Rightarrow \Delta, \psi$ is valid, $\mathfrak{M} \vDash \psi$. But then $\mathfrak{M} \vDash \varphi \wedge \psi$.

3. The last inference is $\vee$L: Exercise.

4. The last inference is $\rightarrow$L. Then π ends in

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma \Rightarrow \Delta, \varphi & \psi, \Pi \Rightarrow \Lambda\end{array}}{\varphi \rightarrow \psi, \Gamma, \Pi \Rightarrow \Delta, \Lambda} \rightarrow\text{L}$$

Again, consider a structure $\mathfrak{M}$ and suppose $\mathfrak{M}$ doesn't satisfy $\Gamma, \Pi \Rightarrow \Delta, \Lambda$. We have to show that $\mathfrak{M} \nvDash \varphi \rightarrow \psi$. If $\mathfrak{M}$ doesn't satisfy $\Gamma, \Pi \Rightarrow \Delta, \Lambda$, it satisfies neither $\Gamma \Rightarrow \Delta$ nor $\Pi \Rightarrow \Lambda$. Since, $\Gamma \Rightarrow \Delta, \varphi$ is valid, we have $\mathfrak{M} \vDash \varphi$. Since $\psi, \Pi \Rightarrow \Lambda$ is valid, we have $\mathfrak{M} \nvDash \psi$. But then $\mathfrak{M} \nvDash \varphi \rightarrow \psi$, which is what we wanted to show. □

Problem 19.8. Complete the proof of Theorem 19.28.

Corollary 19.29. *If $\vdash \varphi$ then φ is valid.*

fol:seq:sou: cor:weak-soundness

Corollary 19.30. *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.*

fol:seq:sou: cor:entailment-soundness

Proof. If $\Gamma \vdash \varphi$ then for some finite subset $\Gamma_0 \subseteq \Gamma$, there is a derivation of $\Gamma_0 \Rightarrow \varphi$. By Theorem 19.28, every structure $\mathfrak{M}$ either makes some $\psi \in \Gamma_0$ false or makes φ true. Hence, if $\mathfrak{M} \vDash \Gamma$ then also $\mathfrak{M} \vDash \varphi$. □

Corollary 19.31. *If Γ is satisfiable, then it is consistent.*

fol:seq:sou: cor:consistency-soundness

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then there is a finite $\Gamma_0 \subseteq \Gamma$ and a derivation of $\Gamma_0 \Rightarrow \quad$. By Theorem 19.28, $\Gamma_0 \Rightarrow$ is valid. In other words, for every structure $\mathfrak{M}$, there is $\chi \in \Gamma_0$ so that $\mathfrak{M} \nvDash \chi$, and since $\Gamma_0 \subseteq \Gamma$, that χ is also in Γ. Thus, no $\mathfrak{M}$ satisfies Γ, and Γ is not satisfiable. □

content/first-order-logic/sequent-calculus/identity.tex

19.13 Derivations with Identity predicate

fol:seq:ide:sec

Derivations with identity predicate require additional initial sequents and inference rules.

Definition 19.32 (Initial sequents for $=$). If t is a closed term, then $\Rightarrow t = t$ is an initial sequent.

The rules for $=$ are (t_1 and t_2 are closed terms):

$$\frac{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_1)}{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_2)} = \qquad \frac{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_2)}{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_1)} =$$

Example 19.33. If s and t are closed terms, then $s = t, \varphi(s) \vdash \varphi(t)$:

$$\dfrac{\dfrac{\varphi(s) \Rightarrow \varphi(s)}{s = t, \varphi(s) \Rightarrow \varphi(s)}\,\mathrm{WL}}{s = t, \varphi(s) \Rightarrow \varphi(t)}\,=$$

This may be familiar as the principle of substitutability of identicals, or Leibniz' Law.

LK proves that $=$ is symmetric and transitive:

$$\dfrac{\dfrac{\Rightarrow t_1 = t_1}{t_1 = t_2 \Rightarrow t_1 = t_1}\,\mathrm{WL}}{t_1 = t_2 \Rightarrow t_2 = t_1}\,= \qquad \dfrac{\dfrac{\dfrac{t_1 = t_2 \Rightarrow t_1 = t_2}{t_2 = t_3, t_1 = t_2 \Rightarrow t_1 = t_2}\,\mathrm{WL}}{t_2 = t_3, t_1 = t_2 \Rightarrow t_1 = t_3}\,=}{t_1 = t_2, t_2 = t_3 \Rightarrow t_1 = t_3}\,\mathrm{XL}$$

In the derivation on the left, the formula $x = t_1$ is our $\varphi(x)$. On the right, we take $\varphi(x)$ to be $t_1 = x$.

Problem 19.9. Give derivations of the following sequents:

1. $\Rightarrow \forall x \forall y \, ((x = y \wedge \varphi(x)) \rightarrow \varphi(y))$
2. $\exists x \, \varphi(x) \wedge \forall y \forall z \, ((\varphi(y) \wedge \varphi(z)) \rightarrow y = z) \Rightarrow \exists x \, (\varphi(x) \wedge \forall y \, (\varphi(y) \rightarrow y = x))$

---

content/first-order-logic/sequent-calculus/soundness-identity.tex

19.14 Soundness with Identity predicate

fol:seq:sid:sec

Proposition 19.34. LK *with initial sequents and rules for identity is sound.*

Proof. Initial sequents of the form $\Rightarrow t = t$ are valid, since for every structure $\mathfrak{M}$, $\mathfrak{M} \vDash t = t$. (Note that we assume the term t to be closed, i.e., it contains no variables, so variable assignments are irrelevant).

Suppose the last inference in a derivation is $=$. Then the premise is $t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_1)$ and the conclusion is $t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_2)$. Consider a structure $\mathfrak{M}$. We need to show that the conclusion is valid, i.e., if $\mathfrak{M} \vDash t_1 = t_2$ and $\mathfrak{M} \vDash \Gamma$, then either $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$ or $\mathfrak{M} \vDash \varphi(t_2)$.

By induction hypothesis, the premise is valid. This means that if $\mathfrak{M} \vDash t_1 = t_2$ and $\mathfrak{M} \vDash \Gamma$ either (a) for some $\chi \in \Delta$, $\mathfrak{M} \vDash \chi$ or (b) $\mathfrak{M} \vDash \varphi(t_1)$. In case (a) we are done. Consider case (b). Let s be a variable assignment with $s(x) = \mathrm{Val}^{\mathfrak{M}}(t_1)$. By Proposition 16.18, $\mathfrak{M}, s \vDash \varphi(t_1)$. Since $s \sim_x s$, by Proposition 16.23, $\mathfrak{M}, s \vDash \varphi(x)$. since $\mathfrak{M} \vDash t_1 = t_2$, we have $\mathrm{Val}^{\mathfrak{M}}(t_1) = \mathrm{Val}^{\mathfrak{M}}(t_2)$, and hence $s(x) = \mathrm{Val}^{\mathfrak{M}}(t_2)$. By applying Proposition 16.23 again, we also have $\mathfrak{M}, s \vDash \varphi(t_2)$. By Proposition 16.18, $\mathfrak{M} \vDash \varphi(t_2)$. □

Chapter 20

Natural Deduction

This chapter presents a natural deduction system in the style of Gentzen/Prawitz.

To include or exclude material relevant to natural deduction as a proof system, use the "prfND" tag.

---

content/first-order-logic/natural-deduction/rules-and-proofs.tex

20.1 Rules and Derivations

explanation

Natural deduction systems are meant to closely parallel the informal reasoning used in mathematical proof (hence it is somewhat "natural"). Natural deduction proofs begin with assumptions. Inference rules are then applied. Assumptions are "discharged" by the ¬Intro, →Intro, ∨Elim and ∃Elim inference rules, and the label of the discharged assumption is placed beside the inference for clarity.

fol:ntd:rul:sec

Definition 20.1 (Assumption). An *assumption* is any sentence in the topmost position of any branch.

Derivations in natural deduction are certain trees of sentences, where the topmost sentences are assumptions, and if a sentence stands below one, two, or three other sequents, it must follow correctly by a rule of inference. The sentences at the top of the inference are called the *premises* and the sentence below the *conclusion* of the inference. The rules come in pairs, an introduction and an elimination rule for each logical operator. They introduce a logical operator in the conclusion or remove a logical operator from a premise of the rule. Some of the rules allow an assumption of a certain type to be *discharged*. To indicate which assumption is discharged by which inference, we also assign labels to both the assumption and the inference. This is indicated by writing the assumption as "$[\varphi]^n$."

It is customary to consider rules for all the logical operators $\wedge$, $\vee$, $\to$, $\neg$, and $\bot$, even if some of those are defined.

---

content/first-order-logic/natural-deduction/propositional-rules.tex

20.2 Propositional Rules

fol:ntd:prl:sec

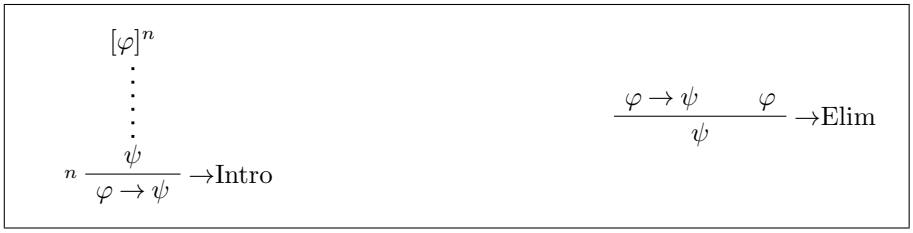
Rules for $\wedge$

$$\frac{\varphi \qquad \psi}{\varphi \wedge \psi}\,\wedge\text{Intro} \qquad \frac{\varphi \wedge \psi}{\varphi}\,\wedge\text{Elim} \qquad \frac{\varphi \wedge \psi}{\psi}\,\wedge\text{Elim}$$

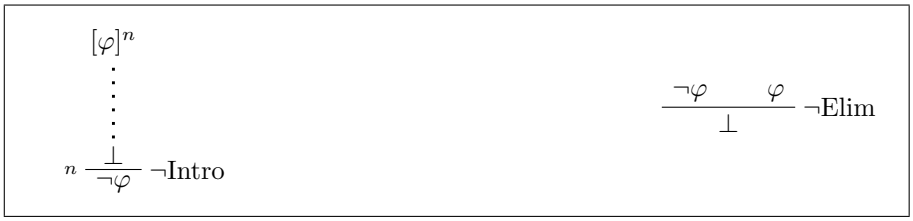
Rules for $\vee$

$$\frac{\varphi}{\varphi \vee \psi}\,\vee\text{Intro} \qquad \frac{\psi}{\varphi \vee \psi}\,\vee\text{Intro} \qquad {}_n\frac{\varphi \vee \psi \qquad \begin{array}{c}[\varphi]^n\\ \vdots\\ \chi\end{array} \qquad \begin{array}{c}[\psi]^n\\ \vdots\\ \chi\end{array}}{\chi}\,\vee\text{Elim}$$

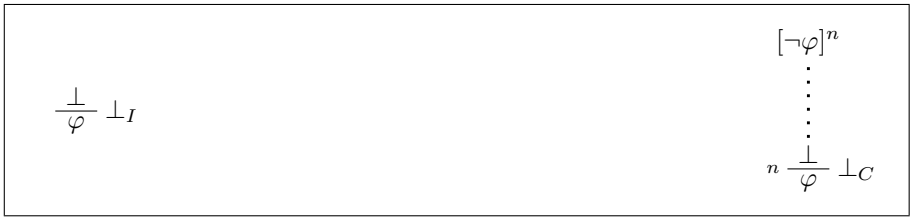
Rules for $\to$



Rules for ¬



Rules for ⊥



Note that ¬Intro and $\bot_C$ are very similar: The difference is that ¬Intro derives a negated sentence $\neg\varphi$ but $\bot_C$ a positive sentence φ.

Whenever a rule indicates that some assumption may be discharged, we take this to be a permission, but not a requirement. E.g., in the →Intro rule, we may discharge any number of assumptions of the form φ in the derivation of the premise ψ, including zero.

---

content/first-order-logic/natural-deduction/quantifier-rules.tex

20.3 Quantifier Rules

Rules for ∀

fol:ntd:qrl:sec

In the rules for $\forall$, t is a closed term (a term that does not contain any variables), and a is a constant symbol which does not occur in the conclusion $\forall x\, \varphi(x)$, or in any assumption which is undischarged in the derivation ending with the premise $\varphi(a)$. We call a the *eigenvariable* of the $\forall$Intro inference.[1]

Rules for $\exists$

$$\frac{\varphi(t)}{\exists x\, \varphi(x)}\ \exists\text{Intro} \qquad\qquad \begin{array}{c} [\varphi(a)]^n \\ \vdots \\ \exists x\, \varphi(x) \quad \chi \\ \hline \chi \end{array}\ n\ \exists\text{Elim}$$

Again, t is a closed term, and a is a constant symbol which does not occur in the premise $\exists x\, \varphi(x)$, in the conclusion χ, or any assumption which is undischarged in the derivations ending with the two premises (other than the assumptions $\varphi(a)$). We call a the *eigenvariable* of the $\exists$Elim inference.

The condition that an eigenvariable neither occur in the premises nor in any assumption that is undischarged in the derivations leading to the premises for the $\forall$Intro or $\exists$Elim inference is called the *eigenvariable condition*.

Recall the convention that when φ is a formula with the variable x free, we indicate this by writing $\varphi(x)$. In the same context, $\varphi(t)$ then is short for $\varphi[t/x]$. So we could also write the $\exists$Intro rule as: explanation

$$\frac{\varphi[t/x]}{\exists x\, \varphi}\ \exists\text{Intro}$$

Note that t may already occur in φ, e.g., φ might be $P(t, x)$. Thus, inferring $\exists x\, P(t, x)$ from $P(t, t)$ is a correct application of $\exists$Intro—you may "replace" one or more, and not necessarily all, occurrences of t in the premise by the bound variable x. However, the eigenvariable conditions in $\forall$Intro and $\exists$Elim require that the constant symbol a does not occur in φ. So, you cannot correctly infer $\forall x\, P(a, x)$ from $P(a, a)$ using $\forall$Intro.

In $\exists$Intro and $\forall$Elim there are no restrictions, and the term t can be anything, so we do not have to worry about any conditions. On the other hand, in the $\exists$Elim and $\forall$Intro rules, the eigenvariable condition requires that the constant symbol a does not occur anywhere in the conclusion or in an undischarged assumption. The condition is necessary to ensure that the system is sound, i.e., only derives sentences from undischarged assumptions from which they follow. Without this condition, the following would be allowed: explanation

[1] We use the term "eigenvariable" even though a in the above rule is a constant. This has historical reasons.

$$\dfrac{\exists x\, \varphi(x) \qquad \dfrac{[\varphi(a)]^1}{\forall x\, \varphi(x)}\,{*}\forall\text{Intro}}{\forall x\, \varphi(x)}\,\exists\text{Elim}$$

However, $\exists x\, \varphi(x) \not\models \forall x\, \varphi(x)$.

As the elimination rules for quantifiers only allow substituting closed terms for variables, it follows that any formula that can be derived from a set of sentences is itself a sentence.

---

content/first-order-logic/natural-deduction/derivations.tex

20.4 Derivations

explanation

We've said what an assumption is, and we've given the rules of inference. Derivations in natural deduction are inductively generated from these: each derivation either is an assumption on its own, or consists of one, two, or three derivations followed by a correct inference.

fol:ntd:der:sec

Definition 20.2 (Derivation). A *derivation* of a sentence φ from assumptions Γ is a finite tree of sentences satisfying the following conditions:

1. The topmost sentences of the tree are either in Γ or are discharged by an inference in the tree.
2. The bottommost sentence of the tree is φ.
3. Every sentence in the tree except the sentence φ at the bottom is a premise of a correct application of an inference rule whose conclusion stands directly below that sentence in the tree.

We then say that φ is the *conclusion* of the derivation and Γ its undischarged assumptions.

If a derivation of φ from Γ exists, we say that φ is *derivable* from Γ, or in symbols: $\Gamma \vdash \varphi$. If there is a derivation of φ in which every assumption is discharged, we write $\vdash \varphi$.

Example 20.3. Every assumption on its own is a derivation. So, e.g., φ by itself is a derivation, and so is ψ by itself. We can obtain a new derivation from these by applying, say, the $\wedge$Intro rule,

$$\dfrac{\varphi \qquad \psi}{\varphi \wedge \psi}\,\wedge\text{Intro}$$

These rules are meant to be general: we can replace the φ and ψ in it with any sentences, e.g., by χ and θ. Then the conclusion would be $\chi \wedge \theta$, and so

$$\dfrac{\chi \qquad \theta}{\chi \wedge \theta}\,\wedge\text{Intro}$$

is a correct derivation. Of course, we can also switch the assumptions, so that θ plays the role of φ and χ that of ψ. Thus,

$$\dfrac{\theta \qquad \chi}{\theta \wedge \chi}\,\wedge\text{Intro}$$

is also a correct derivation.

We can now apply another rule, say, $\rightarrow$Intro, which allows us to conclude a conditional and allows us to discharge any assumption that is identical to the antecedent of that conditional. So both of the following would be correct derivations:

$$1\,\dfrac{\dfrac{[\chi]^1 \qquad \theta}{\chi \wedge \theta}\,\wedge\text{Intro}}{\chi \rightarrow (\chi \wedge \theta)}\,\rightarrow\text{Intro} \qquad 1\,\dfrac{\dfrac{\chi \qquad [\theta]^1}{\chi \wedge \theta}\,\wedge\text{Intro}}{\theta \rightarrow (\chi \wedge \theta)}\,\rightarrow\text{Intro}$$

They show, respectively, that $\theta \vdash \chi \rightarrow (\chi \wedge \theta)$ and $\chi \vdash \theta \rightarrow (\chi \wedge \theta)$.

Remember that discharging of assumptions is a permission, not a requirement: we don't have to discharge the assumptions. In particular, we can apply a rule even if the assumptions are not present in the derivation. For instance, the following is legal, even though there is no assumption φ to be discharged:

$$1\,\dfrac{\psi}{\varphi \rightarrow \psi}\,\rightarrow\text{Intro}$$

---

content/first-order-logic/natural-deduction/proving-things.tex

20.5 Examples of Derivations

fol:ntd:pro:sec

Example 20.4. Let's give a derivation of the sentence $(\varphi \wedge \psi) \rightarrow \varphi$.

We begin by writing the desired conclusion at the bottom of the derivation.

$$\dfrac{}{(\varphi \wedge \psi) \rightarrow \varphi}$$

Next, we need to figure out what kind of inference could result in a sentence of this form. The main operator of the conclusion is $\rightarrow$, so we'll try to arrive at the conclusion using the $\rightarrow$Intro rule. It is best to write down the assumptions involved and label the inference rules as you progress, so it is easy to see whether all assumptions have been discharged at the end of the proof.

$$1\,\dfrac{\begin{array}{c}[\varphi \wedge \psi]^1 \\ \vdots \\ \varphi\end{array}}{(\varphi \wedge \psi) \rightarrow \varphi}\,\rightarrow\text{Intro}$$

We now need to fill in the steps from the assumption $\varphi \wedge \psi$ to φ. Since we only have one connective to deal with, $\wedge$, we must use the $\wedge$ elim rule. This gives us the following proof:

$$\dfrac{\dfrac{[\varphi \wedge \psi]^1}{\varphi}\,\wedge\text{Elim}}{(\varphi \wedge \psi) \to \varphi}\,{}^{1}\!\to\text{Intro}$$

We now have a correct derivation of $(\varphi \wedge \psi) \to \varphi$.

Example 20.5. Now let's give a derivation of $(\neg\varphi \vee \psi) \to (\varphi \to \psi)$.

We begin by writing the desired conclusion at the bottom of the derivation.

$$\dfrac{}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}$$

To find a logical rule that could give us this conclusion, we look at the logical connectives in the conclusion: $\neg$, $\vee$, and $\to$. We only care at the moment about the first occurrence of $\to$ because it is the main operator of the sentence in the end-sequent, while $\neg$, $\vee$ and the second occurrence of $\to$ are inside the scope of another connective, so we will take care of those later. We therefore start with the $\to$Intro rule. A correct application must look like this:

$$\dfrac{\begin{array}{c}[\neg\varphi \vee \psi]^1\\ \vdots\\ \varphi \to \psi\end{array}}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}\,{}^{1}\!\to\text{Intro}$$

This leaves us with two possibilities to continue. Either we can keep working from the bottom up and look for another application of the $\to$Intro rule, or we can work from the top down and apply a $\vee$Elim rule. Let us apply the latter. We will use the assumption $\neg\varphi \vee \psi$ as the leftmost premise of $\vee$Elim. For a valid application of $\vee$Elim, the other two premises must be identical to the conclusion $\varphi \to \psi$, but each may be derived in turn from another assumption, namely one of the two disjuncts of $\neg\varphi \vee \psi$. So our derivation will look like this:

$$\dfrac{\dfrac{\begin{array}{ccc}[\neg\varphi \vee \psi]^1 & \begin{array}{c}[\neg\varphi]^2\\ \vdots\\ \varphi \to \psi\end{array} & \begin{array}{c}[\psi]^2\\ \vdots\\ \varphi \to \psi\end{array}\end{array}}{\varphi \to \psi}\,{}^{2}\!\vee\text{Elim}}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}\,{}^{1}\!\to\text{Intro}$$

In each of the two branches on the right, we want to derive $\varphi \to \psi$, which is best done using $\to$Intro.

$$\dfrac{\dfrac{[\neg\varphi\vee\psi]^1 \qquad \dfrac{\begin{array}{c}[\neg\varphi]^2,[\varphi]^3\\ \vdots\\ \psi\end{array}}{\varphi\to\psi}\,{\scriptstyle 3}\,{\to}\text{Intro} \qquad \dfrac{\begin{array}{c}[\psi]^2,[\varphi]^4\\ \vdots\\ \psi\end{array}}{\varphi\to\psi}\,{\scriptstyle 4}\,{\to}\text{Intro}}{\varphi\to\psi}\,{\scriptstyle 2}\,{\vee}\text{Elim}}{(\neg\varphi\vee\psi)\to(\varphi\to\psi)}\,{\scriptstyle 1}\,{\to}\text{Intro}$$

For the two missing parts of the derivation, we need derivations of ψ from $\neg\varphi$ and φ in the middle, and from φ and ψ on the left. Let's take the former first. $\neg\varphi$ and φ are the two premises of $\neg$Elim:

$$\dfrac{\begin{array}{c}[\neg\varphi]^2 \qquad [\varphi]^3\\ \hline \bot\\ \vdots\\ \psi\end{array}}{}\,\neg\text{Elim}$$

By using $\bot_I$, we can obtain ψ as a conclusion and complete the branch.

$$\dfrac{\dfrac{[\neg\varphi\vee\psi]^1 \qquad \dfrac{\dfrac{\begin{array}{c}[\neg\varphi]^2 \qquad [\varphi]^3\\ \hline \bot\end{array}}{\psi}\,\bot_I}{\varphi\to\psi}\,{\scriptstyle 3}\,{\to}\text{Intro} \qquad \dfrac{\begin{array}{c}[\psi]^2,[\varphi]^4\\ \vdots\\ \psi\end{array}}{\varphi\to\psi}\,{\scriptstyle 4}\,{\to}\text{Intro}}{\varphi\to\psi}\,{\scriptstyle 2}\,{\vee}\text{Elim}}{(\neg\varphi\vee\psi)\to(\varphi\to\psi)}\,{\scriptstyle 1}\,{\to}\text{Intro}$$

Let's now look at the rightmost branch. Here it's important to realize that the definition of derivation *allows assumptions to be discharged* but *does not require* them to be. In other words, if we can derive ψ from one of the assumptions φ and ψ without using the other, that's ok. And to derive ψ from ψ is trivial: ψ by itself is such a derivation, and no inferences are needed. So we can simply delete the assumption φ.

$$\dfrac{\dfrac{[\neg\varphi\vee\psi]^1 \qquad \dfrac{\dfrac{\dfrac{[\neg\varphi]^2 \qquad [\varphi]^3}{\bot}\,\neg\text{Elim}}{\psi}\,\bot_I}{\varphi\to\psi}\,{\scriptstyle 3}\,{\to}\text{Intro} \qquad \dfrac{[\psi]^2}{\varphi\to\psi}\,{\to}\text{Intro}}{\varphi\to\psi}\,{\scriptstyle 2}\,{\vee}\text{Elim}}{(\neg\varphi\vee\psi)\to(\varphi\to\psi)}\,{\scriptstyle 1}\,{\to}\text{Intro}$$

Note that in the finished derivation, the rightmost $\to$Intro inference does not actually discharge any assumptions.

Example 20.6. So far we have not needed the $\bot_C$ rule. It is special in that it allows us to discharge an assumption that isn't a sub-formula of the conclusion of the rule. It is closely related to the $\bot_I$ rule. In fact, the $\bot_I$ rule is a special case of the $\bot_C$ rule—there is a logic called "intuitionistic logic" in which only $\bot_I$ is allowed. The $\bot_C$ rule is a last resort when nothing else works. For instance, suppose we want to derive $\varphi \vee \neg\varphi$. Our usual strategy would be to attempt to derive $\varphi \vee \neg\varphi$ using $\vee$Intro. But this would require us to derive either φ or $\neg\varphi$ from no assumptions, and this can't be done. $\bot_C$ to the rescue!

$$\dfrac{\dfrac{\begin{array}{c}[\neg(\varphi \vee \neg\varphi)]^1 \\ \vdots \\ \bot\end{array}}{\varphi \vee \neg\varphi}\,1}{\bot_C}$$

Now we're looking for a derivation of $\bot$ from $\neg(\varphi \vee \neg\varphi)$. Since $\bot$ is the conclusion of $\neg$Elim we might try that:

$$\dfrac{\dfrac{\begin{array}{c}[\neg(\varphi \vee \neg\varphi)]^1 \\ \vdots \\ \neg\varphi\end{array} \qquad \begin{array}{c}[\neg(\varphi \vee \neg\varphi)]^1 \\ \vdots \\ \varphi\end{array}}{\bot}\,\neg\text{Elim}}{\dfrac{\bot}{\varphi \vee \neg\varphi}\,\bot_C}\,1$$

Our strategy for finding a derivation of $\neg\varphi$ calls for an application of $\neg$Intro:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}[\neg(\varphi \vee \neg\varphi)]^1, [\varphi]^2 \\ \vdots \\ \bot\end{array}}{\neg\varphi}\,\neg\text{Intro}\,2 \qquad \begin{array}{c}[\neg(\varphi \vee \neg\varphi)]^1 \\ \vdots \\ \varphi\end{array}}{\bot}\,\neg\text{Elim}}{\dfrac{\bot}{\varphi \vee \neg\varphi}\,\bot_C}\,1$$

Here, we can get $\bot$ easily by applying $\neg$Elim to the assumption $\neg(\varphi \vee \neg\varphi)$ and $\varphi \vee \neg\varphi$ which follows from our new assumption φ by $\vee$Intro:

$$\dfrac{\dfrac{\dfrac{\dfrac{[\neg(\varphi \vee \neg\varphi)]^1 \qquad \dfrac{[\varphi]^2}{\varphi \vee \neg\varphi}\,\vee\text{Intro}}{\bot}\,\neg\text{Elim}}{\neg\varphi}\,\neg\text{Intro}\,2 \qquad \begin{array}{c}[\neg(\varphi \vee \neg\varphi)]^1 \\ \vdots \\ \varphi\end{array}}{\bot}\,\neg\text{Elim}}{\dfrac{\bot}{\varphi \vee \neg\varphi}\,\bot_C}\,1$$

On the right side we use the same strategy, except we get φ by $\bot_C$:

$$\dfrac{\dfrac{\dfrac{[\neg(\varphi \vee \neg\varphi)]^1 \qquad \dfrac{[\varphi]^2}{\varphi \vee \neg\varphi}\vee\text{Intro}}{\neg\text{Elim}}}{2\ \dfrac{\bot}{\neg\varphi}\neg\text{Intro}} \qquad \dfrac{\dfrac{[\neg(\varphi \vee \neg\varphi)]^1 \qquad \dfrac{[\neg\varphi]^3}{\varphi \vee \neg\varphi}\vee\text{Intro}}{\neg\text{Elim}}}{3\ \dfrac{\bot}{\varphi}\bot_C}}{\neg\text{Elim}}}{1\ \dfrac{\bot}{\varphi \vee \neg\varphi}\bot_C}$$

Problem 20.1. Give derivations that show the following:

1. $\varphi \wedge (\psi \wedge \chi) \vdash (\varphi \wedge \psi) \wedge \chi$.
2. $\varphi \vee (\psi \vee \chi) \vdash (\varphi \vee \psi) \vee \chi$.
3. $\varphi \to (\psi \to \chi) \vdash \psi \to (\varphi \to \chi)$.
4. $\varphi \vdash \neg\neg\varphi$.

Problem 20.2. Give derivations that show the following:

1. $(\varphi \vee \psi) \to \chi \vdash \varphi \to \chi$.
2. $(\varphi \to \chi) \wedge (\psi \to \chi) \vdash (\varphi \vee \psi) \to \chi$.
3. $\vdash \neg(\varphi \wedge \neg\varphi)$.
4. $\psi \to \varphi \vdash \neg\varphi \to \neg\psi$.
5. $\vdash (\varphi \to \neg\varphi) \to \neg\varphi$.
6. $\vdash \neg(\varphi \to \psi) \to \neg\psi$.
7. $\varphi \to \chi \vdash \neg(\varphi \wedge \neg\chi)$.
8. $\varphi \wedge \neg\chi \vdash \neg(\varphi \to \chi)$.
9. $\varphi \vee \psi, \neg\psi \vdash \varphi$.
10. $\neg\varphi \vee \neg\psi \vdash \neg(\varphi \wedge \psi)$.
11. $\vdash (\neg\varphi \wedge \neg\psi) \to \neg(\varphi \vee \psi)$.
12. $\vdash \neg(\varphi \vee \psi) \to (\neg\varphi \wedge \neg\psi)$.

Problem 20.3. Give derivations that show the following:

1. $\neg(\varphi \to \psi) \vdash \varphi$.
2. $\neg(\varphi \wedge \psi) \vdash \neg\varphi \vee \neg\psi$.
3. $\varphi \to \psi \vdash \neg\varphi \vee \psi$.
4. $\vdash \neg\neg\varphi \to \varphi$.
5. $\varphi \to \psi, \neg\varphi \to \psi \vdash \psi$.

6. $(\varphi \land \psi) \to \chi \vdash (\varphi \to \chi) \lor (\psi \to \chi)$.

7. $(\varphi \to \psi) \to \varphi \vdash \varphi$.

8. $\vdash (\varphi \to \psi) \lor (\psi \to \chi)$.

(These all require the $\bot_C$ rule.)

---

content/first-order-logic/natural-deduction/proving-things-quant.tex

20.6 Derivations with Quantifiers

fol:ntd:prq:sec

Example 20.7. When dealing with quantifiers, we have to make sure not to violate the eigenvariable condition, and sometimes this requires us to play around with the order of carrying out certain inferences. In general, it helps to try and take care of rules subject to the eigenvariable condition first (they will be lower down in the finished proof).

Let's see how we'd give a derivation of the formula $\exists x\, \neg\varphi(x) \to \neg\forall x\, \varphi(x)$. Starting as usual, we write

$$\frac{}{\exists x\, \neg\varphi(x) \to \neg\forall x\, \varphi(x)}$$

We start by writing down what it would take to justify that last step using the $\to$Intro rule.

$$\dfrac{\begin{array}{c}[\exists x\, \neg\varphi(x)]^1 \\ \vdots \\ \neg\forall x\, \varphi(x)\end{array}}{\exists x\, \neg\varphi(x) \to \neg\forall x\, \varphi(x)}\, 1\ \to\text{Intro}$$

Since there is no obvious rule to apply to $\neg\forall x\, \varphi(x)$, we will proceed by setting up the derivation so we can use the $\exists$Elim rule. Here we must pay attention to the eigenvariable condition, and choose a constant that does not appear in $\exists x\, \varphi(x)$ or any assumptions that it depends on. (Since no constant symbols appear, however, any choice will do fine.)

$$\dfrac{\dfrac{\begin{array}{c}[\exists x\, \neg\varphi(x)]^1 \\ \\ \end{array} \qquad \begin{array}{c}[\neg\varphi(a)]^2 \\ \vdots \\ \neg\forall x\, \varphi(x)\end{array}}{\neg\forall x\, \varphi(x)}\, 2\ \exists\text{Elim}}{\exists x\, \neg\varphi(x) \to \neg\forall x\, \varphi(x)}\, 1\ \to\text{Intro}$$

In order to derive $\neg\forall x\,\varphi(x)$, we will attempt to use the $\neg$Intro rule: this requires that we derive a contradiction, possibly using $\forall x\,\varphi(x)$ as an additional assumption. Of course, this contradiction may involve the assumption $\neg\varphi(a)$ which will be discharged by the $\exists$Elim inference. We can set it up as follows:

$$\dfrac{\dfrac{[\exists x\,\neg\varphi(x)]^1 \qquad \dfrac{\begin{array}{c}[\neg\varphi(a)]^2,\,[\forall x\,\varphi(x)]^3\\ \vdots\\ \bot\end{array}}{\neg\forall x\,\varphi(x)}\,3\,\neg\text{Intro}}{\neg\forall x\,\varphi(x)}\,2\,\exists\text{Elim}}{\exists x\,\neg\varphi(x)\to\neg\forall x\,\varphi(x)}\,1\,\to\text{Intro}$$

It looks like we are close to getting a contradiction. The easiest rule to apply is the $\forall$Elim, which has no eigenvariable conditions. Since we can use any term we want to replace the universally quantified x, it makes the most sense to continue using a so we can reach a contradiction.

$$\dfrac{\dfrac{[\exists x\,\neg\varphi(x)]^1 \qquad \dfrac{\dfrac{[\neg\varphi(a)]^2 \qquad \dfrac{[\forall x\,\varphi(x)]^3}{\varphi(a)}\,\forall\text{Elim}}{\bot}\,\neg\text{Elim}}{\neg\forall x\,\varphi(x)}\,3\,\neg\text{Intro}}{\neg\forall x\,\varphi(x)}\,2\,\exists\text{Elim}}{\exists x\,\neg\varphi(x)\to\neg\forall x\,\varphi(x)}\,1\,\to\text{Intro}$$

It is important, especially when dealing with quantifiers, to double check at this point that the eigenvariable condition has not been violated. Since the only rule we applied that is subject to the eigenvariable condition was $\exists$Elim, and the eigenvariable a does not occur in any assumptions it depends on, this is a correct derivation.

Example 20.8. Sometimes we may derive a formula from other formulas. In these cases, we may have undischarged assumptions. It is important to keep track of our assumptions as well as the end goal.

Let's see how we'd give a derivation of the formula $\exists x\,\chi(x,b)$ from the assumptions $\exists x\,(\varphi(x)\wedge\psi(x))$ and $\forall x\,(\psi(x)\to\chi(x,b))$. Starting as usual, we write the conclusion at the bottom.

$$\dfrac{}{\exists x\,\chi(x,b)}$$

We have two premises to work with. To use the first, i.e., try to find a derivation of $\exists x\,\chi(x,b)$ from $\exists x\,(\varphi(x)\wedge\psi(x))$ we would use the $\exists$Elim rule. Since it has an eigenvariable condition, we will apply that rule first. We get the following:

$$\dfrac{\exists x\,(\varphi(x) \wedge \psi(x)) \qquad \begin{array}{c}[\varphi(a) \wedge \psi(a)]^1 \\ \vdots \\ \exists x\,\chi(x,b)\end{array}}{\exists x\,\chi(x,b)}\,{}^{1}\exists\text{Elim}$$

The two assumptions we are working with share ψ. It may be useful at this point to apply $\wedge$Elim to separate out $\psi(a)$.

$$\dfrac{\exists x\,(\varphi(x) \wedge \psi(x)) \qquad \begin{array}{c}\dfrac{[\varphi(a) \wedge \psi(a)]^1}{\psi(a)}\,\wedge\text{Elim} \\ \vdots \\ \exists x\,\chi(x,b)\end{array}}{\exists x\,\chi(x,b)}\,{}^{1}\exists\text{Elim}$$

The second assumption we have to work with is $\forall x\,(\psi(x) \to \chi(x,b))$. Since there is no eigenvariable condition we can instantiate x with the constant symbol a using $\forall$Elim to get $\psi(a) \to \chi(a,b)$. We now have both $\psi(a) \to \chi(a,b)$ and $\psi(a)$. Our next move should be a straightforward application of the $\to$Elim rule.

$$\dfrac{\exists x\,(\varphi(x) \wedge \psi(x)) \qquad \begin{array}{c}\dfrac{\dfrac{\forall x\,(\psi(x) \to \chi(x,b))}{\psi(a) \to \chi(a,b)}\,\forall\text{Elim} \qquad \dfrac{[\varphi(a) \wedge \psi(a)]^1}{\psi(a)}\,\wedge\text{Elim}}{\chi(a,b)}\,\to\text{Elim} \\ \vdots \\ \exists x\,\chi(x,b)\end{array}}{\exists x\,\chi(x,b)}\,{}^{1}\exists\text{Elim}$$

We are so close! One application of $\exists$Intro and we have reached our goal.

$$\dfrac{\exists x\,(\varphi(x) \wedge \psi(x)) \qquad \dfrac{\dfrac{\dfrac{\forall x\,(\psi(x) \to \chi(x,b))}{\psi(a) \to \chi(a,b)}\,\forall\text{Elim} \qquad \dfrac{[\varphi(a) \wedge \psi(a)]^1}{\psi(a)}\,\wedge\text{Elim}}{\chi(a,b)}\,\to\text{Elim}}{\exists x\,\chi(x,b)}\,\exists\text{Intro}}{\exists x\,\chi(x,b)}\,{}^{1}\exists\text{Elim}$$

Since we ensured at each step that the eigenvariable conditions were not violated, we can be confident that this is a correct derivation.

Example 20.9. Give a derivation of the formula $\neg\forall x\,\varphi(x)$ from the assumptions $\forall x\,\varphi(x) \to \exists y\,\psi(y)$ and $\neg\exists y\,\psi(y)$. Starting as usual, we write the target formula at the bottom.

$$\frac{}{\neg\forall x\,\varphi(x)}$$

The last line of the derivation is a negation, so let's try using ¬Intro. This will require that we figure out how to derive a contradiction.

$$\dfrac{\begin{array}{c}[\forall x\,\varphi(x)]^1\\ \vdots\\ \bot\end{array}}{\neg\forall x\,\varphi(x)}\,{}^1\,\neg\text{Intro}$$

So far so good. We can use ∀Elim but it's not obvious if that will help us get to our goal. Instead, let's use one of our assumptions. $\forall x\,\varphi(x) \to \exists y\,\psi(y)$ together with $\forall x\,\varphi(x)$ will allow us to use the →Elim rule.

$$\dfrac{\dfrac{\forall x\,\varphi(x) \to \exists y\,\psi(y) \qquad [\forall x\,\varphi(x)]^1}{\exists y\,\psi(y)}\,\to\text{Elim}\\ \vdots\\ \bot}{\neg\forall x\,\varphi(x)}\,{}^1\,\neg\text{Intro}$$

We now have one final assumption to work with, and it looks like this will help us reach a contradiction by using ¬Elim.

$$\dfrac{\dfrac{\neg\exists y\,\psi(y) \qquad \dfrac{\forall x\,\varphi(x) \to \exists y\,\psi(y) \qquad [\forall x\,\varphi(x)]^1}{\exists y\,\psi(y)}\,\to\text{Elim}}{\bot}\,\neg\text{Elim}}{\neg\forall x\,\varphi(x)}\,{}^1\,\neg\text{Intro}$$

Problem 20.4. Give derivations that show the following:

1. $\vdash (\forall x\,\varphi(x) \wedge \forall y\,\psi(y)) \to \forall z\,(\varphi(z) \wedge \psi(z))$.
2. $\vdash (\exists x\,\varphi(x) \vee \exists y\,\psi(y)) \to \exists z\,(\varphi(z) \vee \psi(z))$.
3. $\forall x\,(\varphi(x) \to \psi) \vdash \exists y\,\varphi(y) \to \psi$.
4. $\forall x\,\neg\varphi(x) \vdash \neg\exists x\,\varphi(x)$.
5. $\vdash \neg\exists x\,\varphi(x) \to \forall x\,\neg\varphi(x)$.
6. $\vdash \neg\exists x\,\forall y\,((\varphi(x,y) \to \neg\varphi(y,y)) \wedge (\neg\varphi(y,y) \to \varphi(x,y)))$.

Problem 20.5. Give derivations that show the following:

1. $\vdash \neg\forall x\,\varphi(x) \to \exists x\,\neg\varphi(x)$.
2. $(\forall x\,\varphi(x) \to \psi) \vdash \exists y\,(\varphi(y) \to \psi)$.

3. $\vdash \exists x\,(\varphi(x) \to \forall y\,\varphi(y))$.

(These all require the $\bot_C$ rule.)

---

content/first-order-logic/natural-deduction/proof-theoretic-notions.tex

20.7 Proof-Theoretic Notions

fol:ntd:ptn:sec

This section collects the definitions the provability relation and consistency for natural deduction.

explanation

Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain sentences from others. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

Definition 20.10 (Theorems). A sentence φ is a *theorem* if there is a derivation of φ in natural deduction in which all assumptions are discharged. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 20.11 (Derivability). A sentence φ is *derivable from* a set of sentences Γ, $\Gamma \vdash \varphi$, if there is a derivation with conclusion φ and in which every assumption is either discharged or is in Γ. If φ is not derivable from Γ we write $\Gamma \nvdash \varphi$.

Definition 20.12 (Consistency). A set of sentences Γ is *inconsistent* iff $\Gamma \vdash \bot$. If Γ is not inconsistent, i.e., if $\Gamma \nvdash \bot$, we say it is *consistent*.

Proposition 20.13 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.*

fol:ntd:ptn:prop:reflexivity

Proof. The assumption φ by itself is a derivation of φ where every undischarged assumption (i.e., φ) is in Γ. □

Proposition 20.14 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.*

fol:ntd:ptn:prop:monotonicity

Proof. Any derivation of φ from Γ is also a derivation of φ from Δ. □

Proposition 20.15 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.*

fol:ntd:ptn:prop:transitivity

Proof. If $\Gamma \vdash \varphi$, there is a derivation δ_0 of φ with all undischarged assumptions in Γ. If $\{\varphi\} \cup \Delta \vdash \psi$, then there is a derivation δ_1 of ψ with all undischarged assumptions in $\{\varphi\} \cup \Delta$. Now consider:

$$\dfrac{\dfrac{\Delta, [\varphi]^1 \quad \vdots\ \delta_1 \quad \psi}{\varphi \to \psi}\,{\scriptstyle 1}\ {\to}\text{Intro} \qquad \Gamma \quad \vdots\ \delta_0 \quad \varphi}{\psi}\ {\to}\text{Elim}$$

The undischarged assumptions are now all among $\Gamma \cup \Delta$, so this shows $\Gamma \cup \Delta \vdash \psi$. □

When $\Gamma = \{\varphi_1, \varphi_2, \ldots, \varphi_k\}$ is a finite set we may use the simplified notation $\varphi_1, \varphi_2, \ldots, \varphi_k \vdash \psi$ for $\Gamma \vdash \psi$, in particular $\varphi \vdash \psi$ means that $\{\varphi\} \vdash \psi$.

Note that if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \ldots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

fol:ntd:ptn: prop:incons

Proposition 20.16. *The following are equivalent.*

1. *Γ is inconsistent.*
2. *$\Gamma \vdash \varphi$ for every sentence φ.*
3. *$\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\varphi$ for some sentence φ.*

Proof. Exercise. □

Problem 20.6. Prove Proposition 20.16

fol:ntd:ptn: prop:proves-compact

Proposition 20.17 (Compactness).

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof. 1. If $\Gamma \vdash \varphi$, then there is a derivation δ of φ from Γ. Let Γ_0 be the set of undischarged assumptions of δ. Since any derivation is finite, Γ_0 can only contain finitely many sentences. So, δ is a derivation of φ from a finite $\Gamma_0 \subseteq \Gamma$.

2. This is the contrapositive of (1) for the special case $\varphi \equiv \bot$. □

20.8 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem. fol:ntd:prv:sec

Proposition 20.18. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.* fol:ntd:prv:prop:provability-contr

Proof. Let the derivation of φ from Γ be δ_1 and the derivation of $\bot$ from $\Gamma \cup \{\varphi\}$ be δ_2. We can then derive:

$$\dfrac{\dfrac{\Gamma, [\varphi]^1}{\vdots\,\delta_2}\,\bot}{\neg\varphi}\,{}^1\neg\text{Intro} \qquad \dfrac{\Gamma}{\vdots\,\delta_1}\,\varphi \qquad \neg\text{Elim}$$

In the new derivation, the assumption φ is discharged, so it is a derivation from Γ. □

Proposition 20.19. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.* fol:ntd:prv:prop:prov-incons

Proof. First suppose $\Gamma \vdash \varphi$, i.e., there is a derivation δ_0 of φ from undischarged assumptions Γ. We obtain a derivation of $\bot$ from $\Gamma \cup \{\neg\varphi\}$ as follows:

$$\dfrac{\neg\varphi \qquad \dfrac{\Gamma}{\vdots\,\delta_0}\,\varphi}{\bot}\,\neg\text{Elim}$$

Now assume $\Gamma \cup \{\neg\varphi\}$ is inconsistent, and let δ_1 be the corresponding derivation of $\bot$ from undischarged assumptions in $\Gamma \cup \{\neg\varphi\}$. We obtain a derivation of φ from Γ alone by using $\bot_C$:

$$\dfrac{\dfrac{\Gamma, [\neg\varphi]^1}{\vdots\,\delta_1}\,\bot}{\varphi}\,{}^1\bot_C$$

□

Problem 20.7. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

Proposition 20.20. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.* fol:ntd:prv:prop:explicit-inc

Proof. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there is a derivation δ of φ from Γ. Consider this simple application of the ¬Elim rule:

$$\dfrac{\neg\varphi \qquad \begin{array}{c}\Gamma\\ \vdots\\ \varphi\end{array}}{\bot}\ \neg\text{Elim}$$

Since $\neg\varphi \in \Gamma$, all undischarged assumptions are in Γ, this shows that $\Gamma \vdash \bot$.□

fol:ntd:prv: prop:provability-exhaustive

Proposition 20.21. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

Proof. There are derivations δ_1 and δ_2 of $\bot$ from $\Gamma \cup \{\varphi\}$ and $\bot$ from $\Gamma \cup \{\neg\varphi\}$, respectively. We can then derive

$$\dfrac{\dfrac{\Gamma, [\neg\varphi]^2}{\vdots\,\delta_2}\,\bot}{\neg\neg\varphi}\ {}^2\neg\text{Intro} \qquad \dfrac{\dfrac{\Gamma, [\varphi]^1}{\vdots\,\delta_1}\,\bot}{\neg\varphi}\ {}^1\neg\text{Intro}$$
$$\dfrac{\qquad\qquad}{\bot}\ \neg\text{Elim}$$

Since the assumptions φ and $\neg\varphi$ are discharged, this is a derivation of $\bot$ from Γ alone. Hence Γ is inconsistent. □

---

content/first-order-logic/natural-deduction/provability-propositional.tex

20.9 Derivability and the Propositional Connectives

fol:ntd:ppr:sec

We establish that the derivability relation $\vdash$ of natural deduction is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. explanation

fol:ntd:ppr: prop:provability-land

Proposition 20.22.

fol:ntd:ppr: prop:provability-land-left

1. *Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$*

fol:ntd:ppr: prop:provability-land-right

2. *$\varphi, \psi \vdash \varphi \wedge \psi$.*

Proof. 1. We can derive both

$$\dfrac{\varphi \wedge \psi}{\varphi}\ \wedge\text{Elim} \qquad \dfrac{\varphi \wedge \psi}{\psi}\ \wedge\text{Elim}$$

2. We can derive:

$$\frac{\varphi \qquad \psi}{\varphi \wedge \psi}\,\wedge\text{Intro}$$

□

Proposition 20.23.

fol:ntd:ppr:prop:provability-lor

1. *$\varphi \vee \psi, \neg\varphi, \neg\psi$ is inconsistent.*
2. *Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.*

Proof. 1. Consider the following derivation:

$$\dfrac{\varphi \vee \psi \qquad \dfrac{\neg\varphi \qquad [\varphi]^1}{\bot}\,\neg\text{Elim} \qquad \dfrac{\neg\psi \qquad [\psi]^1}{\bot}\,\neg\text{Elim}}{\bot}\,{}^1\vee\text{Elim}$$

This is a derivation of $\bot$ from undischarged assumptions $\varphi \vee \psi$, $\neg\varphi$, and $\neg\psi$.

2. We can derive both

$$\frac{\varphi}{\varphi \vee \psi}\,\vee\text{Intro} \qquad \frac{\psi}{\varphi \vee \psi}\,\vee\text{Intro}$$

□

Proposition 20.24.

fol:ntd:ppr:prop:provability-lif
fol:ntd:ppr:prop:provability-lif-left
fol:ntd:ppr:prop:provability-lif-right

1. *$\varphi, \varphi \to \psi \vdash \psi$.*
2. *Both $\neg\varphi \vdash \varphi \to \psi$ and $\psi \vdash \varphi \to \psi$.*

Proof. 1. We can derive:

$$\frac{\varphi \to \psi \qquad \varphi}{\psi}\,\to\text{Elim}$$

2. This is shown by the following two derivations:

$$\dfrac{\dfrac{\dfrac{\neg\varphi \qquad [\varphi]^1}{\bot}\,\neg\text{Elim}}{\psi}\,\bot_I}{\varphi \to \psi}\,{}^1\to\text{Intro} \qquad \frac{\psi}{\varphi \to \psi}\,\to\text{Intro}$$

Note that $\to$Intro may, but does not have to, discharge the assumption φ.
□

content/first-order-logic/natural-deduction/provability-quantifiers.tex

20.10 Derivability and the Quantifiers

fol:ntd:qpr:sec

The completeness theorem also requires that the natural deduction rules yield the facts about $\vdash$ established in this section. explanation

fol:ntd:qpr:thm:strong-generalization

Theorem 20.25. *If c is a constant not occurring in Γ or $\varphi(x)$ and $\Gamma \vdash \varphi(c)$, then $\Gamma \vdash \forall x\, \varphi(x)$.*

Proof. Let δ be a derivation of $\varphi(c)$ from Γ. By adding a $\forall$Intro inference, we obtain a derivation of $\forall x\, \varphi(x)$. Since c does not occur in Γ or $\varphi(x)$, the eigenvariable condition is satisfied. □

fol:ntd:qpr:prop:provability-quantifiers

Proposition 20.26.

1. $\varphi(t) \vdash \exists x\, \varphi(x)$.
2. $\forall x\, \varphi(x) \vdash \varphi(t)$.

Proof. 1. The following is a derivation of $\exists x\, \varphi(x)$ from $\varphi(t)$:

$$\dfrac{\varphi(t)}{\exists x\, \varphi(x)}\ \exists\text{Intro}$$

2. The following is a derivation of $\varphi(t)$ from $\forall x\, \varphi(x)$:

$$\dfrac{\forall x\, \varphi(x)}{\varphi(t)}\ \forall\text{Elim}$$

□

---

content/first-order-logic/natural-deduction/soundness.tex

20.11 Soundness

fol:ntd:sou:sec

A derivation system, such as natural deduction, is *sound* if it cannot derive things that do not actually follow. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that explanation

1. every derivable sentence is valid;
2. if a sentence is derivable from some others, it is also a consequence of them;
3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Theorem 20.27 (Soundness). *If φ is derivable from the undischarged assumptions Γ, then $\Gamma \vDash \varphi$.* fol:ntd:sou:thm:soundness

Proof. Let δ be a derivation of φ. We proceed by induction on the number of inferences in δ.

For the induction basis we show the claim if the number of inferences is 0. In this case, δ consists only of a single sentence φ, i.e., an assumption. That assumption is undischarged, since assumptions can only be discharged by inferences, and there are no inferences. So, any structure $\mathfrak{M}$ that satisfies all of the undischarged assumptions of the proof also satisfies φ.

Now for the inductive step. Suppose that δ contains n inferences. The premise(s) of the lowermost inference are derived using sub-derivations, each of which contains fewer than n inferences. We assume the induction hypothesis: The premises of the lowermost inference follow from the undischarged assumptions of the sub-derivations ending in those premises. We have to show that the conclusion φ follows from the undischarged assumptions of the entire proof.

We distinguish cases according to the type of the lowermost inference. First, we consider the possible inferences with only one premise.

1. Suppose that the last inference is ¬Intro: The derivation has the form

$$\dfrac{\begin{array}{c}\Gamma, [\varphi]^n \\ \vdots\,\delta_1\,\vdots \\ \bot\end{array}}{\neg\varphi}\,{}^{n}\,\neg\text{Intro}$$

   By inductive hypothesis, $\bot$ follows from the undischarged assumptions $\Gamma \cup \{\varphi\}$ of δ_1. Consider a structure $\mathfrak{M}$. We need to show that, if $\mathfrak{M} \vDash \Gamma$, then $\mathfrak{M} \vDash \neg\varphi$. Suppose for reductio that $\mathfrak{M} \vDash \Gamma$, but $\mathfrak{M} \nvDash \neg\varphi$, i.e., $\mathfrak{M} \vDash \varphi$. This would mean that $\mathfrak{M} \vDash \Gamma \cup \{\varphi\}$. This is contrary to our inductive hypothesis. So, $\mathfrak{M} \vDash \neg\varphi$.

2. The last inference is ∧Elim: There are two variants: φ or ψ may be inferred from the premise $\varphi \wedge \psi$. Consider the first case. The derivation δ looks like this:

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots\,\delta_1\,\vdots \\ \varphi \wedge \psi\end{array}}{\varphi}\,\wedge\text{Elim}$$

   By inductive hypothesis, $\varphi \wedge \psi$ follows from the undischarged assumptions Γ of δ_1. Consider a structure $\mathfrak{M}$. We need to show that, if $\mathfrak{M} \vDash \Gamma$,

then $\mathfrak{M} \vDash \varphi$. Suppose $\mathfrak{M} \vDash \Gamma$. By our inductive hypothesis ($\Gamma \vDash \varphi \wedge \psi$), we know that $\mathfrak{M} \vDash \varphi \wedge \psi$. By definition, $\mathfrak{M} \vDash \varphi \wedge \psi$ iff $\mathfrak{M} \vDash \varphi$ and $\mathfrak{M} \vDash \psi$. (The case where ψ is inferred from $\varphi \wedge \psi$ is handled similarly.)

3. The last inference is $\vee$Intro: There are two variants: $\varphi \vee \psi$ may be inferred from the premise φ or the premise ψ. Consider the first case. The derivation has the form

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \varphi\end{array}}{\varphi \vee \psi}\ \vee\text{Intro}$$

   By inductive hypothesis, φ follows from the undischarged assumptions Γ of δ_1. Consider a structure $\mathfrak{M}$. We need to show that, if $\mathfrak{M} \vDash \Gamma$, then $\mathfrak{M} \vDash \varphi \vee \psi$. Suppose $\mathfrak{M} \vDash \Gamma$; then $\mathfrak{M} \vDash \varphi$ since $\Gamma \vDash \varphi$ (the inductive hypothesis). So it must also be the case that $\mathfrak{M} \vDash \varphi \vee \psi$. (The case where $\varphi \vee \psi$ is inferred from ψ is handled similarly.)

4. The last inference is $\to$Intro: $\varphi \to \psi$ is inferred from a subproof with assumption φ and conclusion ψ, i.e.,

$$\dfrac{\begin{array}{c}\Gamma, [\varphi]^n \\ \vdots \\ \delta_1 \\ \vdots \\ \psi\end{array}}{\varphi \to \psi}\ {}^{n}\ \to\text{Intro}$$

   By inductive hypothesis, ψ follows from the undischarged assumptions of δ_1, i.e., $\Gamma \cup \{\varphi\} \vDash \psi$. Consider a structure $\mathfrak{M}$. The undischarged assumptions of δ are just Γ, since φ is discharged at the last inference. So we need to show that $\Gamma \vDash \varphi \to \psi$. For reductio, suppose that for some structure $\mathfrak{M}$, $\mathfrak{M} \vDash \Gamma$ but $\mathfrak{M} \nvDash \varphi \to \psi$. So, $\mathfrak{M} \vDash \varphi$ and $\mathfrak{M} \nvDash \psi$. But by hypothesis, ψ is a consequence of $\Gamma \cup \{\varphi\}$, i.e., $\mathfrak{M} \vDash \psi$, which is a contradiction. So, $\Gamma \vDash \varphi \to \psi$.

5. The last inference is $\bot_I$: Here, δ ends in

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \bot\end{array}}{\varphi}\ \bot_I$$

By induction hypothesis, $\Gamma \vDash \bot$. We have to show that $\Gamma \vDash \varphi$. Suppose not; then for some $\mathfrak{M}$ we have $\mathfrak{M} \vDash \Gamma$ and $\mathfrak{M} \nvDash \varphi$. But we always have $\mathfrak{M} \nvDash \bot$, so this would mean that $\Gamma \nvDash \bot$, contrary to the induction hypothesis.

6. The last inference is $\bot_C$: Exercise.

7. The last inference is $\forall$Intro: Then δ has the form

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ \delta_1\\ \vdots\\ \varphi(a)\end{array}}{\forall x\, \varphi(x)}\ \forall\text{Intro}$$

The premise $\varphi(a)$ is a consequence of the undischarged assumptions Γ by induction hypothesis. Consider some structure, $\mathfrak{M}$, such that $\mathfrak{M} \vDash \Gamma$. We need to show that $\mathfrak{M} \vDash \forall x\, \varphi(x)$. Since $\forall x\, \varphi(x)$ is a sentence, this means we have to show that for every variable assignment s, $\mathfrak{M}, s \vDash \varphi(x)$ (Proposition 16.19). Since Γ consists entirely of sentences, $\mathfrak{M}, s \vDash \psi$ for all $\psi \in \Gamma$ by Definition 16.11. Let $\mathfrak{M}'$ be like $\mathfrak{M}$ except that $a^{\mathfrak{M}'} = s(x)$. Since a does not occur in Γ, $\mathfrak{M}' \vDash \Gamma$ by Corollary 16.21. Since $\Gamma \vDash \varphi(a)$, $\mathfrak{M}' \vDash \varphi(a)$. Since $\varphi(a)$ is a sentence, $\mathfrak{M}', s \vDash \varphi(a)$ by Proposition 16.18. $\mathfrak{M}', s \vDash \varphi(x)$ iff $\mathfrak{M}' \vDash \varphi(a)$ by Proposition 16.23 (recall that $\varphi(a)$ is just $\varphi(x)[a/x]$). So, $\mathfrak{M}', s \vDash \varphi(x)$. Since a does not occur in $\varphi(x)$, by Proposition 16.20, $\mathfrak{M}, s \vDash \varphi(x)$. But s was an arbitrary variable assignment, so $\mathfrak{M} \vDash \forall x\, \varphi(x)$.

8. The last inference is $\exists$Intro: Exercise.

9. The last inference is $\forall$Elim: Exercise.

Now let's consider the possible inferences with several premises: $\vee$Elim, $\wedge$Intro, $\to$Elim, and $\exists$Elim.

1. The last inference is $\wedge$Intro. $\varphi \wedge \psi$ is inferred from the premises φ and ψ and δ has the form

$$\dfrac{\begin{array}{cc}\Gamma_1 & \Gamma_2\\ \vdots & \vdots\\ \delta_1 & \delta_2\\ \vdots & \vdots\\ \varphi & \psi\end{array}}{\varphi \wedge \psi}\ \wedge\text{Intro}$$

By induction hypothesis, φ follows from the undischarged assumptions Γ_1 of δ_1 and ψ follows from the undischarged assumptions Γ_2 of δ_2. The undischarged assumptions of δ are $\Gamma_1 \cup \Gamma_2$, so we have to show that

$\Gamma_1 \cup \Gamma_2 \vDash \varphi \wedge \psi$. Consider a structure $\mathfrak{M}$ with $\mathfrak{M} \vDash \Gamma_1 \cup \Gamma_2$. Since $\mathfrak{M} \vDash \Gamma_1$, it must be the case that $\mathfrak{M} \vDash \varphi$ as $\Gamma_1 \vDash \varphi$, and since $\mathfrak{M} \vDash \Gamma_2$, $\mathfrak{M} \vDash \psi$ since $\Gamma_2 \vDash \psi$. Together, $\mathfrak{M} \vDash \varphi \wedge \psi$.

2. The last inference is $\vee$Elim: Exercise.

3. The last inference is $\to$Elim. ψ is inferred from the premises $\varphi \to \psi$ and φ. The derivation δ looks like this:

$$\dfrac{\begin{array}{cc}\Gamma_1 & \Gamma_2 \\ \vdots & \vdots \\ \delta_1 & \delta_2 \\ \vdots & \vdots \\ \varphi \to \psi & \varphi\end{array}}{\psi} \to\text{Elim}$$

By induction hypothesis, $\varphi \to \psi$ follows from the undischarged assumptions Γ_1 of δ_1 and φ follows from the undischarged assumptions Γ_2 of δ_2. Consider a structure $\mathfrak{M}$. We need to show that, if $\mathfrak{M} \vDash \Gamma_1 \cup \Gamma_2$, then $\mathfrak{M} \vDash \psi$. Suppose $\mathfrak{M} \vDash \Gamma_1 \cup \Gamma_2$. Since $\Gamma_1 \vDash \varphi \to \psi$, $\mathfrak{M} \vDash \varphi \to \psi$. Since $\Gamma_2 \vDash \varphi$, we have $\mathfrak{M} \vDash \varphi$. This means that $\mathfrak{M} \vDash \psi$ (For if $\mathfrak{M} \nvDash \psi$, since $\mathfrak{M} \vDash \varphi$, we'd have $\mathfrak{M} \nvDash \varphi \to \psi$, contradicting $\mathfrak{M} \vDash \varphi \to \psi$).

4. The last inference is $\neg$Elim: Exercise.

5. The last inference is $\exists$Elim: Exercise. □

Problem 20.8. Complete the proof of Theorem 20.27.

fol:ntd:sou:cor:weak-soundness

Corollary 20.28. *If $\vdash \varphi$, then φ is valid.*

fol:ntd:sou:cor:consistency-soundness

Corollary 20.29. *If Γ is satisfiable, then it is consistent.*

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then $\Gamma \vdash \bot$, i.e., there is a derivation of $\bot$ from undischarged assumptions in Γ. By Theorem 20.27, any structure $\mathfrak{M}$ that satisfies Γ must satisfy $\bot$. Since $\mathfrak{M} \nvDash \bot$ for every structure $\mathfrak{M}$, no $\mathfrak{M}$ can satisfy Γ, i.e., Γ is not satisfiable. □

---

content/first-order-logic/natural-deduction/identity.tex

20.12 Derivations with Identity predicate

fol:ntd:ide:sec

Derivations with identity predicate require additional inference rules.

$$\frac{}{t = t}\ =\text{Intro} \qquad \frac{t_1 = t_2 \qquad \varphi(t_1)}{\varphi(t_2)}\ =\text{Elim} \qquad \frac{t_1 = t_2 \qquad \varphi(t_2)}{\varphi(t_1)}\ =\text{Elim}$$

In the above rules, t, t_1, and t_2 are closed terms. The =Intro rule allows us to derive any identity statement of the form $t = t$ outright, from no assumptions.

Example 20.30. If s and t are closed terms, then $\varphi(s), s = t \vdash \varphi(t)$:

$$\frac{s = t \qquad \varphi(s)}{\varphi(t)}\ =\text{Elim}$$

This may be familiar as the "principle of substitutability of identicals," or Leibniz' Law.

Problem 20.9. Prove that $=$ is both symmetric and transitive, i.e., give derivations of $\forall x\, \forall y\, (x = y \to y = x)$ and $\forall x\, \forall y\, \forall z((x = y \wedge y = z) \to x = z)$

Example 20.31. We derive the sentence

$$\forall x\, \forall y\, ((\varphi(x) \wedge \varphi(y)) \to x = y)$$

from the sentence

$$\exists x\, \forall y\, (\varphi(y) \to y = x)$$

We develop the derivation backwards:

$$\dfrac{\dfrac{\dfrac{\exists x\, \forall y\, (\varphi(y) \to y = x) \quad [\varphi(a) \wedge \varphi(b)]^1}{\vdots}}{\dfrac{a = b}{((\varphi(a) \wedge \varphi(b)) \to a = b)}\ 1\ \to\text{Intro}}}{\dfrac{\forall y\, ((\varphi(a) \wedge \varphi(y)) \to a = y)}{\forall x\, \forall y\, ((\varphi(x) \wedge \varphi(y)) \to x = y)}\ \forall\text{Intro}}\ \forall\text{Intro}$$

We'll now have to use the main assumption: since it is an existential formula, we use $\exists$Elim to derive the intermediary conclusion $a = b$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\exists x\,\forall y\,(\varphi(y) \to y = x) \qquad \begin{array}{c}[\forall y\,(\varphi(y) \to y = c)]^2\\ [\varphi(a) \wedge \varphi(b)]^1\\ \vdots\\ a = b\end{array}}{a = b}\,\exists\text{Elim}^{2}}{((\varphi(a) \wedge \varphi(b)) \to a = b)}\,{\to}\text{Intro}^{1}}{\forall y\,((\varphi(a) \wedge \varphi(y)) \to a = y)}\,\forall\text{Intro}}{\forall x\,\forall y\,((\varphi(x) \wedge \varphi(y)) \to x = y)}\,\forall\text{Intro}$$

The sub-derivation on the top right is completed by using its assumptions to show that $a = c$ and $b = c$. This requires two separate derivations. The derivation for $a = c$ is as follows:

$$\dfrac{\dfrac{[\forall y\,(\varphi(y) \to y = c)]^2}{\varphi(a) \to a = c}\,\forall\text{Elim} \qquad \dfrac{[\varphi(a) \wedge \varphi(b)]^1}{\varphi(a)}\,\wedge\text{Elim}}{a = c}\,{\to}\text{Elim}$$

From $a = c$ and $b = c$ we derive $a = b$ by $=$Elim.

Problem 20.10. Give derivations of the following formulas:

1. $\forall x\,\forall y\,((x = y \wedge \varphi(x)) \to \varphi(y))$
2. $\exists x\,\varphi(x) \wedge \forall y\,\forall z\,((\varphi(y) \wedge \varphi(z)) \to y = z) \to \exists x\,(\varphi(x) \wedge \forall y\,(\varphi(y) \to y = x))$

---

content/first-order-logic/natural-deduction/soundness-identity.tex

20.13 Soundness with Identity predicate

fol:ntd:sid:sec

Proposition 20.32. *Natural deduction with rules for $=$ is sound.*

Proof. Any formula of the form $t = t$ is valid, since for every structure $\mathfrak{M}$, $\mathfrak{M} \vDash t = t$. (Note that we assume the term t to be closed, i.e., it contains no variables, so variable assignments are irrelevant).

Suppose the last inference in a derivation is $=$Elim, i.e., the derivation has the following form:

$$\dfrac{\begin{array}{c}\Gamma_1\\ \vdots\,\delta_1\\ t_1 = t_2\end{array} \qquad \begin{array}{c}\Gamma_2\\ \vdots\,\delta_2\\ \varphi(t_1)\end{array}}{\varphi(t_2)}\,=\text{Elim}$$

The premises $t_1 = t_2$ and $\varphi(t_1)$ are derived from undischarged assumptions Γ_1 and Γ_2, respectively. We want to show that $\varphi(t_2)$ follows from $\Gamma_1 \cup \Gamma_2$. Consider a structure $\mathfrak{M}$ with $\mathfrak{M} \vDash \Gamma_1 \cup \Gamma_2$. By induction hypothesis, $\mathfrak{M} \vDash \varphi(t_1)$ and $\mathfrak{M} \vDash t_1 = t_2$. Therefore, $\mathrm{Val}^{\mathfrak{M}}(t_1) = \mathrm{Val}^{\mathfrak{M}}(t_2)$. Let s be any variable assignment, and $m = \mathrm{Val}^{\mathfrak{M}}(t_1) = \mathrm{Val}^{\mathfrak{M}}(t_2)$. By Proposition 16.23, $\mathfrak{M}, s \vDash \varphi(t_1)$ iff $\mathfrak{M}, s[m/x] \vDash \varphi(x)$ iff $\mathfrak{M}, s \vDash \varphi(t_2)$. Since $\mathfrak{M} \vDash \varphi(t_1)$, we have $\mathfrak{M} \vDash \varphi(t_2)$. □

Chapter 21

Tableaux

This chapter presents a signed analytic tableaux system.
To include or exclude material relevant to natural deduction as a proof system, use the "prfTab" tag.

---

content/first-order-logic/tableaux/rules-and-proofs.tex

21.1 Rules and Tableaux

fol:tab:rul:sec

A tableau is a systematic survey of the possible ways a sentence can be true or false in a structure. The building blocks of a tableau are signed formulas: sentences plus a truth value "sign," either $\mathbb{T}$ or $\mathbb{F}$. These signed formulas are arranged in a (downward growing) tree.

Definition 21.1. A *signed formula* is a pair consisting of a truth value and a sentence, i.e., either:

$$\mathbb{T}\varphi \text{ or } \mathbb{F}\varphi.$$

Intuitively, we might read $\mathbb{T}\varphi$ as "φ might be true" and $\mathbb{F}\varphi$ as "φ might be false" (in some structure).

Each signed formula in the tree is either an *assumption* (which are listed at the very top of the tree), or it is obtained from a signed formula above it by one of a number of rules of inference. There are two rules for each possible main operator of the preceding formula, one for the case where the sign is $\mathbb{T}$,

and one for the case where the sign is $\mathbb{F}$. Some rules allow the tree to branch, and some only add signed formulas to the branch. A rule may be (and often must be) applied not to the immediately preceding signed formula, but to any signed formula in the branch from the root to the place the rule is applied.

A branch is *closed* when it contains both $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$. A closed tableau is one where every branch is closed. Under the intuitive interpretation, any branch describes a joint possibility, but $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$ are not jointly possible. In other words, if a branch is closed, the possibility it describes has been ruled out. In particular, that means that a closed tableau rules out all possibilities of simultaneously making every assumption of the form $\mathbb{T}\,\varphi$ true and every assumption of the form $\mathbb{F}\,\varphi$ false.

A closed tableau *for* φ is a closed tableau with root $\mathbb{F}\,\varphi$. If such a closed tableau exists, all possibilities for φ being false have been ruled out; i.e., φ must be true in every structure.

---

content/first-order-logic/tableaux/propositional-rules.tex

21.2 Propositional Rules

fol:tab:prl:sec

Rules for $\neg$

$$\frac{\mathbb{T}\,\neg\varphi}{\mathbb{F}\,\varphi}\,\neg\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\neg\varphi}{\mathbb{T}\,\varphi}\,\neg\mathbb{F}$$

Rules for $\wedge$

$$\frac{\mathbb{T}\,\varphi \wedge \psi}{\begin{array}{l}\mathbb{T}\,\varphi\\ \mathbb{T}\,\psi\end{array}}\,\wedge\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\varphi \wedge \psi}{\mathbb{F}\,\varphi \quad\big|\quad \mathbb{F}\,\psi}\,\wedge\mathbb{F}$$

Rules for $\vee$

$$\frac{\mathbb{T}\,\varphi \vee \psi}{\mathbb{T}\,\varphi \quad\big|\quad \mathbb{T}\,\psi}\,\vee\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\varphi \vee \psi}{\begin{array}{l}\mathbb{F}\,\varphi\\ \mathbb{F}\,\psi\end{array}}\,\vee\mathbb{F}$$

Rules for $\to$

$$\frac{\mathbb{T}\,\varphi \to \psi}{\mathbb{F}\,\varphi \quad | \quad \mathbb{T}\,\psi}\ {\to}\mathbb{T} \qquad \frac{\mathbb{F}\,\varphi \to \psi}{\begin{array}{c}\mathbb{T}\,\varphi\\ \mathbb{F}\,\psi\end{array}}\ {\to}\mathbb{F}$$

The Cut Rule

$$\frac{}{\mathbb{T}\,\varphi \quad | \quad \mathbb{F}\,\varphi}\ \text{Cut}$$

The Cut rule is not applied "to" a previous signed formula; rather, it allows every branch in a tableau to be split in two, one branch containing $\mathbb{T}\,\varphi$, the other $\mathbb{F}\,\varphi$. It is not necessary—any set of signed formulas with a closed tableau has one not using Cut—but it allows us to combine tableaux in a convenient way.

---

content/first-order-logic/tableaux/quantifier-rules.tex

21.3 Quantifier Rules

Rules for $\forall$

fol:tab:qrl:sec

$$\frac{\mathbb{T}\,\forall x\,\varphi(x)}{\mathbb{T}\,\varphi(t)}\ \forall\mathbb{T} \qquad \frac{\mathbb{F}\,\forall x\,\varphi(x)}{\mathbb{F}\,\varphi(a)}\ \forall\mathbb{F}$$

In $\forall\mathbb{T}$, t is a closed term (i.e., one without variables). In $\forall\mathbb{F}$, a is a constant symbol which must not occur anywhere in the branch above $\forall\mathbb{F}$ rule. We call a the *eigenvariable* of the $\forall\mathbb{F}$ inference.[1]

Rules for $\exists$

$$\frac{\mathbb{T}\,\exists x\,\varphi(x)}{\mathbb{T}\,\varphi(a)}\ \exists\mathbb{T} \qquad \frac{\mathbb{F}\,\exists x\,\varphi(x)}{\mathbb{F}\,\varphi(t)}\ \exists\mathbb{F}$$

[1] We use the term "eigenvariable" even though a in the above rule is a constant symbol. This has historical reasons.

Again, t is a closed term, and a is a constant symbol which does not occur in the branch above the $\exists\mathbb{T}$ rule. We call a the *eigenvariable* of the $\exists\mathbb{T}$ inference.

The condition that an eigenvariable not occur in the branch above the $\forall\mathbb{F}$ or $\exists\mathbb{T}$ inference is called the *eigenvariable condition*.

Recall the convention that when φ is a formula with the variable x free, we indicate this by writing $\varphi(x)$. In the same context, $\varphi(t)$ then is short for $\varphi[t/x]$. So we could also write the $\exists\mathbb{F}$ rule as: explanation

$$\frac{\mathbb{F}\,\exists x\,\varphi}{\mathbb{F}\,\varphi[t/x]}\,\exists\mathbb{F}$$

Note that t may already occur in φ, e.g., φ might be $P(t,x)$. Thus, inferring $\mathbb{F}\,P(t,t)$ from $\mathbb{F}\,\exists x\,P(t,x)$ is a correct application of $\exists\mathbb{F}$. However, the eigenvariable conditions in $\forall\mathbb{F}$ and $\exists\mathbb{T}$ require that the constant symbol a does not occur in φ. So, you cannot correctly infer $\mathbb{F}\,P(a,a)$ from $\mathbb{F}\,\forall x\,P(a,x)$ using $\forall\mathbb{F}$.

In $\forall\mathbb{T}$ and $\exists\mathbb{F}$ there are no restrictions on the term t. On the other hand, in the $\exists\mathbb{T}$ and $\forall\mathbb{F}$ rules, the eigenvariable condition requires that the constant symbol a does not occur anywhere in the branches above the respective inference. It is necessary to ensure that the system is sound. Without this condition, the following would be a closed tableau for $\exists x\,\varphi(x) \to \forall x\,\varphi(x)$: explanation explanation

1.	$\mathbb{F}\,\exists x\,\varphi(x) \to \forall x\,\varphi(x)$	Assumption
2.	$\mathbb{T}\,\exists x\,\varphi(x)$	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\forall x\,\varphi(x)$	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\varphi(a)$	$\exists\mathbb{T}\,2$
5.	$\mathbb{F}\,\varphi(a)$	$\forall\mathbb{F}\,3$
	$\otimes$	

However, $\exists x\,\varphi(x) \to \forall x\,\varphi(x)$ is not valid.

---

content/first-order-logic/tableaux/derivations.tex

21.4 Tableaux

fol:tab:der:sec

We've said what an assumption is, and we've given the rules of inference. Tableaux are inductively generated from these: each tableau either is a single branch consisting of one or more assumptions, or it results from a tableau by applying one of the rules of inference on a branch. explanation

Definition 21.2 (Tableau). A tableau for assumptions $S_1\varphi_1, \ldots, S_n\varphi_n$ (where each S_i is either $\mathbb{T}$ or $\mathbb{F}$) is a finite tree of signed formulas satisfying the following conditions:

1. The n topmost signed formulas of the tree are $S_i\varphi_i$, one below the other.
2. Every signed formula in the tree that is not one of the assumptions results from a correct application of an inference rule to a signed formula in the branch above it.

A branch of a tableau is *closed* iff it contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, and *open* otherwise. A tableau in which every branch is closed is a *closed tableau* (for its set of assumptions). If a tableau is not closed, i.e., if it contains at least one open branch, it is *open*.

Example 21.3. Every set of assumptions on its own is a tableau, but it will generally not be closed. (Obviously, it is closed only if the assumptions already contain a pair of signed formulas $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$.)

From a tableau (open or closed) we can obtain a new, larger one by applying one of the rules of inference to a signed formula φ in it. The rule will append one or more signed formulas to the end of any branch containing the occurrence of φ to which we apply the rule.

For instance, consider the assumption $\mathbb{T}\varphi \wedge \neg\varphi$. Here is the (open) tableau consisting of just that assumption:

1.	$\mathbb{T}\varphi \wedge \neg\varphi$	Assumption

We obtain a new tableau from it by applying the $\wedge\mathbb{T}$ rule to the assumption. That rule allows us to add two new lines to the tableau, $\mathbb{T}\varphi$ and $\mathbb{T}\neg\varphi$:

1.	$\mathbb{T}\varphi \wedge \neg\varphi$	Assumption
2.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}\,1$
3.	$\mathbb{T}\neg\varphi$	$\wedge\mathbb{T}\,1$

When we write down tableaux, we record the rules we've applied on the right (e.g., $\wedge\mathbb{T}1$ means that the signed formula on that line is the result of applying the $\wedge\mathbb{T}$ rule to the signed formula on line 1). This new tableau now contains additional signed formulas, but to only one ($\mathbb{T}\neg\varphi$) can we apply a rule (in this case, the $\neg\mathbb{T}$ rule). This results in the closed tableau

1.	$\mathbb{T}\varphi \wedge \neg\varphi$	Assumption
2.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}\,1$
3.	$\mathbb{T}\neg\varphi$	$\wedge\mathbb{T}\,1$
4.	$\mathbb{F}\varphi$	$\neg\mathbb{T}\,3$
	$\otimes$	

---

content/first-order-logic/tableaux/proving-things.tex

21.5 Examples of Tableaux

fol:tab:pro:sec

Example 21.4. Let's find a closed tableau for the sentence $(\varphi \wedge \psi) \to \varphi$.

We begin by writing the corresponding assumption at the top of the tableau.

1.	$\mathbb{F}(\varphi \wedge \psi) \to \varphi$	Assumption

There is only one assumption, so only one signed formula to which we can apply a rule. (For every signed formula, there is always at most one rule that can be applied: it's the rule for the corresponding sign and main operator of the sentence.) In this case, this means, we must apply $\to\mathbb{F}$.

1.	$\mathbb{F}\,(\varphi \wedge \psi) \to \varphi$ ✓	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\varphi$	$\to\mathbb{F}\,1$

To keep track of which signed formulas we have applied their corresponding rules to, we write a checkmark next to the sentence. However, *only* write a checkmark if the rule has been applied to all open branches. Once a signed formula has had the corresponding rule applied in every open branch, we will not have to return to it and apply the rule again. In this case, there is only one branch, so the rule only has to be applied once. (Note that checkmarks are only a convenience for constructing tableaux and are not officially part of the syntax of tableaux.)

There is one new signed formula to which we can apply a rule: the $\mathbb{T}\,\varphi \wedge \psi$ on line 2. Applying the $\wedge\mathbb{T}$ rule results in:

1.	$\mathbb{F}\,(\varphi \wedge \psi) \to \varphi$ ✓	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\varphi$	$\wedge\mathbb{T}\,2$
5.	$\mathbb{T}\,\psi$	$\wedge\mathbb{T}\,2$
	$\otimes$	

Since the branch now contains both $\mathbb{T}\,\varphi$ (on line 4) and $\mathbb{F}\,\varphi$ (on line 3), the branch is closed. Since it is the only branch, the tableau is closed. We have found a closed tableau for $(\varphi \wedge \psi) \to \varphi$.

Example 21.5. Now let's find a closed tableau for $(\neg\varphi \vee \psi) \to (\varphi \to \psi)$.

We begin with the corresponding assumption:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$	Assumption

The one signed formula in this tableau has main operator $\to$ and sign $\mathbb{F}$, so we apply the $\to\mathbb{F}$ rule to it to obtain:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$ ✓	Assumption
2.	$\mathbb{T}\,\neg\varphi \vee \psi$	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,(\varphi \to \psi)$	$\to\mathbb{F}\,1$

We now have a choice as to whether to apply $\vee\mathbb{T}$ to line 2 or $\to\mathbb{F}$ to line 3. It actually doesn't matter which order we pick, as long as each signed formula has its corresponding rule applied in every branch. So let's pick the first one. The $\vee\mathbb{T}$ rule allows the tableau to branch, and the two conclusions of the rule

will be the new signed formulas added to the two new branches. This results in:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$ ✓	Assumption
2.	$\mathbb{T}\,\neg\varphi \vee \psi$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,(\varphi \to \psi)$	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\neg\varphi$ | $\mathbb{T}\,\psi$	$\vee\mathbb{T}\,2$

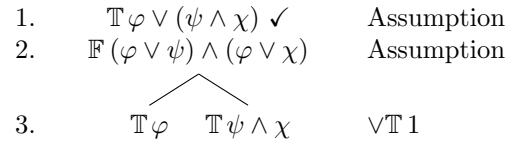
We have not applied the $\to\mathbb{F}$ rule to line 3 yet: let's do that now. To save time, we apply it to both branches. Recall that we write a checkmark next to a signed formula only if we have applied the corresponding rule in every open branch. So it's a good idea to apply a rule at the end of every branch that contains the signed formula the rule applies to. That way we won't have to return to that signed formula lower down in the various branches.

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$ ✓	Assumption
2.	$\mathbb{T}\,\neg\varphi \vee \psi$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,(\varphi \to \psi)$ ✓	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\neg\varphi$ | $\mathbb{T}\,\psi$	$\vee\mathbb{T}\,2$
5.	$\mathbb{T}\,\varphi$ | $\mathbb{T}\,\varphi$	$\to\mathbb{F}\,3$
6.	$\mathbb{F}\,\psi$ | $\mathbb{F}\,\psi$	$\to\mathbb{F}\,3$
	| $\otimes$	

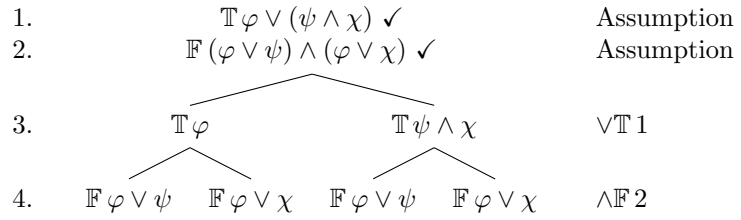
The right branch is now closed. On the left branch, we can still apply the $\neg\mathbb{T}$ rule to line 4. This results in $\mathbb{F}\,\varphi$ and closes the left branch:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$ ✓	Assumption
2.	$\mathbb{T}\,\neg\varphi \vee \psi$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,(\varphi \to \psi)$ ✓	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\neg\varphi$ | $\mathbb{T}\,\psi$	$\vee\mathbb{T}\,2$
5.	$\mathbb{T}\,\varphi$ | $\mathbb{T}\,\varphi$	$\to\mathbb{F}\,3$
6.	$\mathbb{F}\,\psi$ | $\mathbb{F}\,\psi$	$\to\mathbb{F}\,3$
7.	$\mathbb{F}\,\varphi$ | $\otimes$	$\neg\mathbb{T}\,4$
	$\otimes$	

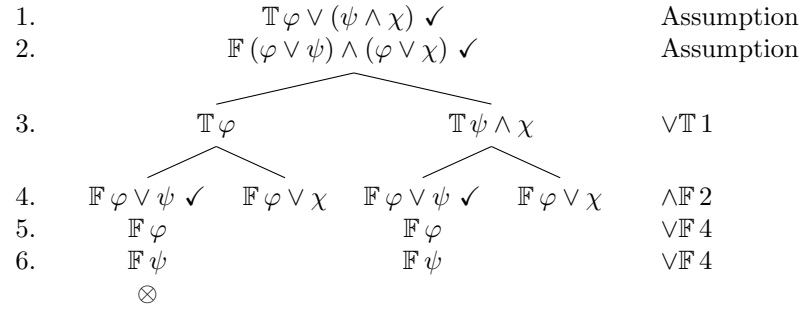
Example 21.6. We can give tableaux for any number of signed formulas as assumptions. Often it is also necessary to apply more than one rule that allows branching; and in general a tableau can have any number of branches. For instance, consider a tableau for $\{\mathbb{T}\,\varphi \vee (\psi \wedge \chi), \mathbb{F}\,(\varphi \vee \psi) \wedge (\varphi \vee \chi)\}$. We start by applying the $\vee\mathbb{T}$ to the first assumption:



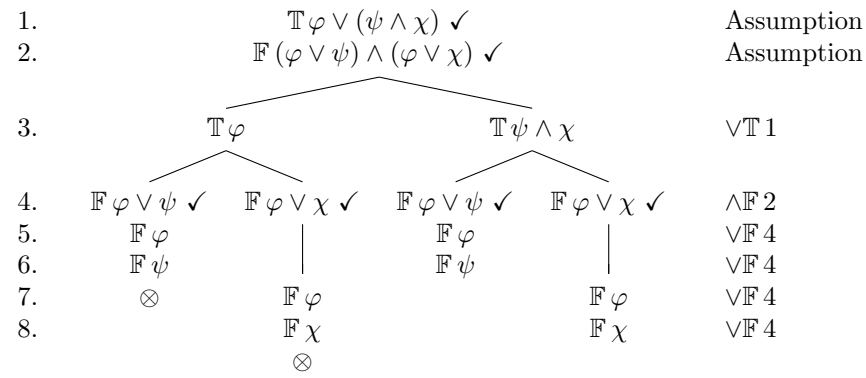
Now we can apply the $\wedge\mathbb{F}$ rule to line 2. We do this on both branches simultaneously, and can therefore check off line 2:



Now we can apply $\vee\mathbb{F}$ to all the branches containing $\varphi \vee \psi$:

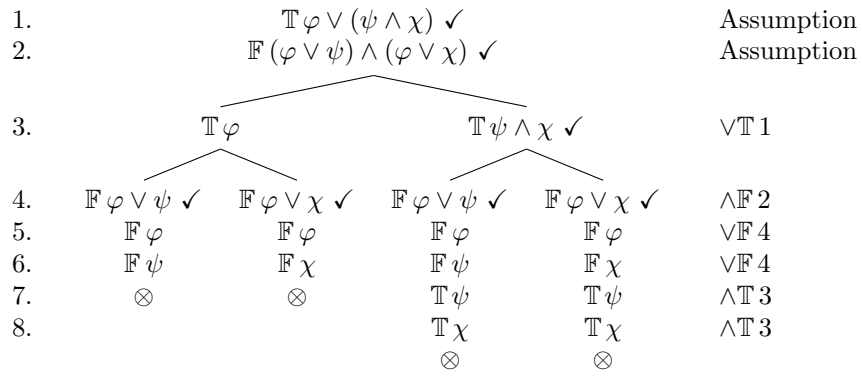


The leftmost branch is now closed. Let's now apply $\vee\mathbb{F}$ to $\varphi \vee \chi$:

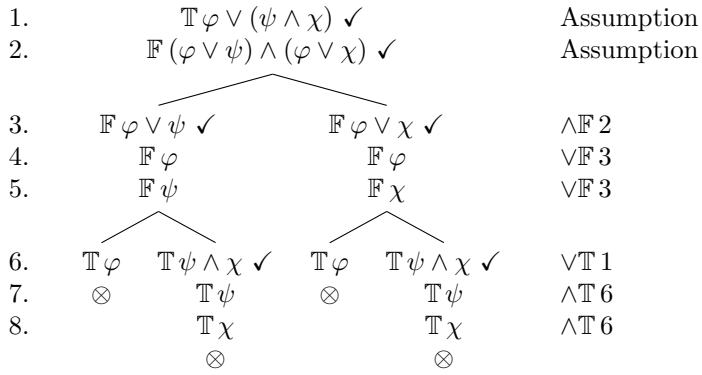


Note that we moved the result of applying $\vee\mathbb{F}$ a second time below for clarity. In this instance it would not have been needed, since the justifications would have been the same.

Two branches remain open, and $\mathbb{T}\,\psi \wedge \chi$ on line 3 remains unchecked. We apply $\wedge\mathbb{T}$ to it to obtain a closed tableau:



For comparison, here's a closed tableau for the same set of assumptions in which the rules are applied in a different order:



Problem 21.1. Give closed tableaux of the following:

1. $\mathbb{T}\,\varphi \wedge (\psi \wedge \chi), \mathbb{F}\,(\varphi \wedge \psi) \wedge \chi$.
2. $\mathbb{T}\,\varphi \vee (\psi \vee \chi), \mathbb{F}\,(\varphi \vee \psi) \vee \chi$.
3. $\mathbb{T}\,\varphi \to (\psi \to \chi), \mathbb{F}\,\psi \to (\varphi \to \chi)$.
4. $\mathbb{T}\,\varphi, \mathbb{F}\,\neg\neg\varphi$.

Problem 21.2. Give closed tableaux of the following:

1. $\mathbb{T}\,(\varphi \vee \psi) \to \chi, \mathbb{F}\,\varphi \to \chi$.
2. $\mathbb{T}\,(\varphi \to \chi) \wedge (\psi \to \chi), \mathbb{F}\,(\varphi \vee \psi) \to \chi$.
3. $\mathbb{F}\,\neg(\varphi \wedge \neg\varphi)$.

4. $\mathbb{T}\,\psi \to \varphi, \mathbb{F}\,\neg\varphi \to \neg\psi$.
5. $\mathbb{F}\,(\varphi \to \neg\varphi) \to \neg\varphi$.
6. $\mathbb{F}\,\neg(\varphi \to \psi) \to \neg\psi$.
7. $\mathbb{T}\,\varphi \to \chi, \mathbb{F}\,\neg(\varphi \wedge \neg\chi)$.
8. $\mathbb{T}\,\varphi \wedge \neg\chi, \mathbb{F}\,\neg(\varphi \to \chi)$.
9. $\mathbb{T}\,\varphi \vee \psi, \neg\psi, \mathbb{F}\,\varphi$.
10. $\mathbb{T}\,\neg\varphi \vee \neg\psi, \mathbb{F}\,\neg(\varphi \wedge \psi)$.
11. $\mathbb{F}\,(\neg\varphi \wedge \neg\psi) \to \neg(\varphi \vee \psi)$.
12. $\mathbb{F}\,\neg(\varphi \vee \psi) \to (\neg\varphi \wedge \neg\psi)$.

Problem 21.3. Give closed tableaux of the following:

1. $\mathbb{T}\,\neg(\varphi \to \psi), \mathbb{F}\,\varphi$.
2. $\mathbb{T}\,\neg(\varphi \wedge \psi), \mathbb{F}\,\neg\varphi \vee \neg\psi$.
3. $\mathbb{T}\,\varphi \to \psi, \mathbb{F}\,\neg\varphi \vee \psi$.
4. $\mathbb{F}\,\neg\neg\varphi \to \varphi$.
5. $\mathbb{T}\,\varphi \to \psi, \mathbb{T}\,\neg\varphi \to \psi, \mathbb{F}\,\psi$.
6. $\mathbb{T}\,(\varphi \wedge \psi) \to \chi, \mathbb{F}\,(\varphi \to \chi) \vee (\psi \to \chi)$.
7. $\mathbb{T}\,(\varphi \to \psi) \to \varphi, \mathbb{F}\,\varphi$.
8. $\mathbb{F}\,(\varphi \to \psi) \vee (\psi \to \chi)$.

---

content/first-order-logic/tableaux/proving-things-quant.tex

21.6 Tableaux with Quantifiers

fol:tab:prq:sec

Example 21.7. When dealing with quantifiers, we have to make sure not to violate the eigenvariable condition, and sometimes this requires us to play around with the order of carrying out certain inferences. In general, it helps to try and take care of rules subject to the eigenvariable condition first (they will be higher up in the finished tableau).

Let's see how we'd give a tableau for the sentence $\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$. Starting as usual, we start by recording the assumption,

1. $\mathbb{F}\,\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$ Assumption

Since the main operator is $\to$, we apply the $\to\mathbb{F}$:

1.	$\mathbb{F}\,\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$ ✓	Assumption
2.	$\mathbb{T}\,\exists x\,\neg\varphi(x)$	$\to\mathbb{F}$ 1
3.	$\mathbb{F}\,\neg\forall x\,\varphi(x)$	$\to\mathbb{F}$ 1

The next line to deal with is 2. We use $\exists\mathbb{T}$. This requires a new constant symbol; since no constant symbols yet occur, we can pick any one, say, a.

1.	$\mathbb{F}\,\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$ ✓	Assumption
2.	$\mathbb{T}\,\exists x\,\neg\varphi(x)$ ✓	$\to\mathbb{F}$ 1
3.	$\mathbb{F}\,\neg\forall x\,\varphi(x)$	$\to\mathbb{F}$ 1
4.	$\mathbb{T}\,\neg\varphi(a)$	$\exists\mathbb{T}$ 2

Now we apply $\neg\mathbb{F}$ to line 3:

1.	$\mathbb{F}\,\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$ ✓	Assumption
2.	$\mathbb{T}\,\exists x\,\neg\varphi(x)$ ✓	$\to\mathbb{F}$ 1
3.	$\mathbb{F}\,\neg\forall x\,\varphi(x)$ ✓	$\to\mathbb{F}$ 1
4.	$\mathbb{T}\,\neg\varphi(a)$	$\exists\mathbb{T}$ 2
5.	$\mathbb{T}\,\forall x\,\varphi(x)$	$\neg\mathbb{F}$ 3

We obtain a closed tableau by applying $\neg\mathbb{T}$ to line 4, followed by $\forall\mathbb{T}$ to line 5.

1.	$\mathbb{F}\,\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$ ✓	Assumption
2.	$\mathbb{T}\,\exists x\,\neg\varphi(x)$ ✓	$\to\mathbb{F}$ 1
3.	$\mathbb{F}\,\neg\forall x\,\varphi(x)$ ✓	$\to\mathbb{F}$ 1
4.	$\mathbb{T}\,\neg\varphi(a)$	$\exists\mathbb{T}$ 2
5.	$\mathbb{T}\,\forall x\,\varphi(x)$	$\neg\mathbb{F}$ 3
6.	$\mathbb{F}\,\varphi(a)$	$\neg\mathbb{T}$ 4
7.	$\mathbb{T}\,\varphi(a)$	$\forall\mathbb{T}$ 5
	$\otimes$	

Example 21.8. Let's see how we'd give a tableau for the set

$$\mathbb{F}\,\exists x\,\chi(x,b), \mathbb{T}\,\exists x\,(\varphi(x) \wedge \psi(x)), \mathbb{T}\forall x\,(\psi(x) \to \chi(x,b)).$$

Starting as usual, we start with the assumptions:

1.	$\mathbb{F}\,\exists x\,\chi(x,b)$	Assumption
2.	$\mathbb{T}\,\exists x\,(\varphi(x) \wedge \psi(x))$	Assumption
3.	$\mathbb{T}\forall x\,(\psi(x) \to \chi(x,b))$	Assumption

We should always apply a rule with the eigenvariable condition first; in this case that would be $\exists\mathbb{T}$ to line 2. Since the assumptions contain the constant symbol b, we have to use a different one; let's pick a again.

1.	$\mathbb{F}\,\exists x\,\chi(x,b)$	Assumption
2.	$\mathbb{T}\,\exists x\,(\varphi(x) \land \psi(x))$ ✓	Assumption
3.	$\mathbb{T}\,\forall x\,(\psi(x) \to \chi(x,b))$	Assumption
4.	$\mathbb{T}\,\varphi(a) \land \psi(a)$	$\exists\mathbb{T}\,2$

If we now apply $\exists\mathbb{F}$ to line 1 or $\forall\mathbb{T}$ to line 3, we have to decide which term t to substitute for x. Since there is no eigenvariable condition for these rules, we can pick any term we like. In some cases we may even have to apply the rule several times with different ts. But as a general rule, it pays to pick one of the terms already occurring in the tableau—in this case, a and b—and in this case we can guess that a will be more likely to result in a closed branch.

1.	$\mathbb{F}\,\exists x\,\chi(x,b)$	Assumption
2.	$\mathbb{T}\,\exists x\,(\varphi(x) \land \psi(x))$ ✓	Assumption
3.	$\mathbb{T}\,\forall x\,(\psi(x) \to \chi(x,b))$	Assumption
4.	$\mathbb{T}\,\varphi(a) \land \psi(a)$	$\exists\mathbb{T}\,2$
5.	$\mathbb{F}\,\chi(a,b)$	$\exists\mathbb{F}\,1$
6.	$\mathbb{T}\,\psi(a) \to \chi(a,b)$	$\forall\mathbb{T}\,3$

We don't check the signed formulas in lines 1 and 3, since we may have to use them again. Now apply $\land\mathbb{T}$ to line 4:

1.	$\mathbb{F}\,\exists x\,\chi(x,b)$	Assumption
2.	$\mathbb{T}\,\exists x\,(\varphi(x) \land \psi(x))$ ✓	Assumption
3.	$\mathbb{T}\,\forall x\,(\psi(x) \to \chi(x,b))$	Assumption
4.	$\mathbb{T}\,\varphi(a) \land \psi(a)$ ✓	$\exists\mathbb{T}\,2$
5.	$\mathbb{F}\,\chi(a,b)$	$\exists\mathbb{F}\,1$
6.	$\mathbb{T}\,\psi(a) \to \chi(a,b)$	$\forall\mathbb{T}\,3$
7.	$\mathbb{T}\,\varphi(a)$	$\land\mathbb{T}\,4$
8.	$\mathbb{T}\,\psi(a)$	$\land\mathbb{T}\,4$

If we now apply $\to\mathbb{T}$ to line 6, the tableau closes:

1.	$\mathbb{F}\,\exists x\,\chi(x,b)$	Assumption
2.	$\mathbb{T}\,\exists x\,(\varphi(x) \land \psi(x))$ ✓	Assumption
3.	$\mathbb{T}\,\forall x\,(\psi(x) \to \chi(x,b))$	Assumption
4.	$\mathbb{T}\,\varphi(a) \land \psi(a)$ ✓	$\exists\mathbb{T}\,2$
5.	$\mathbb{F}\,\chi(a,b)$	$\exists\mathbb{F}\,1$
6.	$\mathbb{T}\,\psi(a) \to \chi(a,b)$ ✓	$\forall\mathbb{T}\,3$
7.	$\mathbb{T}\,\varphi(a)$	$\land\mathbb{T}\,4$
8.	$\mathbb{T}\,\psi(a)$	$\land\mathbb{T}\,4$
9.	$\mathbb{F}\,\psi(a)$ ⊗ | $\mathbb{T}\,\chi(a,b)$ ⊗	$\to\mathbb{T}\,6$

Example 21.9. We construct a tableau for the set

$$\mathbb{T}\forall x\,\varphi(x), \mathbb{T}\forall x\,\varphi(x) \to \exists y\,\psi(y), \mathbb{T}\neg\exists y\,\psi(y).$$

Starting as usual, we write down the assumptions:

1.	$\mathbb{T}\forall x\,\varphi(x)$	Assumption
2.	$\mathbb{T}\forall x\,\varphi(x) \to \exists y\,\psi(y)$	Assumption
3.	$\mathbb{T}\neg\exists y\,\psi(y)$	Assumption

We begin by applying the $\neg\mathbb{T}$ rule to line 3. A corollary to the rule "always apply rules with eigenvariable conditions first" is "defer applying quantifier rules without eigenvariable conditions until needed." Also, defer rules that result in a split.

1.	$\mathbb{T}\forall x\,\varphi(x)$	Assumption
2.	$\mathbb{T}\forall x\,\varphi(x) \to \exists y\,\psi(y)$	Assumption
3.	$\mathbb{T}\neg\exists y\,\psi(y)$ ✓	Assumption
4.	$\mathbb{F}\exists y\,\psi(y)$	$\neg\mathbb{T}\,3$

The new line 4 requires $\exists\mathbb{F}$, a quantifier rule without the eigenvariable condition. So we defer this in favor of using $\to\mathbb{T}$ on line 2.

1.	$\mathbb{T}\forall x\,\varphi(x)$		Assumption
2.	$\mathbb{T}\forall x\,\varphi(x) \to \exists y\,\psi(y)$ ✓		Assumption
3.	$\mathbb{T}\neg\exists y\,\psi(y)$ ✓		Assumption
4.	$\mathbb{F}\exists y\,\psi(y)$		$\neg\mathbb{T}\,3$
5.	$\mathbb{F}\forall x\,\varphi(x)$	$\mathbb{T}\exists y\,\psi(y)$	$\to\mathbb{T}\,2$

Both new signed formulas require rules with eigenvariable conditions, so these should be next:

1.	$\mathbb{T}\forall x\,\varphi(x)$		Assumption
2.	$\mathbb{T}\forall x\,\varphi(x) \to \exists y\,\psi(y)$ ✓		Assumption
3.	$\mathbb{T}\neg\exists y\,\psi(y)$ ✓		Assumption
4.	$\mathbb{F}\exists y\,\psi(y)$		$\neg\mathbb{T}\,3$
5.	$\mathbb{F}\forall x\,\varphi(x)$ ✓	$\mathbb{T}\exists y\,\psi(y)$ ✓	$\to\mathbb{T}\,2$
6.	$\mathbb{F}\,\varphi(b)$	$\mathbb{T}\,\psi(c)$	$\forall\mathbb{F}\,5$; $\exists\mathbb{T}\,5$

To close the branches, we have to use the signed formulas on lines 1 and 3. The corresponding rules ($\forall\mathbb{T}$ and $\exists\mathbb{F}$) don't have eigenvariable conditions, so we are free to pick whichever terms are suitable. In this case, that's b and c, respectively.

1.	$\mathbb{T}\forall x\,\varphi(x)$		Assumption
2.	$\mathbb{T}\forall x\,\varphi(x) \to \exists y\,\psi(y)$ ✓		Assumption
3.	$\mathbb{T}\neg\exists y\,\psi(y)$ ✓		Assumption
4.	$\mathbb{F}\,\exists y\,\psi(y)$		$\neg\mathbb{T}\,3$
5.	$\mathbb{F}\,\forall x\,\varphi(x)$ ✓	$\mathbb{T}\exists y\,\psi(y)$ ✓	$\to\mathbb{T}\,2$
6.	$\mathbb{F}\,\varphi(b)$	$\mathbb{T}\,\psi(c)$	$\forall\mathbb{F}\,5;\ \exists\mathbb{T}\,5$
7.	$\mathbb{T}\,\varphi(b)$	$\mathbb{F}\,\psi(c)$	$\forall\mathbb{T}\,1;\ \exists\mathbb{F}\,4$
	$\otimes$	$\otimes$	

Problem 21.4. Give closed tableaux of the following:

1. $\mathbb{F}\,(\forall x\,\varphi(x) \land \forall y\,\psi(y)) \to \forall z\,(\varphi(z) \land \psi(z))$.
2. $\mathbb{F}\,(\exists x\,\varphi(x) \lor \exists y\,\psi(y)) \to \exists z\,(\varphi(z) \lor \psi(z))$.
3. $\mathbb{T}\forall x\,(\varphi(x) \to \psi), \mathbb{F}\,\exists y\,\varphi(y) \to \psi$.
4. $\mathbb{T}\forall x\,\neg\varphi(x), \mathbb{F}\,\neg\exists x\,\varphi(x)$.
5. $\mathbb{F}\,\neg\exists x\,\varphi(x) \to \forall x\,\neg\varphi(x)$.
6. $\mathbb{F}\,\neg\exists x\,\forall y\,((\varphi(x,y) \to \neg\varphi(y,y)) \land (\neg\varphi(y,y) \to \varphi(x,y)))$.

Problem 21.5. Give closed tableaux of the following:

1. $\mathbb{F}\,\neg\forall x\,\varphi(x) \to \exists x\,\neg\varphi(x)$.
2. $\mathbb{T}\,(\forall x\,\varphi(x) \to \psi), \mathbb{F}\,\exists y\,(\varphi(y) \to \psi)$.
3. $\mathbb{F}\,\exists x\,(\varphi(x) \to \forall y\,\varphi(y))$.

---

content/first-order-logic/tableaux/proof-theoretic-notions.tex

21.7 Proof-Theoretic Notions

fol:tab:ptn:sec

This section collects the definitions of the provability relation and consistency for tableaux.

Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the existence of certain closed tableaux. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

explanation

Definition 21.10 (Theorems). A sentence φ is a *theorem* if there is a closed tableau for $\mathbb{F}\,\varphi$. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 21.11 (Derivability). A sentence φ is *derivable from* a set of sentences Γ, $\Gamma \vdash \varphi$ iff there is a finite set $\{\psi_1, \ldots, \psi_n\} \subseteq \Gamma$ and a closed tableau for the set

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}.$$

If φ is not derivable from Γ we write $\Gamma \nvdash \varphi$.

Definition 21.12 (Consistency). A set of sentences Γ is *inconsistent* iff there is a finite set $\{\psi_1, \ldots, \psi_n\} \subseteq \Gamma$ and a closed tableau for the set

$$\{\mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}.$$

If Γ is not inconsistent, we say it is *consistent*.

Proposition 21.13 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.* *fol:tab:ptn: prop:reflexivity*

Proof. If $\varphi \in \Gamma$, $\{\varphi\}$ is a finite subset of Γ and the tableau

1.	$\mathbb{F}\,\varphi$	Assumption
2.	$\mathbb{T}\,\varphi$	Assumption
	$\otimes$	

is closed. □

Proposition 21.14 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.* *fol:tab:ptn: prop:monotonicity*

Proof. Any finite subset of Γ is also a finite subset of Δ. □

Proposition 21.15 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.* *fol:tab:ptn: prop:transitivity*

Proof. If $\{\varphi\} \cup \Delta \vdash \psi$, then there is a finite subset $\Delta_0 = \{\chi_1, \ldots, \chi_n\} \subseteq \Delta$ such that

$$\{\mathbb{F}\,\psi, \mathbb{T}\,\varphi, \mathbb{T}\,\chi_1, \ldots, \mathbb{T}\,\chi_n\}$$

has a closed tableau. If $\Gamma \vdash \varphi$ then there are $\theta_1, \ldots, \theta_m \subseteq \Gamma$ such that

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\theta_1, \ldots, \mathbb{T}\,\theta_m\}$$

has a closed tableau.

Now consider the tableau with assumptions

$$\mathbb{F}\,\psi, \mathbb{T}\,\chi_1, \ldots, \mathbb{T}\,\chi_n, \mathbb{T}\,\theta_1, \ldots, \mathbb{T}\,\theta_m.$$

Apply the Cut rule on φ. This generates two branches, one has $\mathbb{T}\varphi$ in it, the other $\mathbb{F}\varphi$. Thus, on the one branch, all of

$$\{\mathbb{F}\psi, \mathbb{T}\varphi, \mathbb{T}\chi_1, \ldots, \mathbb{T}\chi_n\}$$

are available. Since there is a closed tableau for these assumptions, we can attach it to that branch; every branch through $\mathbb{T}\varphi$ closes. On the other branch, all of

$$\{\mathbb{F}\varphi, \mathbb{T}\theta_1, \ldots, \mathbb{T}\theta_m\}$$

are available, so we can also complete the other side to obtain a closed tableau. This shows $\Gamma \cup \Delta \vdash \psi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \ldots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

fol:tab:ptn:prop:incons

Proposition 21.16. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every sentence φ.*

Proof. Exercise. □

Problem 21.6. Prove Proposition 21.16

fol:tab:ptn:prop:proves-compact

Proposition 21.17 (Compactness).

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof. 1. If $\Gamma \vdash \varphi$, then there is a finite subset $\Gamma_0 = \{\psi_1, \ldots, \psi_n\}$ and a closed tableau for

$$\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n\}$$

This tableau also shows $\Gamma_0 \vdash \varphi$.

2. If Γ is inconsistent, then for some finite subset $\Gamma_0 = \{\psi_1, \ldots, \psi_n\}$ there is a closed tableau for

$$\{\mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n\}$$

This closed tableau shows that Γ_0 is inconsistent. □

content/first-order-logic/tableaux/provability-consistency.tex

21.8 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem. fol:tab:prv: sec

Proposition 21.18. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.* fol:tab:prv: prop:provability-contr

Proof. There are finite $\Gamma_0 = \{\psi_1, \ldots, \psi_n\}$ and $\Gamma_1 = \{\chi_1, \ldots, \chi_n\} \subseteq \Gamma$ such that

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$
$$\{\mathbb{T}\,\varphi, \mathbb{T}\,\chi_1, \ldots, \mathbb{T}\,\chi_m\}$$

have closed tableaux. Using the Cut rule on φ we can combine these into a single closed tableau that shows $\Gamma_0 \cup \Gamma_1$ is inconsistent. Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$, hence Γ is inconsistent. □

Proposition 21.19. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.* fol:tab:prv: prop:prov-incons

Proof. First suppose $\Gamma \vdash \varphi$, i.e., there is a closed tableau for

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$

Using the $\neg\mathbb{T}$ rule, this can be turned into a closed tableau for

$$\{\mathbb{T}\,\neg\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}.$$

On the other hand, if there is a closed tableau for the latter, we can turn it into a closed tableau of the former by removing every formula that results from $\neg\mathbb{T}$ applied to the first assumption $\mathbb{T}\,\neg\varphi$ as well as that assumption, and adding the assumption $\mathbb{F}\,\varphi$. For if a branch was closed before because it contained the conclusion of $\neg\mathbb{T}$ applied to $\mathbb{T}\,\neg\varphi$, i.e., $\mathbb{F}\,\varphi$, the corresponding branch in the new tableau is also closed. If a branch in the old tableau was closed because it contained the assumption $\mathbb{T}\,\neg\varphi$ as well as $\mathbb{F}\,\neg\varphi$ we can turn it into a closed branch by applying $\neg\mathbb{F}$ to $\mathbb{F}\,\neg\varphi$ to obtain $\mathbb{T}\,\varphi$. This closes the branch since we added $\mathbb{F}\,\varphi$ as an assumption. □

Problem 21.7. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

Proposition 21.20. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.* fol:tab:prv: prop:explicit-inc

Proof. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there are $\psi_1, \ldots, \psi_n \in \Gamma$ such that

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$

has a closed tableau. Replace the assumption $\mathbb{F}\,\varphi$ by $\mathbb{T}\,\neg\varphi$, and insert the conclusion of $\neg\mathbb{T}$ applied to $\mathbb{F}\,\varphi$ after the assumptions. Any sentence in the tableau justified by appeal to line 1 in the old tableau is now justified by appeal to line $n+1$. So if the old tableau was closed, the new one is. It shows that Γ is inconsistent, since all assumptions are in Γ. □

fol:tab:prv: prop:provability-exhaustive

Proposition 21.21. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

Proof. If there are $\psi_1, \ldots, \psi_n \in \Gamma$ and $\chi_1, \ldots, \chi_m \in \Gamma$ such that

$$\{\mathbb{T}\varphi, \mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n\} \text{ and}$$
$$\{\mathbb{T}\neg\varphi, \mathbb{T}\chi_1, \ldots, \mathbb{T}\chi_m\}$$

both have closed tableaux, we can construct a single, combined tableau that shows that Γ is inconsistent by using as assumptions $\mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n$ together with $\mathbb{T}\chi_1, \ldots, \mathbb{T}\chi_m$, followed by an application of the Cut rule. This yields two branches, one starting with $\mathbb{T}\varphi$, the other with $\mathbb{F}\varphi$.

On the left left side, add the part of the first tableau below its assumptions. Here, every rule application is still correct, since each of the assumptions of the first tableau, including $\mathbb{T}\varphi$, is available. Thus, every branch below $\mathbb{T}\varphi$ closes.

On the right side, add the part of the second tableau below its assumption, with the results of any applications of $\neg\mathbb{T}$ to $\mathbb{T}\neg\varphi$ removed. The conclusion of $\neg\mathbb{T}$ to $\mathbb{T}\neg\varphi$ is $\mathbb{F}\varphi$, which is nevertheless available, as it is the conclusion of the Cut rule on the right side of the combined tableau.

If a branch in the second tableau was closed because it contained the assumption $\mathbb{T}\neg\varphi$ (which no longer appears as an assumption in the combined tableau) as well as $\mathbb{F}\neg\varphi$, we can applying $\neg\mathbb{F}$ to $\mathbb{F}\neg\varphi$ to obtain $\mathbb{T}\varphi$. Now the corresponding branch in the combined tableau also closes, because it contains the right-hand conclusion of the Cut rule, $\mathbb{F}\varphi$. If a branch in the second tableau closed for any other reason, the corresponding branch in the combined tableau also closes, since any signed formulas other than $\mathbb{T}\neg\varphi$ occurring on the branch in the old, second tableau also occur on the corresponding branch in the combined tableau. □

---

content/first-order-logic/tableaux/provability-propositional.tex

21.9 Derivability and the Propositional Connectives

fol:tab:ppr: sec

We establish that the derivability relation $\vdash$ of tableaux is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. explanation

fol:tab:ppr: prop:provability-land

Proposition 21.22.

fol:tab:ppr: prop:provability-land-left

1. *Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$.*

fol:tab:ppr: prop:provability-land-right

2. *$\varphi, \psi \vdash \varphi \wedge \psi$.*

Proof. 1. Both $\{\mathbb{F}\varphi, \mathbb{T}\varphi \wedge \psi\}$ and $\{\mathbb{F}\psi, \mathbb{T}\varphi \wedge \psi\}$ have closed tableaux

1.	$\mathbb{F}\,\varphi$	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	Assumption
3.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}\,2$
4.	$\mathbb{T}\psi$	$\wedge\mathbb{T}\,2$
	$\otimes$	

1.	$\mathbb{F}\,\psi$	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	Assumption
3.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}\,2$
4.	$\mathbb{T}\psi$	$\wedge\mathbb{T}\,2$
	$\otimes$	

2. Here is a closed tableau for $\{\mathbb{T}\,\varphi, \mathbb{T}\,\psi, \mathbb{F}\,\varphi \wedge \psi\}$:

1.	$\mathbb{F}\,\varphi \wedge \psi$		Assumption
2.	$\mathbb{T}\,\varphi$		Assumption
3.	$\mathbb{T}\,\psi$		Assumption
4.	$\mathbb{F}\,\varphi$	$\mathbb{F}\,\psi$	$\wedge\mathbb{F}\,1$
	$\otimes$	$\otimes$	

Proposition 21.23.

fol:tab:ppr: prop:provability-lor

1. $\{\varphi \vee \psi, \neg\varphi, \neg\psi\}$ is inconsistent.

2. Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.

Proof. 1. We give a closed tableau of $\{\mathbb{T}\,\varphi \vee \psi, \mathbb{T}\,\neg\varphi, \mathbb{T}\,\neg\psi\}$:

1.	$\mathbb{T}\,\varphi \vee \psi$		Assumption
2.	$\mathbb{T}\,\neg\varphi$		Assumption
3.	$\mathbb{T}\,\neg\psi$		Assumption
4.	$\mathbb{F}\,\varphi$		$\neg\mathbb{T}\,2$
5.	$\mathbb{F}\,\psi$		$\neg\mathbb{T}\,3$
6.	$\mathbb{T}\,\varphi$	$\mathbb{T}\,\psi$	$\vee\mathbb{T}\,1$
	$\otimes$	$\otimes$	

2. Both $\{\mathbb{F}\,\varphi \vee \psi, \mathbb{T}\,\varphi\}$ and $\{\mathbb{F}\,\varphi \vee \psi, \mathbb{T}\,\psi\}$ have closed tableaux:

1.	$\mathbb{F}\,\varphi \vee \psi$	Assumption
2.	$\mathbb{T}\,\varphi$	Assumption
3.	$\mathbb{F}\varphi$	$\vee\mathbb{F}\,1$
4.	$\mathbb{F}\psi$	$\vee\mathbb{F}\,1$
	$\otimes$	

1.	$\mathbb{F}\,\varphi \vee \psi$	Assumption
2.	$\mathbb{T}\,\psi$	Assumption
3.	$\mathbb{F}\varphi$	$\vee\mathbb{F}\,1$
4.	$\mathbb{F}\psi$	$\vee\mathbb{F}\,1$
	$\otimes$	

fol:tab:ppr: prop:provability-lif *fol:tab:ppr: prop:provability-lif-left* *fol:tab:ppr: prop:provability-lif-right*

Proposition 21.24.

1. $\varphi, \varphi \to \psi \vdash \psi$.
2. *Both* $\neg\varphi \vdash \varphi \to \psi$ *and* $\psi \vdash \varphi \to \psi$.

Proof. 1. $\{\mathbb{F}\,\psi, \mathbb{T}\,\varphi \to \psi, \mathbb{T}\,\varphi\}$ has a closed tableau:

1.	$\mathbb{F}\,\psi$	Assumption
2.	$\mathbb{T}\,\varphi \to \psi$	Assumption
3.	$\mathbb{T}\,\varphi$	Assumption
4.	$\mathbb{F}\,\varphi$ $\quad$ $\mathbb{T}\,\psi$	$\to\mathbb{T}\,2$
	$\otimes$ $\quad$ $\otimes$	

2. Both $\{\mathbb{F}\,\varphi \to \psi, \mathbb{T}\,\neg\varphi\}$ and $\{\mathbb{F}\,\varphi \to \psi, \mathbb{T}\,\psi\}$ have closed tableaux:

1.	$\mathbb{F}\,\varphi \to \psi$	Assumption
2.	$\mathbb{T}\,\neg\varphi$	Assumption
3.	$\mathbb{T}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{F}\,\psi$	$\to\mathbb{F}\,1$
5.	$\mathbb{F}\,\varphi$	$\neg\mathbb{T}\,2$
	$\otimes$	

1.	$\mathbb{F}\,\varphi \to \psi$	Assumption
2.	$\mathbb{T}\,\psi$	Assumption
3.	$\mathbb{T}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{F}\,\psi$	$\to\mathbb{F}\,1$
	$\otimes$	

21.10 Derivability and the Quantifiers

explanation

The completeness theorem also requires that the tableaux rules yield the facts about $\vdash$ established in this section.

fol:tab:qpr:sec

Theorem 21.25. *If c is a constant not occurring in Γ or $\varphi(x)$ and $\Gamma \vdash \varphi(c)$, then $\Gamma \vdash \forall x\, \varphi(x)$.*

fol:tab:qpr:thm:strong-generalization

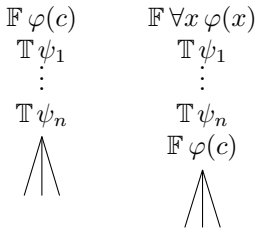
Proof. Suppose $\Gamma \vdash \varphi(c)$, i.e., there are $\psi_1, \ldots, \psi_n \in \Gamma$ and a closed tableau for

$$\{\mathbb{F}\, \varphi(c), \mathbb{T}\, \psi_1, \ldots, \mathbb{T}\, \psi_n\}.$$

We have to show that there is also a closed tableau for

$$\{\mathbb{F}\, \forall x\, \varphi(x), \mathbb{T}\, \psi_1, \ldots, \mathbb{T}\, \psi_n\}.$$

Take the closed tableau and replace the first assumption with $\mathbb{F}\, \forall x\, \varphi(x)$, and insert $\mathbb{F}\, \varphi(c)$ after the assumptions.



The tableau is still closed, since all sentences available as assumptions before are still available at the top of the tableau. The inserted line is the result of a correct application of $\forall\mathbb{F}$, since the constant symbol c does not occur in ψ_1, $\ldots$, ψ_n or $\forall x\, \varphi(x)$, i.e., it does not occur above the inserted line in the new tableau. □

Proposition 21.26.

fol:tab:qpr:prop:provability-quantifiers

1. $\varphi(t) \vdash \exists x\, \varphi(x)$.
2. $\forall x\, \varphi(x) \vdash \varphi(t)$.

Proof. 1. A closed tableau for $\mathbb{F}\, \exists x\, \varphi(x), \mathbb{T}\, \varphi(t)$ is:

1.	$\mathbb{F}\, \exists x\, \varphi(x)$	Assumption
2.	$\mathbb{T}\, \varphi(t)$	Assumption
3.	$\mathbb{F}\, \varphi(t)$	$\exists\mathbb{F}\, 1$
	$\otimes$	

2. A closed tableau for $\mathbb{F}\, \varphi(t), \mathbb{T}\, \forall x\, \varphi(x)$, is:

1.	$\mathbb{F}\,\varphi(t)$	Assumption
2.	$\mathbb{T}\,\forall x\,\varphi(x)$	Assumption
3.	$\mathbb{T}\,\varphi(t)$	$\forall\mathbb{T}\,2$
	$\otimes$	

---

content/first-order-logic/tableaux/soundness.tex

21.11 Soundness

fol:tab:sou:sec

A derivation system, such as tableaux, is *sound* if it cannot derive things that do not actually hold. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that explanation

1. every derivable φ is valid;
2. if a sentence is derivable from some others, it is also a consequence of them;
3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Because all these proof-theoretic properties are defined via closed tableaux of some kind or other, proving (1)–(3) above requires proving something about the semantic properties of closed tableaux. We will first define what it means for a signed formula to be satisfied in a structure, and then show that if a tableau is closed, no structure satisfies all its assumptions. (1)–(3) then follow as corollaries from this result.

Definition 21.27. A structure $\mathfrak{M}$ *satisfies* a signed formula $\mathbb{T}\,\varphi$ iff $\mathfrak{M} \vDash \varphi$, and it satisfies $\mathbb{F}\,\varphi$ iff $\mathfrak{M} \nvDash \varphi$. $\mathfrak{M}$ satisfies a set of signed formulas Γ iff it satisfies every $S\,\varphi \in \Gamma$. Γ is *satisfiable* if there is a structure that satisfies it, and *unsatisfiable* otherwise.

fol:tab:sou:thm:tableau-soundness

Theorem 21.28 (Soundness). *If Γ has a closed tableau, Γ is unsatisfiable.*

Proof. Let's call a branch of a tableau satisfiable iff the set of signed formulas on it is satisfiable, and let's call a tableau satisfiable if it contains at least one satisfiable branch.

We show the following: Extending a satisfiable tableau by one of the rules of inference always results in a satisfiable tableau. This will prove the theorem: any closed tableau results by applying rules of inference to the tableau consisting only of assumptions from Γ. So if Γ were satisfiable, any tableau

for it would be satisfiable. A closed tableau, however, is clearly not satisfiable: every branch contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, and no structure can both satisfy and not satisfy φ.

Suppose we have a satisfiable tableau, i.e., a tableau with at least one satisfiable branch. Applying a rule of inference either adds signed formulas to a branch, or splits a branch in two. If the tableau has a satisfiable branch which is not extended by the rule application in question, it remains a satisfiable branch in the extended tableau, so the extended tableau is satisfiable. So we only have to consider the case where a rule is applied to a satisfiable branch.

Let Γ be the set of signed formulas on that branch, and let $S\varphi \in \Gamma$ be the signed formula to which the rule is applied. If the rule does not result in a split branch, we have to show that the extended branch, i.e., Γ together with the conclusions of the rule, is still satisfiable. If the rule results in a split branch, we have to show that at least one of the two resulting branches is satisfiable.

First, we consider the possible inferences that do not result in a split branch.

1. The branch is expanded by applying $\neg\mathbb{T}$ to $\mathbb{T}\neg\psi \in \Gamma$. Then the extended branch contains the signed formulas $\Gamma \cup \{\mathbb{F}\psi\}$. Suppose $\mathfrak{M} \vDash \Gamma$. In particular, $\mathfrak{M} \vDash \neg\psi$. Thus, $\mathfrak{M} \nvDash \psi$, i.e., $\mathfrak{M}$ satisfies $\mathbb{F}\psi$.

2. The branch is expanded by applying $\neg\mathbb{F}$ to $\mathbb{F}\neg\psi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\wedge\mathbb{T}$ to $\mathbb{T}\psi \wedge \chi \in \Gamma$, which results in two new signed formulas on the branch: $\mathbb{T}\psi$ and $\mathbb{T}\chi$. Suppose $\mathfrak{M} \vDash \Gamma$, in particular $\mathfrak{M} \vDash \psi \wedge \chi$. Then $\mathfrak{M} \vDash \psi$ and $\mathfrak{M} \vDash \chi$. This means that $\mathfrak{M}$ satisfies both $\mathbb{T}\psi$ and $\mathbb{T}\chi$.

4. The branch is expanded by applying $\vee\mathbb{F}$ to $\mathbb{F}\psi \vee \chi \in \Gamma$: Exercise.

5. The branch is expanded by applying $\rightarrow\mathbb{F}$ to $\mathbb{F}\psi \rightarrow \chi \in \Gamma$: This results in two new signed formulas on the branch: $\mathbb{T}\psi$ and $\mathbb{F}\chi$. Suppose $\mathfrak{M} \vDash \Gamma$, in particular $\mathfrak{M} \nvDash \psi \rightarrow \chi$. Then $\mathfrak{M} \vDash \psi$ and $\mathfrak{M} \nvDash \chi$. This means that $\mathfrak{M}$ satisfies both $\mathbb{T}\psi$ and $\mathbb{F}\chi$.

6. The branch is expanded by applying $\forall\mathbb{T}$ to $\mathbb{T}\forall x\,\psi(x) \in \Gamma$: This results in a new signed formula $\mathbb{T}\varphi(t)$ on the branch. Suppose $\mathfrak{M} \vDash \Gamma$, in particular, $\mathfrak{M} \vDash \forall x\,\varphi(x)$. By Proposition 16.31, $\mathfrak{M} \vDash \varphi(t)$. Consequently, $\mathfrak{M}$ satisfies $\mathbb{T}\varphi(t)$.

7. The branch is expanded by applying $\forall\mathbb{F}$ to $\mathbb{F}\forall x\,\psi(x) \in \Gamma$: This results in a new signed formula $\mathbb{F}\varphi(a)$ where a is a constant symbol not occurring in Γ. Since Γ is satisfiable, there is a $\mathfrak{M}$ such that $\mathfrak{M} \vDash \Gamma$, in particular $\mathfrak{M} \nvDash \forall x\,\psi(x)$. We have to show that $\Gamma \cup \{\mathbb{F}\varphi(a)\}$ is satisfiable. To do this, we define a suitable $\mathfrak{M}'$ as follows.

   By Proposition 16.19, $\mathfrak{M} \nvDash \forall x\,\psi(x)$ iff for some s, $\mathfrak{M}, s \nvDash \psi(x)$. Now let $\mathfrak{M}'$ be just like $\mathfrak{M}$, except $a^{\mathfrak{M}'} = s(x)$. By Corollary 16.21, for any $\mathbb{T}\chi \in \Gamma$, $\mathfrak{M}' \vDash \chi$, and for any $\mathbb{F}\chi \in \Gamma$, $\mathfrak{M}' \nvDash \chi$, since a does not occur in Γ.

By Proposition 16.20, $\mathfrak{M}', s \nvDash \varphi(x)$. By Proposition 16.23, $\mathfrak{M}', s \nvDash \varphi(a)$. Since $\varphi(a)$ is a sentence, by Proposition 16.18, $\mathfrak{M}' \nvDash \varphi(a)$, i.e., $\mathfrak{M}'$ satisfies $\mathbb{F}\,\varphi(a)$.

8. The branch is expanded by applying $\exists\mathbb{T}$ to $\mathbb{T}\,\exists x\,\psi(x) \in \Gamma$: Exercise.

9. The branch is expanded by applying $\exists\mathbb{F}$ to $\mathbb{F}\,\exists x\,\psi(x) \in \Gamma$: Exercise.

Now let's consider the possible inferences that result in a split branch.

1. The branch is expanded by applying $\wedge\mathbb{F}$ to $\mathbb{F}\,\psi \wedge \chi \in \Gamma$, which results in two branches, a left one continuing through $\mathbb{F}\,\psi$ and a right one through $\mathbb{F}\,\chi$. Suppose $\mathfrak{M} \vDash \Gamma$, in particular $\mathfrak{M} \nvDash \psi \wedge \chi$. Then $\mathfrak{M} \nvDash \psi$ or $\mathfrak{M} \nvDash \chi$. In the former case, $\mathfrak{M}$ satisfies $\mathbb{F}\,\psi$, i.e., $\mathfrak{M}$ satisfies the formulas on the left branch. In the latter, $\mathfrak{M}$ satisfies $\mathbb{F}\,\chi$, i.e., $\mathfrak{M}$ satisfies the formulas on the right branch.

2. The branch is expanded by applying $\vee\mathbb{T}$ to $\mathbb{T}\,\psi \vee \chi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\to\mathbb{T}$ to $\mathbb{T}\,\psi \to \chi \in \Gamma$: Exercise.

4. The branch is expanded by Cut: This results in two branches, one containing $\mathbb{T}\,\psi$, the other containing $\mathbb{F}\,\psi$. Since $\mathfrak{M} \vDash \Gamma$ and either $\mathfrak{M} \vDash \psi$ or $\mathfrak{M} \nvDash \psi$, $\mathfrak{M}$ satisfies either the left or the right branch. □

Problem 21.8. Complete the proof of Theorem 21.28.

fol:tab:sou: cor:weak-soundness

Corollary 21.29. *If $\vdash \varphi$ then φ is valid.*

fol:tab:sou: cor:entailment-soundness

Corollary 21.30. *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.*

Proof. If $\Gamma \vdash \varphi$ then for some $\psi_1, \ldots, \psi_n \in \Gamma$, $\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$ has a closed tableau. By Theorem 21.28, every structure $\mathfrak{M}$ either makes some ψ_i false or makes φ true. Hence, if $\mathfrak{M} \vDash \Gamma$ then also $\mathfrak{M} \vDash \varphi$. □

fol:tab:sou: cor:consistency-soundness

Corollary 21.31. *If Γ is satisfiable, then it is consistent.*

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then there are $\psi_1, \ldots, \psi_n \in \Gamma$ and a closed tableau for $\{\mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$. By Theorem 21.28, there is no $\mathfrak{M}$ such that $\mathfrak{M} \vDash \psi_i$ for all $i = 1, \ldots, n$. But then Γ is not satisfiable. □

---

content/first-order-logic/tableaux/identity.tex

21.12 Tableaux with Identity predicate

Tableaux with identity predicate require additional inference rules. The rules for $=$ are (t, t_1, and t_2 are closed terms): fol:tab:identity:sec

$$\frac{}{\mathbb{T}\, t = t}\,{=} \qquad \frac{\mathbb{T}\, t_1 = t_2 \quad \mathbb{T}\, \varphi(t_1)}{\mathbb{T}\, \varphi(t_2)}\,{=}\mathbb{T} \qquad \frac{\mathbb{T}\, t_1 = t_2 \quad \mathbb{F}\, \varphi(t_1)}{\mathbb{F}\, \varphi(t_2)}\,{=}\mathbb{F}$$

Note that in contrast to all the other rules, $=\mathbb{T}$ and $=\mathbb{F}$ require that *two* signed formulas already appear on the branch, namely both $\mathbb{T}\, t_1 = t_2$ and $S\, \varphi(t_1)$.

Example 21.32. If s and t are closed terms, then $s = t, \varphi(s) \vdash \varphi(t)$:

1.	$\mathbb{F}\, \varphi(t)$	Assumption
2.	$\mathbb{T}\, s = t$	Assumption
3.	$\mathbb{T}\, \varphi(s)$	Assumption
4.	$\mathbb{T}\, \varphi(t)$	$=\mathbb{T}\, 2, 3$
	$\otimes$	

This may be familiar as the principle of substitutability of identicals, or Leibniz' Law.

Tableaux prove that $=$ is symmetric, i.e., that $s_1 = s_2 \vdash s_2 = s_1$:

1.	$\mathbb{F}\, s_2 = s_1$	Assumption
2.	$\mathbb{T}\, s_1 = s_2$	Assumption
3.	$\mathbb{T}\, s_1 = s_1$	$=$
4.	$\mathbb{T}\, s_2 = s_1$	$=\mathbb{T}\, 2, 3$
	$\otimes$	

Here, line 2 is the first prerequisite formula $\mathbb{T}\, s_1 = s_2$ of $=\mathbb{T}$. Line 3 is the second one, of the form $\mathbb{T}\, \varphi(s_2)$—think of $\varphi(x)$ as $x = s_1$, then $\varphi(s_1)$ is $s_1 = s_1$ and $\varphi(s_2)$ is $s_2 = s_1$.

They also prove that $=$ is transitive, i.e., that $s_1 = s_2, s_2 = s_3 \vdash s_1 = s_3$:

1.	$\mathbb{F}\, s_1 = s_3$	Assumption
2.	$\mathbb{T}\, s_1 = s_2$	Assumption
3.	$\mathbb{T}\, s_2 = s_3$	Assumption
4.	$\mathbb{T}\, s_1 = s_3$	$=\mathbb{T}\, 3, 2$
	$\otimes$	

In this tableau, the first prerequisite formula of $=\mathbb{T}$ is line 3, $\mathbb{T}\, s_2 = s_3$ (s_2 plays the role of t_1, and s_3 the role of t_2). The second prerequisite, of the

form $\mathbb{T}\,\varphi(s_2)$ is line 2. Here, think of $\varphi(x)$ as $s_1 = x$; that makes $\varphi(s_2)$ into $t_1 = t_2$ (i.e., line 2) and $\varphi(s_3)$ into the formula $s_1 = s_3$ in the conclusion.

Problem 21.9. Give closed tableaux for the following:

1. $\mathbb{F}\,\forall x\,\forall y\,((x = y \wedge \varphi(x)) \to \varphi(y))$

2. $\mathbb{F}\,\exists x\,(\varphi(x) \wedge \forall y\,(\varphi(y) \to y = x))$,
   $\mathbb{T}\,\exists x\,\varphi(x) \wedge \forall y\,\forall z\,((\varphi(y) \wedge \varphi(z)) \to y = z)$

---

content/first-order-logic/tableaux/soundness-identity.tex

21.13 Soundness with Identity predicate

fol:tab:sid:sec

Proposition 21.33. *Tableaux with rules for identity are sound: no closed tableau is satisfiable.*

Proof. We just have to show as before that if a tableau has a satisfiable branch, the branch resulting from applying one of the rules for $=$ to it is also satisfiable. Let Γ be the set of signed formulas on the branch, and let $\mathfrak{M}$ be a structure satisfying Γ.

Suppose the branch is expanded using $=$, i.e., by adding the signed formula $\mathbb{T}\,t = t$. Trivially, $\mathfrak{M} \vDash t = t$, so $\mathfrak{M}$ also satisfies $\Gamma \cup \{\mathbb{T}\,t = t\}$.

If the branch is expanded using $=\mathbb{T}$, we add a signed formula $S\,\varphi(t_2)$, but Γ contains both $\mathbb{T}\,t_1 = t_2$ and $\mathbb{T}\,\varphi(t_1)$. Thus we have $\mathfrak{M} \vDash t_1 = t_2$ and $\mathfrak{M} \vDash \varphi(t_1)$. Let s be a variable assignment with $s(x) = \mathrm{Val}^{\mathfrak{M}}(t_1)$. By Proposition 16.18, $\mathfrak{M}, s \vDash \varphi(t_1)$. Since $s \sim_x s$, by Proposition 16.23, $\mathfrak{M}, s \vDash \varphi(x)$. since $\mathfrak{M} \vDash t_1 = t_2$, we have $\mathrm{Val}^{\mathfrak{M}}(t_1) = \mathrm{Val}^{\mathfrak{M}}(t_2)$, and hence $s(x) = \mathrm{Val}^{\mathfrak{M}}(t_2)$. By applying Proposition 16.23 again, we also have $\mathfrak{M}, s \vDash \varphi(t_2)$. By Proposition 16.18, $\mathfrak{M} \vDash \varphi(t_2)$. The case of $=\mathbb{F}$ is treated similarly. □

Chapter 22

Axiomatic Derivations

No effort has been made yet to ensure that the material in this chapter respects various tags indicating which connectives and quantifiers are primitive or defined: all are assumed to be primitive, except $\leftrightarrow$ which is assumed to be defined. If the FOL tag is true, we produce a version with quantifiers, otherwise without.

content/first-order-logic/axiomatic-deduction/rules-and-proofs.tex

22.1 Rules and Derivations

explanation

Axiomatic derivations are perhaps the simplest derivation system for logic. A derivation is just a sequence of formulas. To count as a derivation, every formula in the sequence must either be an instance of an axiom, or must follow from one or more formulas that precede it in the sequence by a rule of inference. A derivation derives its last formula.

fol:axd:rul:sec

Definition 22.1 (Derivability). If Γ is a set of formulas of $\mathcal{L}$ then a *derivation* from Γ is a finite sequence $\varphi_1, \ldots, \varphi_n$ of formulas where for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. φ_i follows from some φ_j (and φ_k) with $j < i$ (and $k < i$) by a rule of inference.

What counts as a correct derivation depends on which inference rules we allow (and of course what we take to be axioms). And an inference rule is an if-then statement that tells us that, under certain conditions, a step A_i in a derivation is a correct inference step.

Definition 22.2 (Rule of inference). A *rule of inference* gives a sufficient condition for what counts as a correct inference step in a derivation from Γ.

For instance, since any one-element sequence φ with $\varphi \in \Gamma$ trivially counts as a derivation, the following might be a very simple rule of inference:

> If $\varphi \in \Gamma$, then φ is always a correct inference step in any derivation from Γ.

Similarly, if φ is one of the axioms, then φ by itself is a derivation, and so this is also a rule of inference:

> If φ is an axiom, then φ is a correct inference step.

It gets more interesting if the rule of inference appeals to formulas that appear before the step considered. The following rule is called *modus ponens:*

> If $\psi \to \varphi$ and ψ occur higher up in the derivation, then φ is a correct inference step.

If this is the only rule of inference, then our definition of derivation above amounts to this: $\varphi_1, \ldots, \varphi_n$ is a derivation iff for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. for some $j < i$, φ_j is $\psi \to \varphi_i$, and for some $k < i$, φ_k is ψ.

The last clause says that φ_i follows from φ_j ($\psi \to \varphi_i$) and φ_k (ψ) by modus ponens. If we can go from 1 to n, and each time we find a formula φ_i that is either in Γ, an axiom, or which a rule of inference tells us that it is a correct inference step, then the entire sequence counts as a correct derivation.

Definition 22.3 (Derivability). A formula φ is *derivable* from Γ, written $\Gamma \vdash \varphi$, if there is a derivation from Γ ending in φ.

Definition 22.4 (Theorems). A formula φ is a *theorem* if there is a derivation of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

---

content/first-order-logic/axiomatic-deduction/axioms-rules-propositional.tex

22.2 Axioms and Rules for the Propositional Connectives

fol:axd:prp:sec

Definition 22.5 (Axioms). The set of Ax_0 of *axioms* for the propositional connectives comprises all formulas of the following forms:

$$(\varphi \land \psi) \to \varphi \tag{22.1}$$ fol:axd:prp:ax:land1

$$(\varphi \land \psi) \to \psi \tag{22.2}$$ fol:axd:prp:ax:land2

$$\varphi \to (\psi \to (\varphi \land \psi)) \tag{22.3}$$ fol:axd:prp:ax:land3

$$\varphi \to (\varphi \lor \psi) \tag{22.4}$$ fol:axd:prp:ax:lor1

$$\varphi \to (\psi \lor \varphi) \tag{22.5}$$ fol:axd:prp:ax:lor2

$$(\varphi \to \chi) \to ((\psi \to \chi) \to ((\varphi \lor \psi) \to \chi)) \tag{22.6}$$ fol:axd:prp:ax:lor3

$$\varphi \to (\psi \to \varphi) \tag{22.7}$$ fol:axd:prp:ax:lif1

$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \tag{22.8}$$ fol:axd:prp:ax:lif2

$$(\varphi \to \psi) \to ((\varphi \to \neg\psi) \to \neg\varphi) \tag{22.9}$$ fol:axd:prp:ax:lnot1

$$\neg\varphi \to (\varphi \to \psi) \tag{22.10}$$ fol:axd:prp:ax:lnot2

$$\top \tag{22.11}$$ fol:axd:prp:ax:ltrue

$$\bot \to \varphi \tag{22.12}$$ fol:axd:prp:ax:lfalse1

$$(\varphi \to \bot) \to \neg\varphi \tag{22.13}$$ fol:axd:prp:ax:lfalse2

$$\neg\neg\varphi \to \varphi \tag{22.14}$$ fol:axd:prp:ax:dne

Definition 22.6 (Modus ponens). If ψ and $\psi \to \varphi$ already occur in a derivation, then φ is a correct inference step.

We'll abbreviate the rule modus ponens as "MP."

---

content/first-order-logic/axiomatic-deduction/axioms-rules-quantifiers.tex

22.3 Axioms and Rules for Quantifiers

fol:axd:qua:sec

Definition 22.7 (Axioms for quantifiers). The *axioms* governing quantifiers are all instances of the following:

$$\forall x\, \psi \to \psi(t), \tag{22.15}$$ fol:axd:qua:ax:q1

$$\psi(t) \to \exists x\, \psi. \tag{22.16}$$ fol:axd:qua:ax:q2

for any closed term t.

Definition 22.8 (Rules for quantifiers).

If $\psi \to \varphi(a)$ already occurs in the derivation and a does not occur in Γ or ψ, then $\psi \to \forall x\, \varphi(x)$ is a correct inference step.

If $\varphi(a) \to \psi$ already occurs in the derivation and a does not occur in Γ or ψ, then $\exists x\, \varphi(x) \to \psi$ is a correct inference step.

We'll abbreviate either of these by "QR."

---

content/first-order-logic/axiomatic-deduction/proving-things.tex

22.4 Examples of Derivations

fol:axd:pro:sec

Example 22.9. Suppose we want to prove $(\neg\theta \vee \alpha) \to (\theta \to \alpha)$. Clearly, this is not an instance of any of our axioms, so we have to use the MP rule to derive it. Our only rule is MP, which given φ and $\varphi \to \psi$ allows us to justify ψ. One strategy would be to use eq. (22.6) with φ being $\neg\theta$, ψ being α, and χ being $\theta \to \alpha$, i.e., the instance

$$(\neg\theta \to (\theta \to \alpha)) \to ((\alpha \to (\theta \to \alpha)) \to ((\neg\theta \vee \alpha) \to (\theta \to \alpha))).$$

Why? Two applications of MP yield the last part, which is what we want. And we easily see that $\neg\theta \to (\theta \to \alpha)$ is an instance of eq. (22.10), and $\alpha \to (\theta \to \alpha)$ is an instance of eq. (22.7). So our derivation is:

1.	$\neg\theta \to (\theta \to \alpha)$	eq. (22.10)
2.	$(\neg\theta \to (\theta \to \alpha)) \to ((\alpha \to (\theta \to \alpha)) \to ((\neg\theta \vee \alpha) \to (\theta \to \alpha)))$	eq. (22.6)
3.	$(\alpha \to (\theta \to \alpha)) \to ((\neg\theta \vee \alpha) \to (\theta \to \alpha))$	1, 2, MP
4.	$\alpha \to (\theta \to \alpha)$	eq. (22.7)
5.	$(\neg\theta \vee \alpha) \to (\theta \to \alpha)$	3, 4, MP

fol:axd:pro:ex:identity

Example 22.10. Let's try to find a derivation of $\theta \to \theta$. It is not an instance of an axiom, so we have to use MP to derive it. eq. (22.7) is an axiom of the form $\varphi \to \psi$ to which we could apply MP. To be useful, of course, the ψ which MP would justify as a correct step in this case would have to be $\theta \to \theta$, since this is what we want to derive. That means φ would also have to be θ, i.e., we might look at this instance of eq. (22.7):

$$\theta \to (\theta \to \theta)$$

In order to apply MP, we would also need to justify the corresponding second premise, namely φ. But in our case, that would be θ, and we won't be able to derive θ by itself. So we need a different strategy.

The other axiom involving just $\to$ is eq. (22.8), i.e.,

$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$$

We could get to the last nested conditional by applying MP twice. Again, that would mean that we want an instance of eq. (22.8) where $\varphi \to \chi$ is $\theta \to \theta$, the

formula we are aiming for. Then of course, φ and χ are both θ. How should we pick ψ so that both $\varphi \to (\psi \to \chi)$ and $\varphi \to \psi$, i.e., in our case $\theta \to (\psi \to \theta)$ and $\theta \to \psi$, are also derivable? Well, the first of these is already an instance of eq. (22.7), whatever we decide ψ to be. And $\theta \to \psi$ would be another instance of eq. (22.7) if ψ were $(\theta \to \theta)$. So, our derivation is:

1.	$\theta \to ((\theta \to \theta) \to \theta)$	eq. (22.7)
2.	$(\theta \to ((\theta \to \theta) \to \theta)) \to ((\theta \to (\theta \to \theta)) \to (\theta \to \theta))$	eq. (22.8)
3.	$(\theta \to (\theta \to \theta)) \to (\theta \to \theta)$	1, 2, MP
4.	$\theta \to (\theta \to \theta)$	eq. (22.7)
5.	$\theta \to \theta$	3, 4, MP

Example 22.11. Sometimes we want to show that there is a derivation of some formula from some other formulas Γ. For instance, let's show that we can derive $\varphi \to \chi$ from $\Gamma = \{\varphi \to \psi, \psi \to \chi\}$. fol:axd:pro:ex:chain

1.	$\varphi \to \psi$	Hyp
2.	$\psi \to \chi$	Hyp
3.	$(\psi \to \chi) \to (\varphi \to (\psi \to \chi))$	eq. (22.7)
4.	$\varphi \to (\psi \to \chi)$	2, 3, MP
5.	$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$	eq. (22.8)
6.	$((\varphi \to \psi) \to (\varphi \to \chi))$	4, 5, MP
7.	$\varphi \to \chi$	1, 6, MP

The lines labelled "Hyp" (for "hypothesis") indicate that the formula on that line is an element of Γ.

Proposition 22.12. *If $\Gamma \vdash \varphi \to \psi$ and $\Gamma \vdash \psi \to \chi$, then $\Gamma \vdash \varphi \to \chi$* fol:axd:pro:prop:chain

Proof. Suppose $\Gamma \vdash \varphi \to \psi$ and $\Gamma \vdash \psi \to \chi$. Then there is a derivation of $\varphi \to \psi$ from Γ; and a derivation of $\psi \to \chi$ from Γ as well. Combine these into a single derivation by concatenating them. Now add lines 3–7 of the derivation in the preceding example. This is a derivation of $\varphi \to \chi$—which is the last line of the new derivation—from Γ. Note that the justifications of lines 4 and 7 remain valid if the reference to line number 1 is replaced by reference to the last line of the derivation of $\varphi \to \psi$, and reference to line number 2 by reference to the last line of the derivation of $\psi \to \chi$. □

Problem 22.1. Show that the following hold by exhibiting derivations from the axioms:

1. $(\varphi \wedge \psi) \to (\psi \wedge \varphi)$
2. $((\varphi \wedge \psi) \to \chi) \to (\varphi \to (\psi \to \chi))$
3. $\neg(\varphi \vee \psi) \to \neg\varphi$

content/first-order-logic/axiomatic-deduction/proving-things-quant.tex

22.5 Derivations with Quantifiers

fol:axd:prq:sec

Example 22.13. Let us give a derivation of $(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \forall x\, (\varphi(x) \land \psi(x))$.

First, note that

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \forall x\, \varphi(x)$$

is an instance of eq. (22.1), and

$$\forall x\, \varphi(x) \to \varphi(a)$$

of eq. (22.15). So, by Proposition 22.12, we know that

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \varphi(a)$$

is derivable. Likewise, since

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \forall y\, \psi(y) \qquad \text{and}$$
$$\forall y\, \psi(y) \to \psi(a)$$

are instances of eq. (22.2) and eq. (22.15), respectively,

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \psi(a)$$

is derivable by Proposition 22.12. Using an appropriate instance of eq. (22.3) and two applications of MP, we see that

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to (\varphi(a) \land \psi(a))$$

is derivable. We can now apply QR to obtain

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \forall x\, (\varphi(x) \land \psi(x)).$$

---

content/first-order-logic/axiomatic-deduction/proof-theoretic-notions.tex

22.6 Proof-Theoretic Notions

fol:axd:ptn:sec

explanation

Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain formulas. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

Definition 22.14 (Derivability). A formula φ is *derivable* from Γ, written $\Gamma \vdash \varphi$, if there is a derivation from Γ ending in φ.

Definition 22.15 (Theorems). A formula φ is a *theorem* if there is a derivation of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 22.16 (Consistency). A set Γ of formulas is *consistent* if and only if $\Gamma \nvdash \bot$; it is *inconsistent* otherwise.

Proposition 22.17 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.*

fol:axd:ptn: prop:reflexivity

Proof. The formula φ by itself is a derivation of φ from Γ. □

Proposition 22.18 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.*

fol:axd:ptn: prop:monotonicity

Proof. Any derivation of φ from Γ is also a derivation of φ from Δ. □

Proposition 22.19 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.*

fol:axd:ptn: prop:transitivity

Proof. Suppose $\{\varphi\} \cup \Delta \vdash \psi$. Then there is a derivation $\psi_1, \ldots, \psi_l = \psi$ from $\{\varphi\} \cup \Delta$. Some of the steps in that derivation will be correct because of a rule which refers to a prior line $\psi_i = \varphi$. By hypothesis, there is a derivation of φ from Γ, i.e., a derivation $\varphi_1, \ldots, \varphi_k = \varphi$ where every φ_i is an axiom, an element of Γ, or correct by a rule of inference. Now consider the sequence

$$\varphi_1, \ldots, \varphi_k = \varphi, \psi_1, \ldots, \psi_l = \psi.$$

This is a correct derivation of ψ from $\Gamma \cup \Delta$ since every $B_i = \varphi$ is now justified by the same rule which justifies $\varphi_k = \varphi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \ldots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

Proposition 22.20. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every φ.*

fol:axd:ptn: prop:incons

Proof. Exercise. □

Problem 22.2. Prove Proposition 22.20.

Proposition 22.21 (Compactness).

fol:axd:ptn: prop:proves-compact

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof. 1. If $\Gamma \vdash \varphi$, then there is a finite sequence of formulas $\varphi_1, \ldots, \varphi_n$ so that $\varphi \equiv \varphi_n$ and each φ_i is either a logical axiom, an element of Γ or follows from previous formulas by modus ponens. Take Γ_0 to be those φ_i which are in Γ. Then the derivation is likewise a derivation from Γ_0, and so $\Gamma_0 \vdash \varphi$.

2. This is the contrapositive of (1) for the special case $\varphi \equiv \bot$. □

---

content/first-order-logic/axiomatic-deduction/deduction-theorem.tex

22.7 The Deduction Theorem

fol:axd:ded:sec

As we've seen, giving derivations in an axiomatic system is cumbersome, and derivations may be hard to find. Rather than actually write out long lists of formulas, it is generally easier to argue that such derivations exist, by making use of a few simple results. We've already established three such results: Proposition 22.17 says we can always assert that $\Gamma \vdash \varphi$ when we know that $\varphi \in \Gamma$. Proposition 22.18 says that if $\Gamma \vdash \varphi$ then also $\Gamma \cup \{\psi\} \vdash \varphi$. And Proposition 22.19 implies that if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. Here's another simple result, a "meta"-version of modus ponens:

fol:axd:ded:prop:mp

Proposition 22.22. *If $\Gamma \vdash \varphi$ and $\Gamma \vdash \varphi \to \psi$, then $\Gamma \vdash \psi$.*

Proof. We have that $\{\varphi, \varphi \to \psi\} \vdash \psi$:

1.	φ	Hyp.
2.	$\varphi \to \psi$	Hyp.
3.	ψ	1, 2, MP

By Proposition 22.19, $\Gamma \vdash \psi$. □

The most important result we'll use in this context is the deduction theorem:

fol:axd:ded:thm:deduction-thm

Theorem 22.23 (Deduction Theorem). *$\Gamma \cup \{\varphi\} \vdash \psi$ if and only if $\Gamma \vdash \varphi \to \psi$.*

Proof. The "if" direction is immediate. If $\Gamma \vdash \varphi \to \psi$ then also $\Gamma \cup \{\varphi\} \vdash \varphi \to \psi$ by Proposition 22.18. Also, $\Gamma \cup \{\varphi\} \vdash \varphi$ by Proposition 22.17. So, by Proposition 22.22, $\Gamma \cup \{\varphi\} \vdash \psi$.

For the "only if" direction, we proceed by induction on the length of the derivation of ψ from $\Gamma \cup \{\varphi\}$.

For the induction basis, we prove the claim for every derivation of length 1. A derivation of ψ from $\Gamma \cup \{\varphi\}$ of length 1 consists of ψ by itself; and if it is correct ψ is either $\in \Gamma \cup \{\varphi\}$ or is an axiom. If $\psi \in \Gamma$ or is an axiom, then $\Gamma \vdash \psi$. We also have that $\Gamma \vdash \psi \to (\varphi \to \psi)$ by eq. (22.7), and Proposition 22.22 gives $\Gamma \vdash \varphi \to \psi$. If $\psi \in \{\varphi\}$ then $\Gamma \vdash \varphi \to \psi$ because the last sentence $\varphi \to \psi$ is the same as $\varphi \to \varphi$, and we have derived that in Example 22.10.

For the inductive step, suppose a derivation of ψ from $\Gamma \cup \{\varphi\}$ ends with a step ψ which is justified by modus ponens. (If it is not justified by modus ponens, $\psi \in \Gamma$, $\psi \equiv \varphi$, or ψ is an axiom, and the same reasoning as in the induction basis applies.) Then some previous steps in the derivation are $\chi \to \psi$ and χ, for some formula χ, i.e., $\Gamma \cup \{\varphi\} \vdash \chi \to \psi$ and $\Gamma \cup \{\varphi\} \vdash \chi$, and the respective derivations are shorter, so the inductive hypothesis applies to them. We thus have both:

$$\Gamma \vdash \varphi \to (\chi \to \psi);$$
$$\Gamma \vdash \varphi \to \chi.$$

But also

$$\Gamma \vdash (\varphi \to (\chi \to \psi)) \to ((\varphi \to \chi) \to (\varphi \to \psi)),$$

by eq. (22.8), and two applications of Proposition 22.22 give $\Gamma \vdash \varphi \to \psi$, as required. □

Notice how eq. (22.7) and eq. (22.8) were chosen precisely so that the Deduction Theorem would hold.

The following are some useful facts about derivability, which we leave as exercises.

Proposition 22.24. *fol:axd:ded: prop:derivfacts*

1. $\vdash (\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$; *fol:axd:ded: derivfacts:a*
2. *If* $\Gamma \cup \{\neg\varphi\} \vdash \neg\psi$ *then* $\Gamma \cup \{\psi\} \vdash \varphi$ *(Contraposition);* *fol:axd:ded: derivfacts:b*
3. $\{\varphi, \neg\varphi\} \vdash \psi$ *(Ex Falso Quodlibet, Explosion);* *fol:axd:ded: derivfacts:c*
4. $\{\neg\neg\varphi\} \vdash \varphi$ *(Double Negation Elimination);* *fol:axd:ded: derivfacts:d*
5. *If* $\Gamma \vdash \neg\neg\varphi$ *then* $\Gamma \vdash \varphi$; *fol:axd:ded: derivfacts:e*

Problem 22.3. Prove Proposition 22.24

---

content/first-order-logic/axiomatic-deduction/deduction-theorem-quantifiers.tex

22.8 The Deduction Theorem with Quantifiers

Theorem 22.25 (Deduction Theorem). *If* $\Gamma \cup \{\varphi\} \vdash \psi$, *then* $\Gamma \vdash \varphi \to \psi$. *fol:axd:ddq: sec fol:axd:ddq: thm:deduction-thm-q*

Proof. We again proceed by induction on the length of the derivation of ψ from $\Gamma \cup \{\varphi\}$.

The proof of the induction basis is identical to that in the proof of Theorem 22.23.

For the inductive step, suppose again that the derivation of ψ from $\Gamma \cup \{\varphi\}$ ends with a step ψ which is justified by an inference rule. If the inference rule is modus ponens, we proceed as in the proof of Theorem 22.23. If the inference rule is QR, we know that $\psi \equiv \chi \to \forall x\, \theta(x)$ and a formula of the form $\chi \to \theta(a)$ appears earlier in the derivation, where a does not occur in χ, φ, or Γ. We thus have that

$$\Gamma \cup \{\varphi\} \vdash \chi \to \theta(a),$$

and the induction hypothesis applies, i.e., we have that

$$\Gamma \vdash \varphi \to (\chi \to \theta(a)).$$

By

$$\vdash (\varphi \to (\chi \to \theta(a))) \to ((\varphi \wedge \chi) \to \theta(a))$$

and modus ponens we get

$$\Gamma \vdash (\varphi \wedge \chi) \to \theta(a).$$

Since the eigenvariable condition still applies, we can add a step to this derivation justified by QR, and get

$$\Gamma \vdash (\varphi \wedge \chi) \to \forall x\, \theta(x).$$

We also have

$$\vdash ((\varphi \wedge \chi) \to \forall x\, \theta(x)) \to (\varphi \to (\chi \to \forall x\, \theta(x)),$$

so by modus ponens,

$$\Gamma \vdash \varphi \to (\chi \to \forall x\, \theta(x)),$$

i.e., $\Gamma \vdash \psi$.

We leave the case where ψ is justified by the rule QR, but is of the form $\exists x\, \theta(x) \to \chi$, as an exercise. □

Problem 22.4. Complete the proof of Theorem 22.25.

---

content/first-order-logic/axiomatic-deduction/provability-consistency.tex

22.9 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem. fol:axd:prv:sec

Proposition 22.26. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.* fol:axd:prv:prop:provability-contr

Proof. If $\Gamma \cup \{\varphi\}$ is inconsistent, then $\Gamma \cup \{\varphi\} \vdash \bot$. By Proposition 22.17, $\Gamma \vdash \psi$ for every $\psi \in \Gamma$. Since also $\Gamma \vdash \varphi$ by hypothesis, $\Gamma \vdash \psi$ for every $\psi \in \Gamma \cup \{\varphi\}$. By Proposition 22.19, $\Gamma \vdash \bot$, i.e., Γ is inconsistent. □

Proposition 22.27. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.* fol:axd:prv:prop:prov-incons

Proof. First suppose $\Gamma \vdash \varphi$. Then $\Gamma \cup \{\neg\varphi\} \vdash \varphi$ by Proposition 22.18. $\Gamma \cup \{\neg\varphi\} \vdash \neg\varphi$ by Proposition 22.17. We also have $\vdash \neg\varphi \to (\varphi \to \bot)$ by eq. (22.10). So by two applications of Proposition 22.22, we have $\Gamma \cup \{\neg\varphi\} \vdash \bot$.

Now assume $\Gamma \cup \{\neg\varphi\}$ is inconsistent, i.e., $\Gamma \cup \{\neg\varphi\} \vdash \bot$. By the deduction theorem, $\Gamma \vdash \neg\varphi \to \bot$. $\Gamma \vdash (\neg\varphi \to \bot) \to \neg\neg\varphi$ by eq. (22.13), so $\Gamma \vdash \neg\neg\varphi$ by Proposition 22.22. Since $\Gamma \vdash \neg\neg\varphi \to \varphi$ (eq. (22.14)), we have $\Gamma \vdash \varphi$ by Proposition 22.22 again. □

Problem 22.5. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

Proposition 22.28. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.* fol:axd:prv:prop:explicit-inc

Proof. $\Gamma \vdash \neg\varphi \to (\varphi \to \bot)$ by eq. (22.10). $\Gamma \vdash \bot$ by two applications of Proposition 22.22. □

Proposition 22.29. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.* fol:axd:prv:prop:provability-exhaustive

Proof. Exercise. □

Problem 22.6. Prove Proposition 22.29

---

content/first-order-logic/axiomatic-deduction/provability-propositional.tex

22.10 Derivability and the Propositional Connectives

explanation We establish that the derivability relation $\vdash$ of axiomatic deduction is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \land \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. fol:axd:ppr:sec

Proposition 22.30. fol:axd:ppr:prop:provability-land

fol:axd:ppr: prop:provability-land-left
fol:axd:ppr: prop:provability-land-right

1. *Both* $\varphi \land \psi \vdash \varphi$ *and* $\varphi \land \psi \vdash \psi$
2. $\varphi, \psi \vdash \varphi \land \psi$.

Proof. 1. From eq. (22.1) and eq. (22.1) by modus ponens.

2. From eq. (22.3) by two applications of modus ponens. □

fol:axd:ppr: prop:provability-lor

Proposition 22.31.

1. $\varphi \lor \psi, \neg\varphi, \neg\psi$ *is inconsistent.*
2. *Both* $\varphi \vdash \varphi \lor \psi$ *and* $\psi \vdash \varphi \lor \psi$.

Proof. 1. From eq. (22.9) we get $\vdash \neg\varphi \to (\varphi \to \bot)$ and $\vdash \neg\psi \to (\psi \to \bot)$. So by the deduction theorem, we have $\{\neg\varphi\} \vdash \varphi \to \bot$ and $\{\neg\psi\} \vdash \psi \to \bot$. From eq. (22.6) we get $\{\neg\varphi, \neg\psi\} \vdash (\varphi \lor \psi) \to \bot$. By the deduction theorem, $\{\varphi \lor \psi, \neg\varphi, \neg\psi\} \vdash \bot$.

2. From eq. (22.4) and eq. (22.5) by modus ponsens. □

fol:axd:ppr: prop:provability-lif
fol:axd:ppr: prop:provability-lif-left
fol:axd:ppr: prop:provability-lif-right

Proposition 22.32.

1. $\varphi, \varphi \to \psi \vdash \psi$.
2. *Both* $\neg\varphi \vdash \varphi \to \psi$ *and* $\psi \vdash \varphi \to \psi$.

Proof. 1. We can derive:

1.	φ	Hyp
2.	$\varphi \to \psi$	Hyp
3.	ψ	1, 2, MP

2. By eq. (22.10) and eq. (22.7) and the deduction theorem, respectively. □

---

content/first-order-logic/axiomatic-deduction/provability-quantifiers.tex

22.11 Derivability and the Quantifiers

fol:axd:qpr: sec

The completeness theorem also requires that axiomatic deductions yield the facts about $\vdash$ established in this section. explanation

fol:axd:qpr: thm:strong-generalization

Theorem 22.33. *If c is a constant symbol not occurring in Γ or $\varphi(x)$ and $\Gamma \vdash \varphi(c)$, then $\Gamma \vdash \forall x\, \varphi(x)$.*

Proof. By the deduction theorem, $\Gamma \vdash \top \to \varphi(c)$. Since c does not occur in Γ or $\top$, we get $\Gamma \vdash \top \to \forall x\, \varphi(x)$. By the deduction theorem again, $\Gamma \vdash \forall x\, \varphi(x)$. □

Proposition 22.34. *fol:axd:qpr: prop:provability-quantifiers*

1. $\varphi(t) \vdash \exists x\, \varphi(x)$.
2. $\forall x\, \varphi(x) \vdash \varphi(t)$.

Proof. 1. By eq. (22.16) and the deduction theorem.

2. By eq. (22.15) and the deduction theorem. □

---

content/first-order-logic/axiomatic-deduction/soundness.tex

22.12 Soundness

explanation A derivation system, such as axiomatic deduction, is *sound* if it cannot derive things that do not actually hold. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that *fol:axd:sou: sec*

1. every derivable φ is valid;
2. if φ is derivable from some others Γ, it is also a consequence of them;
3. if a set of formulas Γ is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Proposition 22.35. *If φ is an axiom, then $\mathfrak{M}, s \vDash \varphi$ for each structure $\mathfrak{M}$ and assignment s.*

Proof. We have to verify that all the axioms are valid. For instance, here is the case for eq. (22.15): suppose t is free for x in φ, and assume $\mathfrak{M}, s \vDash \forall x\, \varphi$. Then by definition of satisfaction, for each $s' \sim_x s$, also $\mathfrak{M}, s' \vDash \varphi$, and in particular this holds when $s'(x) = \mathrm{Val}^{\mathfrak{M}}_s(t)$. By Proposition 16.23, $\mathfrak{M}, s \vDash \varphi[t/x]$. This shows that $\mathfrak{M}, s \vDash (\forall x\, \varphi \to \varphi[t/x])$. □

Theorem 22.36 (Soundness). *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.* *fol:axd:sou: thm:soundness*

Proof. By induction on the length of the derivation of φ from Γ. If there are no steps justified by inferences, then all formulas in the derivation are either instances of axioms or are in Γ. By the previous proposition, all the axioms are valid, and hence if φ is an axiom then $\Gamma \vDash \varphi$. If $\varphi \in \Gamma$, then trivially $\Gamma \vDash \varphi$.

If the last step of the derivation of φ is justified by modus ponens, then there are formulas ψ and $\psi \to \varphi$ in the derivation, and the induction hypothesis applies to the part of the derivation ending in those formulas (since they contain

at least one fewer step justified by an inference). So, by induction hypothesis, $\Gamma \vDash \psi$ and $\Gamma \vDash \psi \to \varphi$. Then $\Gamma \vDash \varphi$ by Theorem 16.30.

Now suppose the last step is justified by QR. Then that step has the form $\chi \to \forall x\, B(x)$ and there is a preceding step $\chi \to \psi(c)$ with c not in Γ, χ, or $\forall x\, B(x)$. By induction hypothesis, $\Gamma \vDash \chi \to \psi(c)$. By Theorem 16.30, $\Gamma \cup \{\chi\} \vDash \psi(c)$.

Consider some structure $\mathfrak{M}$ such that $\mathfrak{M} \vDash \Gamma \cup \{\chi\}$. We need to show that $\mathfrak{M} \vDash \forall x\, \psi(x)$. Since $\forall x\, \psi(x)$ is a sentence, this means we have to show that for every variable assignment s, $\mathfrak{M}, s \vDash \psi(x)$ (Proposition 16.19). Since $\Gamma \cup \{\chi\}$ consists entirely of sentences, $\mathfrak{M}, s \vDash \theta$ for all $\theta \in \Gamma$ by Definition 16.11. Let $\mathfrak{M}'$ be like $\mathfrak{M}$ except that $c^{\mathfrak{M}'} = s(x)$. Since c does not occur in Γ or χ, $\mathfrak{M}' \vDash \Gamma \cup \{\chi\}$ by Corollary 16.21. Since $\Gamma \cup \{\chi\} \vDash \psi(c)$, $\mathfrak{M}' \vDash B(c)$. Since $\psi(c)$ is a sentence, $\mathfrak{M}', s \vDash \psi(c)$ by Proposition 16.18. $\mathfrak{M}', s \vDash \psi(x)$ iff $\mathfrak{M}', s \vDash \psi(c)$ by Proposition 16.23 (recall that $\psi(c)$ is just $\psi(x)[c/x]$). So, $\mathfrak{M}', s \vDash \psi(x)$. Since c does not occur in $\psi(x)$, by Proposition 16.20, $\mathfrak{M}, s \vDash \psi(x)$. But s was an arbitrary variable assignment, so $\mathfrak{M} \vDash \forall x\, \psi(x)$. Thus $\Gamma \cup \{\chi\} \vDash \forall x\, \psi(x)$. By Theorem 16.30, $\Gamma \vDash \chi \to \forall x\, \psi(x)$.

The case where φ is justified by QR but is of the form $\exists x\, \psi(x) \to \chi$ is left as an exercise. □

Problem 22.7. Complete the proof of Theorem 22.36.

fol:axd:sou: cor:weak-soundness **Corollary 22.37.** *If $\vdash \varphi$, then φ is valid.*

fol:axd:sou: cor:consistency-soundness **Corollary 22.38.** *If Γ is satisfiable, then it is consistent.*

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then $\Gamma \vdash \bot$, i.e., there is a derivation of $\bot$ from Γ. By Theorem 22.36, any structure $\mathfrak{M}$ that satisfies Γ must satisfy $\bot$. Since $\mathfrak{M} \nvDash \bot$ for every structure $\mathfrak{M}$, no $\mathfrak{M}$ can satisfy Γ, i.e., Γ is not satisfiable. □

---

content/first-order-logic/axiomatic-deduction/identity.tex

22.13 Derivations with Identity predicate

fol:axd:ide: sec In order to accommodate $=$ in derivations, we simply add new axiom schemas. The definition of derivation and $\vdash$ remains the same, we just also allow the new axioms.

Definition 22.39 (Axioms for identity predicate).

fol:axd:ide: ax:id1
$$t = t, \tag{22.17}$$

fol:axd:ide: ax:id2
$$t_1 = t_2 \to (\psi(t_1) \to \psi(t_2)), \tag{22.18}$$

for any closed terms t, t_1, t_2.

Proposition 22.40. *The axioms eq. (22.17) and eq. (22.18) are valid.* fol:axd:ide:prop:sound

Proof. Exercise. □

Problem 22.8. Prove Proposition 22.40.

Proposition 22.41. *$\Gamma \vdash t = t$, for any term t and set Γ.* fol:axd:ide:prop:iden1

Proposition 22.42. *If $\Gamma \vdash \varphi(t_1)$ and $\Gamma \vdash t_1 = t_2$, then $\Gamma \vdash \varphi(t_2)$.* fol:axd:ide:prop:iden2

Proof. The formula

$$(t_1 = t_2 \to (\varphi(t_1) \to \varphi(t_2)))$$

is an instance of eq. (22.18). The conclusion follows by two applications of MP. □

Chapter 23

The Completeness Theorem

---

content/first-order-logic/completeness/introduction.tex

23.1 Introduction

fol:com:int:sec

The completeness theorem is one of the most fundamental results about logic. It comes in two formulations, the equivalence of which we'll prove. In its first formulation it says something fundamental about the relationship between semantic consequence and our derivation system: if a sentence φ follows from some sentences Γ, then there is also a derivation that establishes $\Gamma \vdash \varphi$. Thus, the derivation system is as strong as it can possibly be without proving things that don't actually follow.

In its second formulation, it can be stated as a model existence result: every consistent set of sentences is satisfiable. Consistency is a proof-theoretic notion: it says that our derivation system is unable to produce certain derivations. But who's to say that just because there are no derivations of a certain sort from Γ, it's guaranteed that there is a structure $\mathfrak{M}$? Before the completeness theorem was first proved—in fact before we had the derivation systems we now do—the great German mathematician David Hilbert held the view that consistency of

mathematical theories guarantees the existence of the objects they are about. He put it as follows in a letter to Gottlob Frege:

> If the arbitrarily given axioms do not contradict one another with all their consequences, then they are true and the things defined by the axioms exist. This is for me the criterion of truth and existence.

Frege vehemently disagreed. The second formulation of the completeness theorem shows that Hilbert was right in at least the sense that if the axioms are consistent, then *some* structure exists that makes them all true.

These aren't the only reasons the completeness theorem—or rather, its proof—is important. It has a number of important consequences, some of which we'll discuss separately. For instance, since any derivation that shows $\Gamma \vdash \varphi$ is finite and so can only use finitely many of the sentences in Γ, it follows by the completeness theorem that if φ is a consequence of Γ, it is already a consequence of a finite subset of Γ. This is called *compactness*. Equivalently, if every finite subset of Γ is consistent, then Γ itself must be consistent.

Although the compactness theorem follows from the completeness theorem via the detour through derivations, it is also possible to use the *the proof of* the completeness theorem to establish it directly. For what the proof does is take a set of sentences with a certain property—consistency—and constructs a structure out of this set that has certain properties (in this case, that it satisfies the set). Almost the very same construction can be used to directly establish compactness, by starting from "finitely satisfiable" sets of sentences instead of consistent ones. The construction also yields other consequences, e.g., that any satisfiable set of sentences has a finite or denumerable model. (This result is called the Löwenheim–Skolem theorem.) In general, the construction of structures from sets of sentences is used often in logic, and sometimes even in philosophy.

---

content/first-order-logic/completeness/outline.tex

23.2 Outline of the Proof

fol:com:out:sec

The proof of the completeness theorem is a bit complex, and upon first reading it, it is easy to get lost. So let us outline the proof. The first step is a shift of perspective, that allows us to see a route to a proof. When completeness is thought of as "whenever $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$," it may be hard to even come up with an idea: for to show that $\Gamma \vdash \varphi$ we have to find a derivation, and it does not look like the hypothesis that $\Gamma \vDash \varphi$ helps us for this in any way. For some proof systems it is possible to directly construct a derivation, but we will take a slightly different approach. The shift in perspective required is this: completeness can also be formulated as: "if Γ is consistent, it is satisfiable." Perhaps we can use the information in Γ together with the hypothesis that it is consistent to construct a structure that satisfies every sentence in Γ. After all,

we know what kind of structure we are looking for: one that is as Γ describes it!

If Γ contains only atomic sentences, it is easy to construct a model for it. Suppose the atomic sentences are all of the form $P(a_1, \ldots, a_n)$ where the a_i are constant symbols. All we have to do is come up with a domain $|\mathfrak{M}|$ and an assignment for P so that $\mathfrak{M} \vDash P(a_1, \ldots, a_n)$. But that's not very hard: put $|\mathfrak{M}| = \mathbb{N}$, $c_i^{\mathfrak{M}} = i$, and for every $P(a_1, \ldots, a_n) \in \Gamma$, put the tuple $\langle k_1, \ldots, k_n \rangle$ into $P^{\mathfrak{M}}$, where k_i is the index of the constant symbol a_i (i.e., $a_i \equiv c_{k_i}$).

Now suppose Γ contains some formula $\neg\psi$, with ψ atomic. We might worry that the construction of $\mathfrak{M}$ interferes with the possibility of making $\neg\psi$ true. But here's where the consistency of Γ comes in: if $\neg\psi \in \Gamma$, then $\psi \notin \Gamma$, or else Γ would be inconsistent. And if $\psi \notin \Gamma$, then according to our construction of $\mathfrak{M}$, $\mathfrak{M} \nvDash \psi$, so $\mathfrak{M} \vDash \neg\psi$. So far so good.

What if Γ contains complex, non-atomic formulas? Say it contains $\varphi \wedge \psi$. To make that true, we should proceed as if both φ and ψ were in Γ. And if $\varphi \vee \psi \in \Gamma$, then we will have to make at least one of them true, i.e., proceed as if one of them was in Γ.

This suggests the following idea: we add additional formulas to Γ so as to (a) keep the resulting set consistent and (b) make sure that for every possible atomic sentence φ, either φ is in the resulting set, or $\neg\varphi$ is, and (c) such that, whenever $\varphi \wedge \psi$ is in the set, so are both φ and ψ, if $\varphi \vee \psi$ is in the set, at least one of φ or ψ is also, etc. We keep doing this (potentially forever). Call the set of all formulas so added Γ^*. Then our construction above would provide us with a structure $\mathfrak{M}$ for which we could prove, by induction, that it satisfies all sentences in Γ^*, and hence also all sentence in Γ since $\Gamma \subseteq \Gamma^*$. It turns out that guaranteeing (a) and (b) is enough. A set of sentences for which (b) holds is called *complete*. So our task will be to extend the consistent set Γ to a consistent and complete set Γ^*.

There is one wrinkle in this plan: if $\exists x\, \varphi(x) \in \Gamma$ we would hope to be able to pick some constant symbol c and add $\varphi(c)$ in this process. But how do we know we can always do that? Perhaps we only have a few constant symbols in our language, and for each one of them we have $\neg\varphi(c) \in \Gamma$. We can't also add $\varphi(c)$, since this would make the set inconsistent, and we wouldn't know whether $\mathfrak{M}$ has to make $\varphi(c)$ or $\neg\varphi(c)$ true. Moreover, it might happen that Γ contains only sentences in a language that has no constant symbols at all (e.g., the language of set theory).

The solution to this problem is to simply add infinitely many constants at the beginning, plus sentences that connect them with the quantifiers in the right way. (Of course, we have to verify that this cannot introduce an inconsistency.)

Our original construction works well if we only have constant symbols in the atomic sentences. But the language might also contain function symbols. In that case, it might be tricky to find the right functions on $\mathbb{N}$ to assign to these function symbols to make everything work. So here's another trick: instead of using i to interpret c_i, just take the set of constant symbols itself as the domain. Then $\mathfrak{M}$ can assign every constant symbol to itself: $c_i^{\mathfrak{M}} = c_i$. But why not go all the way: let $|\mathfrak{M}|$ be all *terms* of the language! If we do this, there is an

obvious assignment of functions (that take terms as arguments and have terms as values) to function symbols: we assign to the function symbol f_i^n the function which, given n terms $t_1, \ldots, t_n$ as input, produces the term $f_i^n(t_1, \ldots, t_n)$ as value.

The last piece of the puzzle is what to do with $=$. The predicate symbol $=$ has a fixed interpretation: $\mathfrak{M} \vDash t = t'$ iff $\mathrm{Val}^{\mathfrak{M}}(t) = \mathrm{Val}^{\mathfrak{M}}(t')$. Now if we set things up so that the value of a term t is t itself, then this structure will make *no* sentence of the form $t = t'$ true unless t and t' are one and the same term. And of course this is a problem, since basically every interesting theory in a language with function symbols will have as theorems sentences $t = t'$ where t and t' are not the same term (e.g., in theories of arithmetic: $(\mathsf{o} + \mathsf{o}) = \mathsf{o}$). To solve this problem, we change the domain of $\mathfrak{M}$: instead of using terms as the objects in $|\mathfrak{M}|$, we use sets of terms, and each set is so that it contains all those terms which the sentences in Γ require to be equal. So, e.g., if Γ is a theory of arithmetic, one of these sets will contain: o, $(\mathsf{o} + \mathsf{o})$, $(\mathsf{o} \times \mathsf{o})$, etc. This will be the set we assign to o, and it will turn out that this set is also the value of all the terms in it, e.g., also of $(\mathsf{o} + \mathsf{o})$. Therefore, the sentence $(\mathsf{o} + \mathsf{o}) = \mathsf{o}$ will be true in this revised structure.

So here's what we'll do. First we investigate the properties of complete consistent sets, in particular we prove that a complete consistent set contains $\varphi \wedge \psi$ iff it contains both φ and ψ, $\varphi \vee \psi$ iff it contains at least one of them, etc. (Proposition 23.2). Then we define and investigate "saturated" sets of sentences. A saturated set is one which contains conditionals that link each quantified sentence to instances of it (Definition 23.5). We show that any consistent set Γ can always be extended to a saturated set Γ' (Lemma 23.6). If a set is consistent, saturated, and complete it also has the property that it contains $\exists x\, \varphi(x)$ iff it contains $\varphi(t)$ for some closed term t and $\forall x\, \varphi(x)$ iff it contains $\varphi(t)$ for all closed terms t (Proposition 23.7). We'll then take the saturated consistent set Γ' and show that it can be extended to a saturated, consistent, and complete set Γ^* (Lemma 23.8). This set Γ^* is what we'll use to define our term model $\mathfrak{M}(\Gamma^*)$. The term model has the set of closed terms as its domain, and the interpretation of its predicate symbols is given by the atomic sentences in Γ^* (Definition 23.9). We'll use the properties of saturated, complete consistent sets to show that indeed $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\varphi \in \Gamma^*$ (Lemma 23.12), and thus in particular, $\mathfrak{M}(\Gamma^*) \vDash \Gamma$. Finally, we'll consider how to define a term model if Γ contains $=$ as well (Definition 23.16) and show that it satisfies Γ^* (Lemma 23.19).

---

content/first-order-logic/completeness/complete-consistent-sets.tex

23.3 Complete Consistent Sets of Sentences

fol:com:ccs:sec
fol:com:ccs:def:complete-set

Definition 23.1 (Complete set). A set Γ of sentences is *complete* iff for any sentence φ, either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$.

explanation Complete sets of sentences leave no questions unanswered. For any sentence φ, Γ "says" if φ is true or false. The importance of complete sets extends beyond the proof of the completeness theorem. A theory which is complete and axiomatizable, for instance, is always decidable.

explanation Complete consistent sets are important in the completeness proof since we can guarantee that every consistent set of sentences Γ is contained in a complete consistent set Γ^*. A complete consistent set contains, for each sentence φ, either φ or its negation $\neg\varphi$, but not both. This is true in particular for atomic sentences, so from a complete consistent set in a language suitably expanded by constant symbols, we can construct a structure where the interpretation of predicate symbols is defined according to which atomic sentences are in Γ^*. This structure can then be shown to make all sentences in Γ^* (and hence also all those in Γ) true. The proof of this latter fact requires that $\neg\varphi \in \Gamma^*$ iff $\varphi \notin \Gamma^*$, $(\varphi \vee \psi) \in \Gamma^*$ iff $\varphi \in \Gamma^*$ or $\psi \in \Gamma^*$, etc.

In what follows, we will often tacitly use the properties of reflexivity, monotonicity, and transitivity of $\vdash$ (see sections 19.8, 20.7, 21.7 and 22.6).

Proposition 23.2. *Suppose Γ is complete and consistent. Then:*

fol:com:ccs: prop:ccs

1. *If $\Gamma \vdash \varphi$, then $\varphi \in \Gamma$.* *fol:com:ccs: prop:ccs-prov-in*
2. *$\varphi \wedge \psi \in \Gamma$ iff both $\varphi \in \Gamma$ and $\psi \in \Gamma$.* *fol:com:ccs: prop:ccs-and*
3. *$\varphi \vee \psi \in \Gamma$ iff either $\varphi \in \Gamma$ or $\psi \in \Gamma$.* *fol:com:ccs: prop:ccs-or*
4. *$\varphi \to \psi \in \Gamma$ iff either $\varphi \notin \Gamma$ or $\psi \in \Gamma$.* *fol:com:ccs: prop:ccs-if*

Proof. Let us suppose for all of the following that Γ is complete and consistent.

1. If $\Gamma \vdash \varphi$, then $\varphi \in \Gamma$.

   Suppose that $\Gamma \vdash \varphi$. Suppose to the contrary that $\varphi \notin \Gamma$. Since Γ is complete, $\neg\varphi \in \Gamma$. By Propositions 19.20, 20.20, 21.20 and 22.28, Γ is inconsistent. This contradicts the assumption that Γ is consistent. Hence, it cannot be the case that $\varphi \notin \Gamma$, so $\varphi \in \Gamma$.

2. $\varphi \wedge \psi \in \Gamma$ iff both $\varphi \in \Gamma$ and $\psi \in \Gamma$:

   For the forward direction, suppose $\varphi \wedge \psi \in \Gamma$. Then by Propositions 19.22, 20.22, 21.22 and 22.30, item (1), $\Gamma \vdash \varphi$ and $\Gamma \vdash \psi$. By (1), $\varphi \in \Gamma$ and $\psi \in \Gamma$, as required.

   For the reverse direction, let $\varphi \in \Gamma$ and $\psi \in \Gamma$. By Propositions 19.22, 20.22, 21.22 and 22.30, item (2), $\Gamma \vdash \varphi \wedge \psi$. By (1), $\varphi \wedge \psi \in \Gamma$.

3. First we show that if $\varphi \vee \psi \in \Gamma$, then either $\varphi \in \Gamma$ or $\psi \in \Gamma$. Suppose $\varphi \vee \psi \in \Gamma$ but $\varphi \notin \Gamma$ and $\psi \notin \Gamma$. Since Γ is complete, $\neg\varphi \in \Gamma$ and $\neg\psi \in \Gamma$. By Propositions 19.23, 20.23, 21.23 and 22.31, item (1), Γ is inconsistent, a contradiction. Hence, either $\varphi \in \Gamma$ or $\psi \in \Gamma$.

   For the reverse direction, suppose that $\varphi \in \Gamma$ or $\psi \in \Gamma$. By Propositions 19.23, 20.23, 21.23 and 22.31, item (2), $\Gamma \vdash \varphi \vee \psi$. By (1), $\varphi \vee \psi \in \Gamma$, as required.

4. For the forward direction, suppose $\varphi \to \psi \in \Gamma$, and suppose to the contrary that $\varphi \in \Gamma$ and $\psi \notin \Gamma$. On these assumptions, $\varphi \to \psi \in \Gamma$ and $\varphi \in \Gamma$. By Propositions 19.24, 20.24, 21.24 and 22.32, item (1), $\Gamma \vdash \psi$. But then by (1), $\psi \in \Gamma$, contradicting the assumption that $\psi \notin \Gamma$.

   For the reverse direction, first consider the case where $\varphi \notin \Gamma$. Since Γ is complete, $\neg\varphi \in \Gamma$. By Propositions 19.24, 20.24, 21.24 and 22.32, item (2), $\Gamma \vdash \varphi \to \psi$. Again by (1), we get that $\varphi \to \psi \in \Gamma$, as required.

   Now consider the case where $\psi \in \Gamma$. By Propositions 19.24, 20.24, 21.24 and 22.32, item (2) again, $\Gamma \vdash \varphi \to \psi$. By (1), $\varphi \to \psi \in \Gamma$. □

Problem 23.1. Complete the proof of Proposition 23.2.

---

content/first-order-logic/completeness/henkin-expansions.tex

23.4 Henkin Expansion

fol:com:hen:sec

Part of the challenge in proving the completeness theorem is that the model we construct from a complete consistent set Γ must make all the quantified formulas in Γ true. In order to guarantee this, we use a trick due to Leon Henkin. In essence, the trick consists in expanding the language by infinitely many constant symbols and adding, for each formula with one free variable $\varphi(x)$ a formula of the form $\exists x\, \varphi(x) \to \varphi(c)$, where c is one of the new constant symbols. When we construct the structure satisfying Γ, this will guarantee that each true existential sentence has a witness among the new constants. explanation

fol:com:hen:prop:lang-exp

Proposition 23.3. *If Γ is consistent in $\mathcal{L}$ and $\mathcal{L}'$ is obtained from $\mathcal{L}$ by adding a denumerable set of new constant symbols d_0, d_1, ..., then Γ is consistent in $\mathcal{L}'$.*

Definition 23.4 (Saturated set). A set Γ of formulas of a language $\mathcal{L}$ is *saturated* iff for each formula $\varphi(x) \in \mathrm{Frm}(\mathcal{L})$ with one free variable x there is a constant symbol $c \in \mathcal{L}$ such that $\exists x\, \varphi(x) \to \varphi(c) \in \Gamma$.

The following definition will be used in the proof of the next theorem.

fol:com:hen:defn:henkin-exp

Definition 23.5. Let $\mathcal{L}'$ be as in Proposition 23.3. Fix an enumeration $\varphi_0(x_0), \varphi_1(x_1), \ldots$ of all formulas $\varphi_i(x_i)$ of $\mathcal{L}'$ in which one variable (x_i) occurs free. We define the sentences θ_n by induction on n.

Let c_0 be the first constant symbol among the d_i we added to $\mathcal{L}$ which does not occur in $\varphi_0(x_0)$. Assuming that $\theta_0, \ldots, \theta_{n-1}$ have already been defined, let c_n be the first among the new constant symbols d_i that occurs neither in $\theta_0, \ldots, \theta_{n-1}$ nor in $\varphi_n(x_n)$.

Now let θ_n be the formula $\exists x_n\, \varphi_n(x_n) \to \varphi_n(c_n)$.

Lemma 23.6. *Every consistent set Γ can be extended to a saturated consistent set Γ'.* fol:com:hen: lem:henkin

Proof. Given a consistent set of sentences Γ in a language $\mathcal{L}$, expand the language by adding a denumerable set of new constant symbols to form $\mathcal{L}'$. By Proposition 23.3, Γ is still consistent in the richer language. Further, let θ_i be as in Definition 23.5. Let

$$\Gamma_0 = \Gamma$$
$$\Gamma_{n+1} = \Gamma_n \cup \{\theta_n\}$$

i.e., $\Gamma_{n+1} = \Gamma \cup \{\theta_0, \ldots, \theta_n\}$, and let $\Gamma' = \bigcup_n \Gamma_n$. Γ' is clearly saturated.

If Γ' were inconsistent, then for some n, Γ_n would be inconsistent (Exercise: explain why). So to show that Γ' is consistent it suffices to show, by induction on n, that each set Γ_n is consistent.

The induction basis is simply the claim that $\Gamma_0 = \Gamma$ is consistent, which is the hypothesis of the theorem. For the induction step, suppose that Γ_n is consistent but $\Gamma_{n+1} = \Gamma_n \cup \{\theta_n\}$ is inconsistent. Recall that θ_n is $\exists x_n\, \varphi_n(x_n) \to \varphi_n(c_n)$, where $\varphi_n(x_n)$ is a formula of $\mathcal{L}'$ with only the variable x_n free. By the way we've chosen the c_n (see Definition 23.5), c_n does not occur in $\varphi_n(x_n)$ nor in Γ_n.

If $\Gamma_n \cup \{\theta_n\}$ is inconsistent, then $\Gamma_n \vdash \neg\theta_n$, and hence both of the following hold:

$$\Gamma_n \vdash \exists x_n\, \varphi_n(x_n) \qquad \Gamma_n \vdash \neg\varphi_n(c_n)$$

Since c_n does not occur in Γ_n or in $\varphi_n(x_n)$, Theorems 19.25, 20.25, 21.25 and 22.33 applies. From $\Gamma_n \vdash \neg\varphi_n(c_n)$, we obtain $\Gamma_n \vdash \forall x_n\, \neg\varphi_n(x_n)$. Thus we have that both $\Gamma_n \vdash \exists x_n\, \varphi_n(x_n)$ and $\Gamma_n \vdash \forall x_n\, \neg\varphi_n(x_n)$, so Γ_n itself is inconsistent. (Note that $\forall x_n\, \neg\varphi_n(x_n) \vdash \neg\exists x_n\, \varphi_n(x_n)$.) Contradiction: Γ_n was supposed to be consistent. Hence $\Gamma_n \cup \{\theta_n\}$ is consistent. □

explanation

We'll now show that *complete*, consistent sets which are saturated have the property that it contains a universally quantified sentence iff it contains all its instances and it contains an existentially quantified sentence iff it contains at least one instance. We'll use this to show that the structure we'll generate from a complete, consistent, saturated set makes all its quantified sentences true.

Proposition 23.7. *Suppose Γ is complete, consistent, and saturated.* fol:com:hen: prop:saturated-instances

1. *$\exists x\, \varphi(x) \in \Gamma$ iff $\varphi(t) \in \Gamma$ for at least one closed term t.*
2. *$\forall x\, \varphi(x) \in \Gamma$ iff $\varphi(t) \in \Gamma$ for all closed terms t.*

Proof. 1. First suppose that $\exists x\, \varphi(x) \in \Gamma$. Because Γ is saturated, $(\exists x\, \varphi(x) \to \varphi(c)) \in \Gamma$ for some constant symbol c. By Propositions 19.24, 20.24, 21.24 and 22.32, item (1), and Proposition 23.2(1), $\varphi(c) \in \Gamma$.

For the other direction, saturation is not necessary: Suppose $\varphi(t) \in \Gamma$. Then $\Gamma \vdash \exists x\, \varphi(x)$ by Propositions 19.26, 20.26, 21.26 and 22.34, item (1). By Proposition 23.2(1), $\exists x\, \varphi(x) \in \Gamma$.

2. Suppose that $\varphi(t) \in \Gamma$ for all closed terms t. By way of contradiction, assume $\forall x\, \varphi(x) \notin \Gamma$. Since Γ is complete, $\neg\forall x\, \varphi(x) \in \Gamma$. By saturation, $(\exists x\, \neg\varphi(x) \to \neg\varphi(c)) \in \Gamma$ for some constant symbol c. By assumption, since c is a closed term, $\varphi(c) \in \Gamma$. But this would make Γ inconsistent. (Exercise: give the derivation that shows

$$\neg\forall x\, \varphi(x), \exists x\, \neg\varphi(x) \to \neg\varphi(c), \varphi(c)$$

is inconsistent.)

For the reverse direction, we do not need saturation: Suppose $\forall x\, \varphi(x) \in \Gamma$. Then $\Gamma \vdash \varphi(t)$ by Propositions 19.26, 20.26, 21.26 and 22.34, item (2). We get $\varphi(t) \in \Gamma$ by Proposition 23.2. □

---

content/first-order-logic/completeness/lindenbaums-lemma.tex

23.5 Lindenbaum's Lemma

fol:com:lin:sec

We now prove a lemma that shows that any consistent set of sentences is contained in some set of sentences which is not just consistent, but also complete. The proof works by adding one sentence at a time, guaranteeing at each step that the set remains consistent. We do this so that for every φ, either φ or $\neg\varphi$ gets added at some stage. The union of all stages in that construction then contains either φ or its negation $\neg\varphi$ and is thus complete. It is also consistent, since we make sure at each stage not to introduce an inconsistency.

explanation

fol:com:lin:lem:lindenbaum

Lemma 23.8 (Lindenbaum's Lemma). *Every consistent set Γ in a language $\mathcal{L}$ can be extended to a complete and consistent set Γ^*.*

Proof. Let Γ be consistent. Let φ_0, φ_1, ... be an enumeration of all the sentences of $\mathcal{L}$. Define $\Gamma_0 = \Gamma$, and

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\varphi_n\} & \text{if } \Gamma_n \cup \{\varphi_n\} \text{ is consistent;} \\ \Gamma_n \cup \{\neg\varphi_n\} & \text{otherwise.} \end{cases}$$

Let $\Gamma^* = \bigcup_{n \geq 0} \Gamma_n$.

Each Γ_n is consistent: Γ_0 is consistent by definition. If $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$, this is because the latter is consistent. If it isn't, $\Gamma_{n+1} = \Gamma_n \cup \{\neg\varphi_n\}$. We have to verify that $\Gamma_n \cup \{\neg\varphi_n\}$ is consistent. Suppose it's not. Then *both* $\Gamma_n \cup \{\varphi_n\}$ and $\Gamma_n \cup \{\neg\varphi_n\}$ are inconsistent. This means that Γ_n would be inconsistent by Propositions 19.21, 20.21, 21.21 and 22.29, contrary to the induction hypothesis.

For every n and every $i < n$, $\Gamma_i \subseteq \Gamma_n$. This follows by a simple induction on n. For $n = 0$, there are no $i < 0$, so the claim holds automatically. For the inductive step, suppose it is true for n. We show that if $i < n + 1$ then $\Gamma_i \subseteq \Gamma_{n+1}$. We have $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$ or $= \Gamma_n \cup \{\neg\varphi_n\}$ by construction. So

$\Gamma_n \subseteq \Gamma_{n+1}$. If $i < n+1$, then $\Gamma_i \subseteq \Gamma_n$ by inductive hypothesis (if $i < n$) or the trivial fact that $\Gamma_n \subseteq \Gamma_n$ (if $i = n$). We get that $\Gamma_i \subseteq \Gamma_{n+1}$ by transitivity of $\subseteq$.

From this it follows that Γ^* is consistent. Here's why: Let $\Gamma' \subseteq \Gamma^*$ be finite. Each $\psi \in \Gamma'$ is also in Γ_i for some i. Let n be the largest of these. Since $\Gamma_i \subseteq \Gamma_n$ if $i \leq n$, every $\psi \in \Gamma'$ is also $\in \Gamma_n$, i.e., $\Gamma' \subseteq \Gamma_n$, and Γ_n is consistent. So, every finite subset $\Gamma' \subseteq \Gamma^*$ is consistent. By Propositions 19.17, 20.17, 21.17 and 22.21, Γ^* is consistent.

Every sentence of $\mathrm{Frm}(\mathcal{L})$ appears on the list used to define Γ^*. If $\varphi_n \notin \Gamma^*$, then that is because $\Gamma_n \cup \{\varphi_n\}$ was inconsistent. But then $\neg\varphi_n \in \Gamma^*$, so Γ^* is complete. □

---

content/first-order-logic/completeness/construction-of-model.tex

23.6 Construction of a Model

explanation

Right now we are not concerned about $=$, i.e., we only want to show that a consistent set Γ of sentences not containing $=$ is satisfiable. We first extend Γ to a consistent, complete, and saturated set Γ^*. In this case, the definition of a model $\mathfrak{M}(\Gamma^*)$ is simple: We take the set of closed terms of $\mathcal{L}'$ as the domain. We assign every constant symbol to itself, and make sure that more generally, for every closed term t, $\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t) = t$. The predicate symbols are assigned extensions in such a way that an atomic sentence is true in $\mathfrak{M}(\Gamma^*)$ iff it is in Γ^*. This will obviously make all the atomic sentences in Γ^* true in $\mathfrak{M}(\Gamma^*)$. The rest are true provided the Γ^* we start with is consistent, complete, and saturated.

fol:com:mod:sec

Definition 23.9 (Term model). Let Γ^* be a complete and consistent, saturated set of sentences in a language $\mathcal{L}$. The *term model* $\mathfrak{M}(\Gamma^*)$ of Γ^* is the structure defined as follows:

fol:com:mod:defn:termmodel

1. The domain $|\mathfrak{M}(\Gamma^*)|$ is the set of all closed terms of $\mathcal{L}$.
2. The interpretation of a constant symbol c is c itself: $c^{\mathfrak{M}(\Gamma^*)} = c$.
3. The function symbol f is assigned the function which, given as arguments the closed terms $t_1, \ldots, t_n$, has as value the closed term $f(t_1, \ldots, t_n)$:
$$f^{\mathfrak{M}(\Gamma^*)}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$$
4. If R is an n-place predicate symbol, then
$$\langle t_1, \ldots, t_n \rangle \in R^{\mathfrak{M}(\Gamma^*)} \text{ iff } R(t_1, \ldots, t_n) \in \Gamma^*.$$

We will now check that we indeed have $\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t) = t$.

fol:com:mod: lem:val-in-termmodel **Lemma 23.10.** *Let $\mathfrak{M}(\Gamma^*)$ be the term model of Definition 23.9, then $\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t) = t$.*

Proof. The proof is by induction on t, where the base case, when t is a constant symbol, follows directly from the definition of the term model. For the induction step assume $t_1, \ldots, t_n$ are closed terms such that $\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t_i) = t_i$ and that f is an n-ary function symbol. Then

$$\begin{aligned}\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(f(t_1, \ldots, t_n)) &= f^{\mathfrak{M}(\Gamma^*)}(\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t_n)) \\ &= f^{\mathfrak{M}(\Gamma^*)}(t_1, \ldots, t_n) \\ &= f(t_1, \ldots, t_n),\end{aligned}$$

and so by induction this holds for every closed term t. □

explanation

A structure $\mathfrak{M}$ may make an existentially quantified sentence $\exists x\, \varphi(x)$ true without there being an instance $\varphi(t)$ that it makes true. A structure $\mathfrak{M}$ may make all instances $\varphi(t)$ of a universally quantified sentence $\forall x\, \varphi(x)$ true, without making $\forall x\, \varphi(x)$ true. This is because in general not every element of $|\mathfrak{M}|$ is the value of a closed term ($\mathfrak{M}$ may not be covered). This is the reason the satisfaction relation is defined via variable assignments. However, for our term model $\mathfrak{M}(\Gamma^*)$ this wouldn't be necessary—because it is covered. This is the content of the next result.

fol:com:mod: prop:quant-termmodel **Proposition 23.11.** *Let $\mathfrak{M}(\Gamma^*)$ be the term model of Definition 23.9.*

1. *$\mathfrak{M}(\Gamma^*) \vDash \exists x\, \varphi(x)$ iff $\mathfrak{M}(\Gamma^*) \vDash \varphi(t)$ for at least one closed term t.*
2. *$\mathfrak{M}(\Gamma^*) \vDash \forall x\, \varphi(x)$ iff $\mathfrak{M}(\Gamma^*) \vDash \varphi(t)$ for all closed terms t.*

Proof. 1. By Proposition 16.19, $\mathfrak{M}(\Gamma^*) \vDash \exists x\, \varphi(x)$ iff for at least one variable assignment s, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(x)$. As $|\mathfrak{M}(\Gamma^*)|$ consists of the closed terms of $\mathcal{L}$, this is the case iff there is at least one closed term t such that $s(x) = t$ and $\mathfrak{M}(\Gamma^*), s \vDash \varphi(x)$. By Proposition 16.23, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(x)$ iff $\mathfrak{M}(\Gamma^*), s \vDash \varphi(t)$, where $s(x) = t$. By Proposition 16.18, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(t)$ iff $\mathfrak{M}(\Gamma^*) \vDash \varphi(t)$, since $\varphi(t)$ is a sentence.

2. By Proposition 16.19, $\mathfrak{M}(\Gamma^*) \vDash \forall x\, \varphi(x)$ iff for every variable assignment s, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(x)$. Recall that $|\mathfrak{M}(\Gamma^*)|$ consists of the closed terms of $\mathcal{L}$, so for every closed term t, $s(x) = t$ is such a variable assignment, and for any variable assignment, $s(x)$ is some closed term t. By Proposition 16.23, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(x)$ iff $\mathfrak{M}(\Gamma^*), s \vDash \varphi(t)$, where $s(x) = t$. By Proposition 16.18, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(t)$ iff $\mathfrak{M}(\Gamma^*) \vDash \varphi(t)$, since $\varphi(t)$ is a sentence. □

fol:com:mod: lem:truth **Lemma 23.12 (Truth Lemma).** *Suppose φ does not contain $=$. Then $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\varphi \in \Gamma^*$.*

Proof. We prove both directions simultaneously, and by induction on φ.

1. $\varphi \equiv \bot$: $\mathfrak{M}(\Gamma^*) \nvDash \bot$ by definition of satisfaction. On the other hand, $\bot \notin \Gamma^*$ since Γ^* is consistent.

2. $\varphi \equiv \top$: $\mathfrak{M}(\Gamma^*) \vDash \top$ by definition of satisfaction. On the other hand, $\top \in \Gamma^*$ since Γ^* is consistent and complete, and $\Gamma^* \vdash \top$.

3. $\varphi \equiv R(t_1, \ldots, t_n)$: $\mathfrak{M}(\Gamma^*) \vDash R(t_1, \ldots, t_n)$ iff $\langle t_1, \ldots, t_n \rangle \in R^{\mathfrak{M}(\Gamma^*)}$ (by the definition of satisfaction) iff $R(t_1, \ldots, t_n) \in \Gamma^*$ (by the construction of $\mathfrak{M}(\Gamma^*)$).

4. $\varphi \equiv \neg\psi$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{M}(\Gamma^*) \nvDash \psi$ (by definition of satisfaction). By induction hypothesis, $\mathfrak{M}(\Gamma^*) \nvDash \psi$ iff $\psi \notin \Gamma^*$. Since Γ^* is consistent and complete, $\psi \notin \Gamma^*$ iff $\neg\psi \in \Gamma^*$.

5. $\varphi \equiv \psi \wedge \chi$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff we have both $\mathfrak{M}(\Gamma^*) \vDash \psi$ and $\mathfrak{M}(\Gamma^*) \vDash \chi$ (by definition of satisfaction) iff both $\psi \in \Gamma^*$ and $\chi \in \Gamma^*$ (by the induction hypothesis). By Proposition 23.2(2), this is the case iff $(\psi \wedge \chi) \in \Gamma^*$.

6. $\varphi \equiv \psi \vee \chi$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{M}(\Gamma^*) \vDash \psi$ or $\mathfrak{M}(\Gamma^*) \vDash \chi$ (by definition of satisfaction) iff $\psi \in \Gamma^*$ or $\chi \in \Gamma^*$ (by induction hypothesis). This is the case iff $(\psi \vee \chi) \in \Gamma^*$ (by Proposition 23.2(3)).

7. $\varphi \equiv \psi \to \chi$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{M}(\Gamma^*) \nvDash \psi$ or $\mathfrak{M}(\Gamma^*) \vDash \chi$ (by definition of satisfaction) iff $\psi \notin \Gamma^*$ or $\chi \in \Gamma^*$ (by induction hypothesis). This is the case iff $(\psi \to \chi) \in \Gamma^*$ (by Proposition 23.2(4)).

8. $\varphi \equiv \forall x\, \psi(x)$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{M}(\Gamma^*) \vDash \psi(t)$ for all terms t (Proposition 23.11). By induction hypothesis, this is the case iff $\psi(t) \in \Gamma^*$ for all terms t, by Proposition 23.7, this in turn is the case iff $\forall x\, \varphi(x) \in \Gamma^*$.

9. $\varphi \equiv \exists x\, \psi(x)$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{M}(\Gamma^*) \vDash \psi(t)$ for at least one term t (Proposition 23.11). By induction hypothesis, this is the case iff $\psi(t) \in \Gamma^*$ for at least one term t. By Proposition 23.7, this in turn is the case iff $\exists x\, \psi(x) \in \Gamma^*$. □

---

content/first-order-logic/completeness/identity.tex

23.7 Identity

explanation

The construction of the term model given in the preceding section is enough to establish completeness for first-order logic for sets Γ that do not contain $=$. The term model satisfies every $\varphi \in \Gamma^*$ which does not contain $=$ (and hence all $\varphi \in \Gamma$). It does not work, however, if $=$ is present. The reason is that Γ^* then may contain a sentence $t = t'$, but in the term model the value of any term is that term itself. Hence, if t and t' are different terms, their values in the term model—i.e., t and t', respectively—are different, and so $t = t'$ is false. We can fix this, however, using a construction known as "factoring."

fol:com:ide:sec

Definition 23.13. Let Γ^* be a consistent and complete set of sentences in $\mathcal{L}$. We define the relation $\approx$ on the set of closed terms of $\mathcal{L}$ by

$$t \approx t' \quad \text{iff} \quad t = t' \in \Gamma^*$$

fol:com:ide: prop:approx-equiv

Proposition 23.14. *The relation $\approx$ has the following properties:*

1. *$\approx$ is reflexive.*
2. *$\approx$ is symmetric.*
3. *$\approx$ is transitive.*
4. *If $t \approx t'$, f is a function symbol, and $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ are closed terms, then*

$$f(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n) \approx f(t_1, \ldots, t_{i-1}, t', t_{i+1}, \ldots, t_n).$$

5. *If $t \approx t'$, R is a predicate symbol, and $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ are closed terms, then*

$$R(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n) \in \Gamma^* \text{ iff } R(t_1, \ldots, t_{i-1}, t', t_{i+1}, \ldots, t_n) \in \Gamma^*.$$

Proof. Since Γ^* is consistent and complete, $t = t' \in \Gamma^*$ iff $\Gamma^* \vdash t = t'$. Thus it is enough to show the following:

1. $\Gamma^* \vdash t = t$ for all closed terms t.
2. If $\Gamma^* \vdash t = t'$ then $\Gamma^* \vdash t' = t$.
3. If $\Gamma^* \vdash t = t'$ and $\Gamma^* \vdash t' = t''$, then $\Gamma^* \vdash t = t''$.
4. If $\Gamma^* \vdash t = t'$, then

$$\Gamma^* \vdash f(t_1, \ldots, t_{i-1}, t, t_{i+1}, , \ldots, t_n) = f(t_1, \ldots, t_{i-1}, t', t_{i+1}, \ldots, t_n)$$

   for every n-place function symbol f and closed terms $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$.

5. If $\Gamma^* \vdash t = t'$ and $\Gamma^* \vdash R(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n)$, then $\Gamma^* \vdash R(t_1, \ldots, t_{i-1}, t', t_{i+1}, \ldots, t_n)$ for every n-place predicate symbol R and closed terms $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$. □

Problem 23.2. Complete the proof of Proposition 23.14.

Definition 23.15. Suppose Γ^* is a consistent and complete set in a language $\mathcal{L}$, t is a closed term, and $\approx$ as in the previous definition. Then:

$$[t]_\approx = \{t' : t' \in \mathrm{Trm}(\mathcal{L}), t \approx t'\}$$

and $\mathrm{Trm}(\mathcal{L})/_\approx = \{[t]_\approx : t \in \mathrm{Trm}(\mathcal{L})\}$.

Definition 23.16. Let $\mathfrak{M} = \mathfrak{M}(\Gamma^*)$ be the term model for Γ^* from Definition 23.9. Then $\mathfrak{M}/{\approx}$ is the following structure: fol:com:ide: defn:term-model-factor

1. $|\mathfrak{M}/{\approx}| = \mathrm{Trm}(\mathcal{L})/{\approx}$.
2. $c^{\mathfrak{M}/\approx} = [c]_\approx$
3. $f^{\mathfrak{M}/\approx}([t_1]_\approx, \ldots, [t_n]_\approx) = [f(t_1, \ldots, t_n)]_\approx$
4. $\langle [t_1]_\approx, \ldots, [t_n]_\approx \rangle \in R^{\mathfrak{M}/\approx}$ iff $\mathfrak{M} \vDash R(t_1, \ldots, t_n)$, i.e., iff $R(t_1, \ldots, t_n) \in \Gamma^*$.

explanation

Note that we have defined $f^{\mathfrak{M}/\approx}$ and $R^{\mathfrak{M}/\approx}$ for elements of $\mathrm{Trm}(\mathcal{L})/{\approx}$ by referring to them as $[t]_\approx$, i.e., via *representatives* $t \in [t]_\approx$. We have to make sure that these definitions do not depend on the choice of these representatives, i.e., that for some other choices t' which determine the same equivalence classes ($[t]_\approx = [t']_\approx$), the definitions yield the same result. For instance, if R is a one-place predicate symbol, the last clause of the definition says that $[t]_\approx \in R^{\mathfrak{M}/\approx}$ iff $\mathfrak{M} \vDash R(t)$. If for some other term t' with $t \approx t'$, $\mathfrak{M} \nvDash R(t)$, then the definition would require $[t']_\approx \notin R^{\mathfrak{M}/\approx}$. If $t \approx t'$, then $[t]_\approx = [t']_\approx$, but we can't have both $[t]_\approx \in R^{\mathfrak{M}/\approx}$ and $[t]_\approx \notin R^{\mathfrak{M}/\approx}$. However, Proposition 23.14 guarantees that this cannot happen.

Proposition 23.17. *$\mathfrak{M}/{\approx}$ is well defined, i.e., if $t_1, \ldots, t_n, t'_1, \ldots, t'_n$ are closed terms, and $t_i \approx t'_i$ then*

1. *$[f(t_1, \ldots, t_n)]_\approx = [f(t'_1, \ldots, t'_n)]_\approx$, i.e.,*

$$f(t_1, \ldots, t_n) \approx f(t'_1, \ldots, t'_n)$$

   and

2. *$\mathfrak{M} \vDash R(t_1, \ldots, t_n)$ iff $\mathfrak{M} \vDash R(t'_1, \ldots, t'_n)$, i.e.,*

$$R(t_1, \ldots, t_n) \in \Gamma^* \text{ iff } R(t'_1, \ldots, t'_n) \in \Gamma^*.$$

Proof. Follows from Proposition 23.14 by induction on n. □

As in the case of the term model, before proving the truth lemma we need the following lemma.

Lemma 23.18. *Let $\mathfrak{M} = \mathfrak{M}(\Gamma^*)$, then $\mathrm{Val}^{\mathfrak{M}/\approx}(t) = [t]_\approx$.* fol:com:ide: lem:val-in-termmodel-factored

Proof. The proof is similar to that of Lemma 23.10. □

Problem 23.3. Complete the proof of Lemma 23.18.

Lemma 23.19. *$\mathfrak{M}/{\approx} \vDash \varphi$ iff $\varphi \in \Gamma^*$ for all sentences φ.* fol:com:ide: lem:truth

Proof. By induction on φ, just as in the proof of Lemma 23.12. The only case that needs additional attention is when $\varphi \equiv t = t'$.

$$\begin{aligned}\mathfrak{M}/_{\approx} \vDash t = t' &\text{ iff } [t]_{\approx} = [t']_{\approx} \text{ (by definition of } \mathfrak{M}/_{\approx})\\ &\text{iff } t \approx t' \text{ (by definition of } [t]_{\approx})\\ &\text{iff } t = t' \in \Gamma^* \text{ (by definition of } \approx).\end{aligned}$$ □

Note that while $\mathfrak{M}(\Gamma^*)$ is always enumerable and infinite, $\mathfrak{M}/_{\approx}$ may be finite, since it may turn out that there are only finitely many classes $[t]_{\approx}$. This is to be expected, since Γ may contain sentences which require any structure in which they are true to be finite. For instance, $\forall x \forall y\, x = y$ is a consistent sentence, but is satisfied only in structures with a domain that contains exactly one element. digression

---

content/first-order-logic/completeness/completeness-thm.tex

23.8 The Completeness Theorem

fol:com:cth:sec Let's combine our results: we arrive at the completeness theorem. explanation

fol:com:cth:thm:completeness **Theorem 23.20 (Completeness Theorem).** *Let Γ be a set of sentences. If Γ is consistent, it is satisfiable.*

Proof. Suppose Γ is consistent. By Lemma 23.6, there is a saturated consistent set $\Gamma' \supseteq \Gamma$. By Lemma 23.8, there is a $\Gamma^* \supseteq \Gamma'$ which is consistent and complete. Since $\Gamma' \subseteq \Gamma^*$, for each formula $\varphi(x)$, Γ^* contains a sentence of the form $\exists x\, \varphi(x) \to \varphi(c)$ and so Γ^* is saturated. If Γ does not contain $=$, then by Lemma 23.12, $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\varphi \in \Gamma^*$. From this it follows in particular that for all $\varphi \in \Gamma$, $\mathfrak{M}(\Gamma^*) \vDash \varphi$, so Γ is satisfiable. If Γ does contain $=$, then by Lemma 23.19, for all sentences φ, $\mathfrak{M}/_{\approx} \vDash \varphi$ iff $\varphi \in \Gamma^*$. In particular, $\mathfrak{M}/_{\approx} \vDash \varphi$ for all $\varphi \in \Gamma$, so Γ is satisfiable. □

fol:com:cth:cor:completeness **Corollary 23.21 (Completeness Theorem, Second Version).** *For all Γ and sentences φ: if $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$.*

Proof. Note that the Γ's in Corollary 23.21 and Theorem 23.20 are universally quantified. To make sure we do not confuse ourselves, let us restate Theorem 23.20 using a different variable: for any set of sentences Δ, if Δ is consistent, it is satisfiable. By contraposition, if Δ is not satisfiable, then Δ is inconsistent. We will use this to prove the corollary.

Suppose that $\Gamma \vDash \varphi$. Then $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable by Proposition 16.28. Taking $\Gamma \cup \{\neg\varphi\}$ as our Δ, the previous version of Theorem 23.20 gives us that $\Gamma \cup \{\neg\varphi\}$ is inconsistent. By Propositions 19.19, 20.19, 21.19 and 22.27, $\Gamma \vdash \varphi$. □

Problem 23.4. Use Corollary 23.21 to prove Theorem 23.20, thus showing that the two formulations of the completeness theorem are equivalent.

Problem 23.5. In order for a derivation system to be complete, its rules must be strong enough to prove every unsatisfiable set inconsistent. Which of the rules of derivation were necessary to prove completeness? Are any of these rules not used anywhere in the proof? In order to answer these questions, make a list or diagram that shows which of the rules of derivation were used in which results that lead up to the proof of Theorem 23.20. Be sure to note any tacit uses of rules in these proofs.

---

content/first-order-logic/completeness/compactness.tex

23.9 The Compactness Theorem

fol:com:com:sec

One important consequence of the completeness theorem is the compactness theorem. The compactness theorem states that if each *finite* subset of a set of sentences is satisfiable, the entire set is satisfiable—even if the set itself is infinite. This is far from obvious. There is nothing that seems to rule out, at first glance at least, the possibility of there being infinite sets of sentences which are contradictory, but the contradiction only arises, so to speak, from the infinite number. The compactness theorem says that such a scenario can be ruled out: there are no unsatisfiable infinite sets of sentences each finite subset of which is satisfiable. Like the completeness theorem, it has a version related to entailment: if an infinite set of sentences entails something, already a finite subset does.

Definition 23.22. A set Γ of formulas is *finitely satisfiable* iff every finite $\Gamma_0 \subseteq \Gamma$ is satisfiable.

fol:com:com:thm:compactness

Theorem 23.23 (Compactness Theorem). *The following hold for any sentences Γ and φ:*

1. *$\Gamma \vDash \varphi$ iff there is a finite $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vDash \varphi$.*
2. *Γ is satisfiable iff it is finitely satisfiable.*

Proof. We prove (2). If Γ is satisfiable, then there is a structure $\mathfrak{M}$ such that $\mathfrak{M} \vDash \varphi$ for all $\varphi \in \Gamma$. Of course, this $\mathfrak{M}$ also satisfies every finite subset of Γ, so Γ is finitely satisfiable.

Now suppose that Γ is finitely satisfiable. Then every finite subset $\Gamma_0 \subseteq \Gamma$ is satisfiable. By soundness (Corollaries 19.31, 20.29, 21.31 and 22.38), every finite subset is consistent. Then Γ itself must be consistent by Propositions 19.17, 20.17, 21.17 and 22.21. By completeness (Theorem 23.20), since Γ is consistent, it is satisfiable. □

Problem 23.6. Prove (1) of Theorem 23.23.

Example 23.24. In every model $\mathfrak{M}$ of a theory Γ, each term t of course picks out an element of $|\mathfrak{M}|$. Can we guarantee that it is also true that every element of $|\mathfrak{M}|$ is picked out by some term or other? In other words, are there theories Γ all models of which are covered? The compactness theorem shows that this is not the case if Γ has infinite models. Here's how to see this: Let $\mathfrak{M}$ be an infinite model of Γ, and let c be a constant symbol not in the language of Γ. Let Δ be the set of all sentences $c \neq t$ for t a term in the language $\mathcal{L}$ of Γ, i.e.,

$$\Delta = \{c \neq t : t \in \mathrm{Trm}(\mathcal{L})\}.$$

A finite subset of $\Gamma \cup \Delta$ can be written as $\Gamma' \cup \Delta'$, with $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$. Since Δ' is finite, it can contain only finitely many terms. Let $a \in |\mathfrak{M}|$ be an element of $|\mathfrak{M}|$ not picked out by any of them, and let $\mathfrak{M}'$ be the structure that is just like $\mathfrak{M}$, but also $c^{\mathfrak{M}'} = a$. Since $a \neq \mathrm{Val}^{\mathfrak{M}}(t)$ for all t occurring in Δ', $\mathfrak{M}' \vDash \Delta'$. Since $\mathfrak{M} \vDash \Gamma$, $\Gamma' \subseteq \Gamma$, and c does not occur in Γ, also $\mathfrak{M}' \vDash \Gamma'$. Together, $\mathfrak{M}' \vDash \Gamma' \cup \Delta'$ for every finite subset $\Gamma' \cup \Delta'$ of $\Gamma \cup \Delta$. So every finite subset of $\Gamma \cup \Delta$ is satisfiable. By compactness, $\Gamma \cup \Delta$ itself is satisfiable. So there are models $\mathfrak{M} \vDash \Gamma \cup \Delta$. Every such $\mathfrak{M}$ is a model of Γ, but is not covered, since $\mathrm{Val}^{\mathfrak{M}}(c) \neq \mathrm{Val}^{\mathfrak{M}}(t)$ for all terms t of $\mathcal{L}$.

Example 23.25. Consider a language $\mathcal{L}$ containing the predicate symbol $<$, constant symbols $\mathtt{0}$, $\mathtt{1}$, and function symbols $+$, $\times$, and $-$. Let Γ be the set of all sentences in this language true in the structure $\mathfrak{Q}$ with domain $\mathbb{Q}$ and the obvious interpretations. Γ is the set of all sentences of $\mathcal{L}$ true about the rational numbers. Of course, in $\mathbb{Q}$ (and even in $\mathbb{R}$), there are no numbers r which are greater than 0 but less than $1/k$ for all $k \in \mathbb{Z}^+$. Such a number, if it existed, would be an *infinitesimal:* non-zero, but infinitely small. The compactness theorem can be used to show that there are models of Γ in which infinitesimals exist. We do not have a function symbol for division in our language (division by zero is undefined, and function symbols have to be interpreted by total functions). However, we can still express that $r < 1/k$, since this is the case iff $r \cdot k < 1$. Now let c be a new constant symbol and let Δ be

$$\{\mathtt{0} < c\} \cup \{c \times \overline{k} < \mathtt{1} : k \in \mathbb{Z}^+\}$$

(where $\overline{k} = (\mathtt{1} + (\mathtt{1} + \cdots + (\mathtt{1} + \mathtt{1}) \ldots))$ with k $\mathtt{1}$'s). For any finite subset Δ_0 of Δ there is a K such that for all the sentences $c \times \overline{k} < \mathtt{1}$ in Δ_0 have $k < K$. If we expand $\mathfrak{Q}$ to $\mathfrak{Q}'$ with $c^{\mathfrak{Q}'} = 1/K$ we have that $\mathfrak{Q}' \vDash \Gamma_0 \cup \Delta_0$ for any finite $\Gamma_0 \subseteq \Gamma$, and so $\Gamma \cup \Delta$ is finitely satisfiable (Exercise: prove this in detail). By compactness, $\Gamma \cup \Delta$ is satisfiable. Any model $\mathfrak{S}$ of $\Gamma \cup \Delta$ contains an infinitesimal, namely $c^{\mathfrak{S}}$.

Problem 23.7. In the standard model of arithmetic $\mathfrak{N}$, there is no element $k \in |\mathfrak{N}|$ which satisfies every formula $\overline{n} < x$ (where $\overline{n}$ is $\mathtt{0}'^{\cdots'}$ with n $'$'s). Use the compactness theorem to show that the set of sentences in the language of arithmetic which are true in the standard model of arithmetic $\mathfrak{N}$ are also true in a structure $\mathfrak{N}'$ that contains an element which *does* satisfy every formula $\overline{n} < x$.

Example 23.26. We know that first-order logic with identity predicate can express that the size of the domain must have some minimal size: The sentence $\varphi_{\geq n}$ (which says "there are at least n distinct objects") is true only in structures where $|\mathfrak{M}|$ has at least n objects. So if we take

$$\Delta = \{\varphi_{\geq n} : n \geq 1\}$$

then any model of Δ must be infinite. Thus, we can guarantee that a theory only has infinite models by adding Δ to it: the models of $\Gamma \cup \Delta$ are all and only the infinite models of Γ.

So first-order logic can express infinitude. The compactness theorem shows that it cannot express finitude, however. For suppose some set of sentences Λ were satisfied in all and only finite structures. Then $\Delta \cup \Lambda$ is finitely satisfiable. Why? Suppose $\Delta' \cup \Lambda' \subseteq \Delta \cup \Lambda$ is finite with $\Delta' \subseteq \Delta$ and $\Lambda' \subseteq \Lambda$. Let n be the largest number such that $\varphi_{\geq n} \in \Delta'$. Λ, being satisfied in all finite structures, has a model $\mathfrak{M}$ with finitely many but $\geq n$ elements. But then $\mathfrak{M} \vDash \Delta' \cup \Lambda'$. By compactness, $\Delta \cup \Lambda$ has an infinite model, contradicting the assumption that Λ is satisfied only in finite structures.

---

content/first-order-logic/completeness/compactness-direct.tex

23.10 A Direct Proof of the Compactness Theorem

fol:com:cpd:sec

We can prove the Compactness Theorem directly, without appealing to the Completeness Theorem, using the same ideas as in the proof of the completeness theorem. In the proof of the Completeness Theorem we started with a consistent set Γ of sentences, expanded it to a consistent, saturated, and complete set Γ^* of sentences, and then showed that in the term model $\mathfrak{M}(\Gamma^*)$ constructed from Γ^*, all sentences of Γ are true, so Γ is satisfiable.

We can use the same method to show that a finitely satisfiable set of sentences is satisfiable. We just have to prove the corresponding versions of the results leading to the truth lemma where we replace "consistent" with "finitely satisfiable."

fol:com:cpd:prop:fsat-ccs

Proposition 23.27. *Suppose Γ is complete and finitely satisfiable. Then:*

1. *$(\varphi \wedge \psi) \in \Gamma$ iff both $\varphi \in \Gamma$ and $\psi \in \Gamma$.*
2. *$(\varphi \vee \psi) \in \Gamma$ iff either $\varphi \in \Gamma$ or $\psi \in \Gamma$.*
3. *$(\varphi \to \psi) \in \Gamma$ iff either $\varphi \notin \Gamma$ or $\psi \in \Gamma$.*

Problem 23.8. Prove Proposition 23.27. Avoid the use of $\vdash$.

fol:com:cpd:lem:fsat-henkin

Lemma 23.28. *Every finitely satisfiable set Γ can be extended to a saturated finitely satisfiable set Γ'.*

Problem 23.9. Prove Lemma 23.28. (Hint: The crucial step is to show that if Γ_n is finitely satisfiable, so is $\Gamma_n \cup \{\theta_n\}$, without any appeal to derivations or consistency.)

fol:com:cpd: prop:fsat-instances

Proposition 23.29. *Suppose Γ is complete, finitely satisfiable, and saturated.*

1. *$\exists x\, \varphi(x) \in \Gamma$ iff $\varphi(t) \in \Gamma$ for at least one closed term t.*
2. *$\forall x\, \varphi(x) \in \Gamma$ iff $\varphi(t) \in \Gamma$ for all closed terms t.*

Problem 23.10. Prove Proposition 23.29.

fol:com:cpd: lem:fsat-lindenbaum

Lemma 23.30. *Every finitely satisfiable set Γ can be extended to a complete and finitely satisfiable set Γ^*.*

Problem 23.11. Prove Lemma 23.30. (Hint: the crucial step is to show that if Γ_n is finitely satisfiable, then either $\Gamma_n \cup \{\varphi_n\}$ or $\Gamma_n \cup \{\neg\varphi_n\}$ is finitely satisfiable.)

fol:com:cpd: thm:compactness-direct

Theorem 23.31 (Compactness). *Γ is satisfiable if and only if it is finitely satisfiable.*

Proof. If Γ is satisfiable, then there is a structure $\mathfrak{M}$ such that $\mathfrak{M} \vDash \varphi$ for all $\varphi \in \Gamma$. Of course, this $\mathfrak{M}$ also satisfies every finite subset of Γ, so Γ is finitely satisfiable.

Now suppose that Γ is finitely satisfiable. By Lemma 23.28, there is a finitely satisfiable, saturated set $\Gamma' \supseteq \Gamma$. By Lemma 23.30, Γ' can be extended to a complete and finitely satisfiable set Γ^*, and Γ^* is still saturated. Construct the term model $\mathfrak{M}(\Gamma^*)$ as in Definition 23.9. Note that Proposition 23.11 did not rely on the fact that Γ^* is consistent (or complete or saturated, for that matter), but just on the fact that $\mathfrak{M}(\Gamma^*)$ is covered. The proof of the Truth Lemma (Lemma 23.12) goes through if we replace references to Proposition 23.2 and Proposition 23.7 by references to Proposition 23.27 and Proposition 23.29□

Problem 23.12. Write out the complete proof of the Truth Lemma (Lemma 23.12) in the version required for the proof of Theorem 23.31.

---

content/first-order-logic/completeness/downward-ls.tex

23.11 The Löwenheim–Skolem Theorem

fol:com:dls: sec

The Löwenheim–Skolem Theorem says that if a theory has an infinite model, then it also has a model that is at most denumerable. An immediate consequence of this fact is that first-order logic cannot express that the size of a structure is non-enumerable: any sentence or set of sentences satisfied in all non-enumerable structures is also satisfied in some enumerable structure.

Theorem 23.32. *If Γ is consistent then it has an enumerable model, i.e., it is satisfiable in a structure whose domain is either finite or denumerable.* *fol:com:dls: thm:downward-ls*

Proof. If Γ is consistent, the structure $\mathfrak{M}$ delivered by the proof of the completeness theorem has a domain $|\mathfrak{M}|$ that is no larger than the set of the terms of the language $\mathcal{L}$. So $\mathfrak{M}$ is at most denumerable. □

Theorem 23.33. *If Γ is a consistent set of sentences in the language of first-order logic without identity, then it has a denumerable model, i.e., it is satisfiable in a structure whose domain is infinite and enumerable.* *fol:com:dls: noidentity-ls*

Proof. If Γ is consistent and contains no sentences in which identity appears, then the structure $\mathfrak{M}$ delivered by the proof of the completeness theorem has a domain $|\mathfrak{M}|$ identical to the set of terms of the language $\mathcal{L}'$. So $\mathfrak{M}$ is denumerable, since $\mathrm{Trm}(\mathcal{L}')$ is. □

Example 23.34 (Skolem's Paradox). Zermelo–Fraenkel set theory **ZFC** is a very powerful framework in which practically all mathematical statements can be expressed, including facts about the sizes of sets. So for instance, **ZFC** can prove that the set $\mathbb{R}$ of real numbers is non-enumerable, it can prove Cantor's Theorem that the power set of any set is larger than the set itself, etc. If **ZFC** is consistent, its models are all infinite, and moreover, they all contain elements about which the theory says that they are non-enumerable, such as the element that makes true the theorem of **ZFC** that the power set of the natural numbers exists. By the Löwenheim–Skolem Theorem, **ZFC** also has enumerable models—models that contain "non-enumerable" sets but which themselves are enumerable.

Chapter 24

Beyond First-order Logic

This chapter, adapted from Jeremy Avigad's logic notes, gives the briefest of glimpses into which other logical systems there are. It is intended as a chapter suggesting further topics for study in a course that does

not cover them. Each one of the topics mentioned here will—hopefully—eventually receive its own part-level treatment in the Open Logic Project.

---

content/first-order-logic/beyond/introduction.tex

24.1 Overview

fol:byd:int:sec

First-order logic is not the only system of logic of interest: there are many extensions and variations of first-order logic. A logic typically consists of the formal specification of a language, usually, but not always, a deductive system, and usually, but not always, an intended semantics. But the technical use of the term raises an obvious question: what do logics that are not first-order logic have to do with the word "logic," used in the intuitive or philosophical sense? All of the systems described below are designed to model reasoning of some form or another; can we say what makes them logical?

No easy answers are forthcoming. The word "logic" is used in different ways and in different contexts, and the notion, like that of "truth," has been analyzed from numerous philosophical stances. For example, one might take the goal of logical reasoning to be the determination of which statements are necessarily true, true a priori, true independent of the interpretation of the nonlogical terms, true by virtue of their form, or true by linguistic convention; and each of these conceptions requires a good deal of clarification. Even if one restricts one's attention to the kind of logic used in mathematics, there is little agreement as to its scope. For example, in the *Principia Mathematica*, Russell and Whitehead tried to develop mathematics on the basis of logic, in the *logicist* tradition begun by Frege. Their system of logic was a form of higher-type logic similar to the one described below. In the end they were forced to introduce axioms which, by most standards, do not seem purely logical (notably, the axiom of infinity, and the axiom of reducibility), but one might nonetheless hold that some forms of higher-order reasoning should be accepted as logical. In contrast, Quine, whose ontology does not admit "propositions" as legitimate objects of discourse, argues that second-order and higher-order logic are really manifestations of set theory in sheep's clothing; in other words, systems involving quantification over predicates are not purely logical.

For now, it is best to leave such philosophical issues for a rainy day, and simply think of the systems below as formal idealizations of various kinds of reasoning, logical or otherwise.

---

content/first-order-logic/beyond/many-sorted-logic.tex

24.2 Many-Sorted Logic

fol:byd:msl:sec

In first-order logic, variables and quantifiers range over a single domain. But it is often useful to have multiple (disjoint) domains: for example, you might want to have a domain of numbers, a domain of geometric objects, a domain of functions from numbers to numbers, a domain of abelian groups, and so on.

Many-sorted logic provides this kind of framework. One starts with a list of "sorts"—the "sort" of an object indicates the "domain" it is supposed to inhabit. One then has variables and quantifiers for each sort, and (usually) an identity predicate for each sort. Functions and relations are also "typed" by the sorts of objects they can take as arguments. Otherwise, one keeps the usual rules of first-order logic, with versions of the quantifier-rules repeated for each sort.

For example, to study international relations we might choose a language with two sorts of objects, French citizens and German citizens. We might have a unary relation, "drinks wine," for objects of the first sort; another unary relation, "eats wurst," for objects of the second sort; and a binary relation, "forms a multinational married couple," which takes two arguments, where the first argument is of the first sort and the second argument is of the second sort. If we use variables a, b, c to range over French citizens and x, y, z to range over German citizens, then

$$\forall a \forall x[(\mathit{MarriedTo}(a, x) \to (\mathit{DrinksWine}(a) \vee \neg \mathit{EatsWurst}(x))]]$$

asserts that if any French person is married to a German, either the French person drinks wine or the German doesn't eat wurst.

Many-sorted logic can be embedded in first-order logic in a natural way, by lumping all the objects of the many-sorted domains together into one first-order domain, using unary predicate symbols to keep track of the sorts, and relativizing quantifiers. For example, the first-order language corresponding to the example above would have unary predicate symbols "German" and "French," in addition to the other relations described, with the sort requirements erased. A sorted quantifier $\forall x\, \varphi$, where x is a variable of the German sort, translates to

$$\forall x\, (\mathit{German}(x) \to \varphi).$$

We need to add axioms that insure that the sorts are separate—e.g., $\forall x\, \neg(\mathit{German}(x) \wedge \mathit{French}(x))$—as well as axioms that guarantee that "drinks wine" only holds of objects satisfying the predicate $\mathit{French}(x)$, etc. With these conventions and axioms, it is not difficult to show that many-sorted sentences translate to first-order sentences, and many-sorted derivations translate to first-order derivations. Also, many-sorted structures "translate" to corresponding first-order structures and vice-versa, so we also have a completeness theorem for many-sorted logic.

---

content/first-order-logic/beyond/second-order-logic.tex

24.3 Second-Order logic

fol:byd:sol:sec

The language of second-order logic allows one to quantify not just over a domain of individuals, but over relations on that domain as well. Given a first-order language $\mathcal{L}$, for each k one adds variables R which range over k-ary relations, and allows quantification over those variables. If R is a variable for a k-ary relation, and t_1, ..., t_k are ordinary (first-order) terms, $R(t_1, \ldots, t_k)$ is an atomic formula. Otherwise, the set of formulas is defined just as in the case of first-order logic, with additional clauses for second-order quantification. Note that we only have the identity predicate for first-order terms: if R and S are relation variables of the same arity k, we can define $R = S$ to be an abbreviation for

$$\forall x_1 \ldots \forall x_k \, (R(x_1, \ldots, x_k) \leftrightarrow S(x_1, \ldots, x_k)).$$

The rules for second-order logic simply extend the quantifier rules to the new second order variables. Here, however, one has to be a little bit careful to explain how these variables interact with the predicate symbols of $\mathcal{L}$, and with formulas of $\mathcal{L}$ more generally. At the bare minimum, relation variables count as terms, so one has inferences of the form

$$\varphi(R) \vdash \exists R \, \varphi(R)$$

But if $\mathcal{L}$ is the language of arithmetic with a constant relation symbol $<$, one would also expect the following inference to be valid:

$$x < y \vdash \exists R \, R(x, y)$$

or for a given formula φ,

$$\varphi(x_1, \ldots, x_k) \vdash \exists R \, R(x_1, \ldots, x_k)$$

More generally, we might want to allow inferences of the form

$$\varphi[\lambda \vec{x}. \, \psi(\vec{x})/R] \vdash \exists R \, \varphi$$

where $\varphi[\lambda \vec{x}. \, \psi(\vec{x})/R]$ denotes the result of replacing every atomic formula of the form $Rt_1, \ldots, t_k$ in φ by $\psi(t_1, \ldots, t_k)$. This last rule is equivalent to having a *comprehension schema*, i.e., an axiom of the form

$$\exists R \, \forall x_1, \ldots, x_k \, (\varphi(x_1, \ldots, x_k) \leftrightarrow R(x_1, \ldots, x_k)),$$

one for each formula φ in the second-order language, in which R is not a free variable. (Exercise: show that if R is allowed to occur in φ, this schema is inconsistent!)

When logicians refer to the "axioms of second-order logic" they usually mean the minimal extension of first-order logic by second-order quantifier rules together with the comprehension schema. But it is often interesting to study weaker subsystems of these axioms and rules. For example, note that in its full generality the axiom schema of comprehension is *impredicative*: it allows one to

assert the existence of a relation $R(x_1, \ldots, x_k)$ that is "defined" by a formula with second-order quantifiers; and these quantifiers range over the set of all such relations—a set which includes R itself! Around the turn of the twentieth century, a common reaction to Russell's paradox was to lay the blame on such definitions, and to avoid them in developing the foundations of mathematics. If one prohibits the use of second-order quantifiers in the formula φ, one has a *predicative* form of comprehension, which is somewhat weaker.

From the semantic point of view, one can think of a second-order structure as consisting of a first-order structure for the language, coupled with a set of relations on the domain over which the second-order quantifiers range (more precisely, for each k there is a set of relations of arity k). Of course, if comprehension is included in the derivation system, then we have the added requirement that there are enough relations in the "second-order part" to satisfy the comprehension axioms—otherwise the derivation system is not sound! One easy way to ensure that there are enough relations around is to take the second-order part to consist of *all* the relations on the first-order part. Such a structure is called *full*, and, in a sense, is really the "intended structure" for the language. If we restrict our attention to full structures we have what is known as the *full* second-order semantics. In that case, specifying a structure boils down to specifying the first-order part, since the contents of the second-order part follow from that implicitly.

To summarize, there is some ambiguity when talking about second-order logic. In terms of the derivation system, one might have in mind either

1. A "minimal" second-order derivation system, together with some comprehension axioms.

2. The "standard" second-order derivation system, with full comprehension.

In terms of the semantics, one might be interested in either

1. The "weak" semantics, where a structure consists of a first-order part, together with a second-order part big enough to satisfy the comprehension axioms.

2. The "standard" second-order semantics, in which one considers full structures only.

When logicians do not specify the derivation system or the semantics they have in mind, they are usually referring to the second item on each list. The advantage to using this semantics is that, as we will see, it gives us categorical descriptions of many natural mathematical structures; at the same time, the derivation system is quite strong, and sound for this semantics. The drawback is that the derivation system is *not* complete for the semantics; in fact, *no* effectively given derivation system is complete for the full second-order semantics. On the other hand, we will see that the derivation system *is* complete for the weakened semantics; this implies that if a sentence is not provable, then there is *some* structure, not necessarily the full one, in which it is false.

The language of second-order logic is quite rich. One can identify unary relations with subsets of the domain, and so in particular you can quantify over these sets; for example, one can express induction for the natural numbers with a single axiom

$$\forall R\,((R(\mathsf{o}) \wedge \forall x\,(R(x) \to R(x'))) \to \forall x\,R(x)).$$

If one takes the language of arithmetic to have symbols $\mathsf{o}, \prime, +, \times$ and $<$, one can add the following axioms to describe their behavior:

1. $\forall x\,\neg x' = \mathsf{o}$
2. $\forall x\,\forall y\,(s(x) = s(y) \to x = y)$
3. $\forall x\,(x + \mathsf{o}) = x$
4. $\forall x\,\forall y\,(x + y') = (x + y)'$
5. $\forall x\,(x \times \mathsf{o}) = \mathsf{o}$
6. $\forall x\,\forall y\,(x \times y') = ((x \times y) + x)$
7. $\forall x\,\forall y\,(x < y \leftrightarrow \exists z\,y = (x + z'))$

It is not difficult to show that these axioms, together with the axiom of induction above, provide a categorical description of the structure $\mathfrak{N}$, the standard model of arithmetic, provided we are using the full second-order semantics. Given any structure $\mathfrak{M}$ in which these axioms are true, define a function f from $\mathbb{N}$ to the domain of $\mathfrak{M}$ using ordinary recursion on $\mathbb{N}$, so that $f(0) = \mathsf{o}^{\mathfrak{M}}$ and $f(x + 1) = \prime^{\mathfrak{M}}(f(x))$. Using ordinary induction on $\mathbb{N}$ and the fact that axioms (1) and (2) hold in $\mathfrak{M}$, we see that f is injective. To see that f is surjective, let P be the set of elements of $|\mathfrak{M}|$ that are in the range of f. Since $\mathfrak{M}$ is full, P is in the second-order domain. By the construction of f, we know that $\mathsf{o}^{\mathfrak{M}}$ is in P, and that P is closed under $\prime^{\mathfrak{M}}$. The fact that the induction axiom holds in $\mathfrak{M}$ (in particular, for P) guarantees that P is equal to the entire first-order domain of $\mathfrak{M}$. This shows that f is a bijection. Showing that f is a homomorphism is no more difficult, using ordinary induction on $\mathbb{N}$ repeatedly.

In set-theoretic terms, a function is just a special kind of relation; for example, a unary function f can be identified with a binary relation R satisfying $\forall x\,\exists! y\,R(x, y)$. As a result, one can quantify over functions too. Using the full semantics, one can then define the class of infinite structures to be the class of structures $\mathfrak{M}$ for which there is an injective function from the domain of $\mathfrak{M}$ to a proper subset of itself:

$$\exists f\,(\forall x\,\forall y\,(f(x) = f(y) \to x = y) \wedge \exists y\,\forall x\,f(x) \neq y).$$

The negation of this sentence then defines the class of finite structures.

In addition, one can define the class of well-orderings, by adding the following to the definition of a linear ordering:

$$\forall P\,(\exists x\,P(x) \to \exists x\,(P(x) \wedge \forall y\,(y < x \to \neg P(y)))).$$

This asserts that every non-empty set has a least element, modulo the identification of "set" with "one-place relation". For another example, one can express the notion of connectedness for graphs, by saying that there is no nontrivial separation of the vertices into disconnected parts:

$$\neg\exists A\,(\exists x\,A(x) \wedge \exists y\,\neg A(y) \wedge \forall w\,\forall z\,((A(w) \wedge \neg A(z)) \rightarrow \neg R(w,z))).$$

For yet another example, you might try as an exercise to define the class of finite structures whose domain has even size. More strikingly, one can provide a categorical description of the real numbers as a complete ordered field containing the rationals.

In short, second-order logic is much more expressive than first-order logic. That's the good news; now for the bad. We have already mentioned that there is no effective derivation system that is complete for the full second-order semantics. For better or for worse, many of the properties of first-order logic are absent, including compactness and the Löwenheim–Skolem theorems.

On the other hand, if one is willing to give up the full second-order semantics in terms of the weaker one, then the minimal second-order derivation system is complete for this semantics. In other words, if we read $\vdash$ as "proves in the minimal system" and $\models$ as "logically implies in the weaker semantics", we can show that whenever $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$. If one wants to include specific comprehension axioms in the derivation system, one has to restrict the semantics to second-order structures that satisfy these axioms: for example, if Δ consists of a set of comprehension axioms (possibly all of them), we have that if $\Gamma \cup \Delta \models \varphi$, then $\Gamma \cup \Delta \vdash \varphi$. In particular, if φ is not provable using the comprehension axioms we are considering, then there is a model of $\neg\varphi$ in which these comprehension axioms nonetheless hold.

The easiest way to see that the completeness theorem holds for the weaker semantics is to think of second-order logic as a many-sorted logic, as follows. One sort is interpreted as the ordinary "first-order" domain, and then for each k we have a domain of "relations of arity k." We take the language to have built-in relation symbols "$\mathit{true}_k(R, x_1, \ldots, x_k)$" which is meant to assert that R holds of $x_1, \ldots, x_k$, where R is a variable of the sort "k-ary relation" and $x_1, \ldots, x_k$ are objects of the first-order sort.

With this identification, the weak second-order semantics is essentially the usual semantics for many-sorted logic; and we have already observed that many-sorted logic can be embedded in first-order logic. Modulo the translations back and forth, then, the weaker conception of second-order logic is really a form of first-order logic in disguise, where the domain contains both "objects" and "relations" governed by the appropriate axioms.

---

content/first-order-logic/beyond/higher-order-logic.tex

24.4 Higher-Order logic

fol:byd:hol:sec

Passing from first-order logic to second-order logic enabled us to talk about sets of objects in the first-order domain, within the formal language. Why stop there? For example, third-order logic should enable us to deal with sets of sets of objects, or perhaps even sets which contain both objects and sets of objects. And fourth-order logic will let us talk about sets of objects of that kind. As you may have guessed, one can iterate this idea arbitrarily.

In practice, higher-order logic is often formulated in terms of functions instead of relations. (Modulo the natural identifications, this difference is inessential.) Given some basic "sorts" A, B, C, ... (which we will now call "types"), we can create new ones by stipulating

> If σ and τ are finite types then so is $\sigma \to \tau$.

Think of types as syntactic "labels," which classify the objects we want in our domain; $\sigma \to \tau$ describes those objects that are functions which take objects of type σ to objects of type τ. For example, we might want to have a type Ω of truth values, "true" and "false," and a type $\mathbb{N}$ of natural numbers. In that case, you can think of objects of type $\mathbb{N} \to \Omega$ as unary relations, or subsets of $\mathbb{N}$; objects of type $\mathbb{N} \to \mathbb{N}$ are functions from natural numbers to natural numbers; and objects of type $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ are "functionals," that is, higher-type functions that take functions to numbers.

As in the case of second-order logic, one can think of higher-order logic as a kind of many-sorted logic, where there is a sort for each type of object we want to consider. But it is usually clearer just to define the syntax of higher-type logic from the ground up. For example, we can define a set of finite types inductively, as follows:

1. $\mathbb{N}$ is a finite type.
2. If σ and τ are finite types, then so is $\sigma \to \tau$.
3. If σ and τ are finite types, so is $\sigma \times \tau$.

Intuitively, $\mathbb{N}$ denotes the type of the natural numbers, $\sigma \to \tau$ denotes the type of functions from σ to τ, and $\sigma \times \tau$ denotes the type of pairs of objects, one from σ and one from τ. We can then define a set of terms inductively, as follows:

1. For each type σ, there is a stock of variables x, y, z, ... of type σ
2. $\mathsf{0}$ is a term of type $\mathbb{N}$
3. S (successor) is a term of type $\mathbb{N} \to \mathbb{N}$
4. If s is a term of type σ, and t is a term of type $\mathbb{N} \to (\sigma \to \sigma)$, then R_{st} is a term of type $\mathbb{N} \to \sigma$
5. If s is a term of type $\tau \to \sigma$ and t is a term of type τ, then $s(t)$ is a term of type σ

6. If s is a term of type σ and x is a variable of type τ, then $\lambda x.\, s$ is a term of type $\tau \to \sigma$.

7. If s is a term of type σ and t is a term of type τ, then $\langle s, t \rangle$ is a term of type $\sigma \times \tau$.

8. If s is a term of type $\sigma \times \tau$ then $p_1(s)$ is a term of type σ and $p_2(s)$ is a term of type τ.

Intuitively, R_{st} denotes the function defined recursively by

$$R_{st}(0) = s$$
$$R_{st}(x+1) = t(x, R_{st}(x)),$$

$\langle s, t \rangle$ denotes the pair whose first component is s and whose second component is t, and $p_1(s)$ and $p_2(s)$ denote the first and second elements ("projections") of s. Finally, $\lambda x.\, s$ denotes the function f defined by

$$f(x) = s$$

for any x of type σ; so item (6) gives us a form of comprehension, enabling us to define functions using terms. Formulas are built up from identity predicate statements $s = t$ between terms of the same type, the usual propositional connectives, and higher-type quantification. One can then take the axioms of the system to be the basic equations governing the terms defined above, together with the usual rules of logic with quantifiers and identity predicate.

If one augments the finite type system with a type Ω of truth values, one has to include axioms which govern its use as well. In fact, if one is clever, one can get rid of complex formulas entirely, replacing them with terms of type Ω! The proof system can then be modified accordingly. The result is essentially the *simple theory of types* set forth by Alonzo Church in the 1930s.

As in the case of second-order logic, there are different versions of higher-type semantics that one might want to use. In the full version, variables of type $\sigma \to \tau$ range over the set of *all* functions from the objects of type σ to objects of type τ. As you might expect, this semantics is too strong to admit a complete, effective derivation system. But one can consider a weaker semantics, in which a structure consists of sets of elements T_τ for each type τ, together with appropriate operations for application, projection, etc. If the details are carried out correctly, one can obtain completeness theorems for the kinds of derivation systems described above.

Higher-type logic is attractive because it provides a framework in which we can embed a good deal of mathematics in a natural way: starting with $\mathbb{N}$, one can define real numbers, continuous functions, and so on. It is also particularly attractive in the context of intuitionistic logic, since the types have clear "constructive" interpretations. In fact, one can develop constructive versions of higher-type semantics (based on intuitionistic, rather than classical logic) that clarify these constructive interpretations quite nicely, and are, in many ways, more interesting than the classical counterparts.

content/first-order-logic/beyond/intuitionistic-logic.tex

24.5 Intuitionistic Logic

fol:byd:il:sec

In contrast to second-order and higher-order logic, intuitionistic first-order logic represents a restriction of the classical version, intended to model a more "constructive" kind of reasoning. The following examples may serve to illustrate some of the underlying motivations.

Suppose someone came up to you one day and announced that they had determined a natural number x, with the property that if x is prime, the Riemann hypothesis is true, and if x is composite, the Riemann hypothesis is false. Great news! Whether the Riemann hypothesis is true or not is one of the big open questions of mathematics, and here they seem to have reduced the problem to one of calculation, that is, to the determination of whether a specific number is prime or not.

What is the magic value of x? They describe it as follows: x is the natural number that is equal to 7 if the Riemann hypothesis is true, and 9 otherwise.

Angrily, you demand your money back. From a classical point of view, the description above does in fact determine a unique value of x; but what you really want is a value of x that is given *explicitly*.

To take another, perhaps less contrived example, consider the following question. We know that it is possible to raise an irrational number to a rational power, and get a rational result. For example, $\sqrt{2}^2 = 2$. What is less clear is whether or not it is possible to raise an irrational number to an *irrational* power, and get a rational result. The following theorem answers this in the affirmative:

Theorem 24.1. *There are irrational numbers a and b such that a^b is rational.*

Proof. Consider $\sqrt{2}^{\sqrt{2}}$. If this is rational, we are done: we can let $a = b = \sqrt{2}$. Otherwise, it is irrational. Then we have

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\cdot\sqrt{2}} = \sqrt{2}^2 = 2,$$

which is certainly rational. So, in this case, let a be $\sqrt{2}^{\sqrt{2}}$, and let b be $\sqrt{2}$. □

Does this constitute a valid proof? Most mathematicians feel that it does. But again, there is something a little bit unsatisfying here: we have proved the existence of a pair of real numbers with a certain property, without being able to say *which* pair of numbers it is. It is possible to prove the same result, but in such a way that the pair a, b *is* given in the proof: take $a = \sqrt{3}$ and $b = \log_3 4$. Then

$$a^b = \sqrt{3}^{\log_3 4} = 3^{1/2\cdot\log_3 4} = (3^{\log_3 4})^{1/2} = 4^{1/2} = 2,$$

since $3^{\log_3 x} = x$.

Intuitionistic logic is designed to model a kind of reasoning where moves like the one in the first proof are disallowed. Proving the existence of an x satisfying $\varphi(x)$ means that you have to give a specific x, and a proof that it satisfies φ, like in the second proof. Proving that φ or ψ holds requires that you can prove one or the other.

Formally speaking, intuitionistic first-order logic is what you get if you restrict a derivation system for first-order logic in a certain way. Similarly, there are intuitionistic versions of second-order or higher-order logic. From the mathematical point of view, these are just formal deductive systems, but, as already noted, they are intended to model a kind of mathematical reasoning. One can take this to be the kind of reasoning that is justified on a certain philosophical view of mathematics (such as Brouwer's intuitionism); one can take it to be a kind of mathematical reasoning which is more "concrete" and satisfying (along the lines of Bishop's constructivism); and one can argue about whether or not the formal description captures the informal motivation. But whatever philosophical positions we may hold, we can study intuitionistic logic as a formally presented logic; and for whatever reasons, many mathematical logicians find it interesting to do so.

There is an informal constructive interpretation of the intuitionist connectives, usually known as the BHK interpretation (named after Brouwer, Heyting, and Kolmogorov). It runs as follows: a proof of $\varphi \wedge \psi$ consists of a proof of φ paired with a proof of ψ; a proof of $\varphi \vee \psi$ consists of either a proof of φ, or a proof of ψ, where we have explicit information as to which is the case; a proof of $\varphi \to \psi$ consists of a procedure, which transforms a proof of φ to a proof of ψ; a proof of $\forall x\, \varphi(x)$ consists of a procedure which returns a proof of $\varphi(x)$ for any value of x; and a proof of $\exists x\, \varphi(x)$ consists of a value of x, together with a proof that this value satisfies φ. One can describe the interpretation in computational terms known as the "Curry–Howard isomorphism" or the "formulas-as-types paradigm": think of a formula as specifying a certain kind of data type, and proofs as computational objects of these data types that enable us to see that the corresponding formula is true.

Intuitionistic logic is often thought of as being classical logic "minus" the law of the excluded middle. This following theorem makes this more precise.

Theorem 24.2. *Intuitionistically, the following axiom schemata are equivalent:*

1. $(\neg\varphi \to \bot) \to \varphi$.
2. $\varphi \vee \neg\varphi$
3. $\neg\neg\varphi \to \varphi$

Obtaining instances of one schema from either of the others is a good exercise in intuitionistic logic.

The first deductive systems for intuitionistic propositional logic, put forth as formalizations of Brouwer's intuitionism, are due, independently, to Kolmogorov, Glivenko, and Heyting. The first formalization of intuitionistic first-order logic (and parts of intuitionist mathematics) is due to Heyting. Though a number of classically valid schemata are not intuitionistically valid, many are.

The *double-negation translation* describes an important relationship between classical and intuitionist logic. It is defined inductively follows (think of φ^N as the "intuitionist" translation of the classical formula φ):

$$\begin{aligned}
\varphi^N &\equiv \neg\neg\varphi \quad \text{for atomic formulas } \varphi \\
(\varphi \wedge \psi)^N &\equiv (\varphi^N \wedge \psi^N) \\
(\varphi \vee \psi)^N &\equiv \neg\neg(\varphi^N \vee \psi^N) \\
(\varphi \to \psi)^N &\equiv (\varphi^N \to \psi^N) \\
(\forall x\, \varphi)^N &\equiv \forall x\, \varphi^N \\
(\exists x\, \varphi)^N &\equiv \neg\neg\exists x\, \varphi^N
\end{aligned}$$

Kolmogorov and Glivenko had versions of this translation for propositional logic; for predicate logic, it is due to Gödel and Gentzen, independently. We have

Theorem 24.3. *1. $\varphi \leftrightarrow \varphi^N$ is provable classically*

2. If φ is provable classically, then φ^N is provable intuitionistically.

We can now envision the following dialogue. Classical mathematician: "I've proved φ!" Intuitionist mathematician: "Your proof isn't valid. What you've really proved is φ^N." Classical mathematician: "Fine by me!" As far as the classical mathematician is concerned, the intuitionist is just splitting hairs, since the two are equivalent. But the intuitionist insists there is a difference.

Note that the above translation concerns pure logic only; it does not address the question as to what the appropriate *nonlogical* axioms are for classical and intuitionistic mathematics, or what the relationship is between them. But the following slight extension of the theorem above provides some useful information:

Theorem 24.4. *If Γ proves φ classically, Γ^N proves φ^N intuitionistically.*

In other words, if φ is provable from some hypotheses classically, then φ^N is provable from their double-negation translations.

To show that a sentence or propositional formula is intuitionistically valid, all you have to do is provide a proof. But how can you show that it is not valid? For that purpose, we need a semantics that is sound, and preferably complete. A semantics due to Kripke nicely fits the bill.

We can play the same game we did for classical logic: define the semantics, and prove soundness and completeness. It is worthwhile, however, to note

the following distinction. In the case of classical logic, the semantics was the "obvious" one, in a sense implicit in the meaning of the connectives. Though one can provide some intuitive motivation for Kripke semantics, the latter does not offer the same feeling of inevitability. In addition, the notion of a classical structure is a natural mathematical one, so we can either take the notion of a structure to be a tool for studying classical first-order logic, or take classical first-order logic to be a tool for studying mathematical structures. In contrast, Kripke structures can only be viewed as a logical construct; they don't seem to have independent mathematical interest.

A Kripke structure $\mathfrak{M} = \langle W, R, V \rangle$ for a propositional language consists of a set W, partial order R on W with a least element, and an "monotone" assignment of propositional variables to the elements of W. The intuition is that the elements of W represent "worlds," or "states of knowledge"; an element $v \geq u$ represents a "possible future state" of u; and the propositional variables assigned to u are the propositions that are known to be true in state u. The forcing relation $\mathfrak{M}, w \Vdash \varphi$ then extends this relationship to arbitrary formulas in the language; read $\mathfrak{M}, w \Vdash \varphi$ as "φ is true in state w." The relationship is defined inductively, as follows:

1. $\mathfrak{M}, w \Vdash p_i$ iff p_i is one of the propositional variables assigned to w.
2. $\mathfrak{M}, w \nVdash \bot$.
3. $\mathfrak{M}, w \Vdash (\varphi \wedge \psi)$ iff $\mathfrak{M}, w \Vdash \varphi$ and $\mathfrak{M}, w \Vdash \psi$.
4. $\mathfrak{M}, w \Vdash (\varphi \vee \psi)$ iff $\mathfrak{M}, w \Vdash \varphi$ or $\mathfrak{M}, w \Vdash \psi$.
5. $\mathfrak{M}, w \Vdash (\varphi \to \psi)$ iff, whenever $w' \geq w$ and $\mathfrak{M}, w' \Vdash \varphi$, then $\mathfrak{M}, w' \Vdash \psi$.

It is a good exercise to try to show that $\neg(p \wedge q) \to (\neg p \vee \neg q)$ is not intuitionistically valid, by cooking up a Kripke structure that provides a counterexample.

---

content/first-order-logic/beyond/modal-logics.tex

24.6 Modal Logics

fol:byd:mod:sec

Consider the following example of a conditional sentence:

> If Jeremy is alone in that room, then he is drunk and naked and dancing on the chairs.

This is an example of a conditional assertion that may be materially true but nonetheless misleading, since it seems to suggest that there is a stronger link between the antecedent and conclusion other than simply that either the antecedent is false or the consequent true. That is, the wording suggests that the claim is not only true in this particular world (where it may be trivially true, because Jeremy is not alone in the room), but that, moreover, the conclusion *would have* been true *had* the antecedent been true. In other words, one can

take the assertion to mean that the claim is true not just in this world, but in any "possible" world; or that it is *necessarily* true, as opposed to just true in this particular world.

Modal logic was designed to make sense of this kind of necessity. One obtains modal propositional logic from ordinary propositional logic by adding a box operator; which is to say, if φ is a formula, so is $\Box\varphi$. Intuitively, $\Box\varphi$ asserts that φ is *necessarily* true, or true in any possible world. $\Diamond\varphi$ is usually taken to be an abbreviation for $\neg\Box\neg\varphi$, and can be read as asserting that φ is *possibly* true. Of course, modality can be added to predicate logic as well.

Kripke structures can be used to provide a semantics for modal logic; in fact, Kripke first designed this semantics with modal logic in mind. Rather than restricting to partial orders, more generally one has a set of "possible worlds," P, and a binary "accessibility" relation $R(x, y)$ between worlds. Intuitively, $R(p, q)$ asserts that the world q is compatible with p; i.e., if we are "in" world p, we have to entertain the possibility that the world could have been like q.

Modal logic is sometimes called an "intensional" logic, as opposed to an "extensional" one. The intended semantics for an extensional logic, like classical logic, will only refer to a single world, the "actual" one; while the semantics for an "intensional" logic relies on a more elaborate ontology. In addition to structureing necessity, one can use modality to structure other linguistic constructions, reinterpreting $\Box$ and $\Diamond$ according to the application. For example:

1. In provability logic, $\Box\varphi$ is read "φ is provable" and $\Diamond\varphi$ is read "φ is consistent."

2. In epistemic logic, one might read $\Box\varphi$ as "I know φ" or "I believe φ."

3. In temporal logic, one can read $\Box\varphi$ as "φ is always true" and $\Diamond\varphi$ as "φ is sometimes true."

One would like to augment logic with rules and axioms dealing with modality. For example, the system **S4** consists of the ordinary axioms and rules of propositional logic, together with the following axioms:

$$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$$
$$\Box\varphi \to \varphi$$
$$\Box\varphi \to \Box\Box\varphi$$

as well as a rule, "from φ conclude $\Box\varphi$." **S5** adds the following axiom:

$$\Diamond\varphi \to \Box\Diamond\varphi$$

Variations of these axioms may be suitable for different applications; for example, S5 is usually taken to characterize the notion of logical necessity. And the nice thing is that one can usually find a semantics for which the derivation system is sound and complete by restricting the accessibility relation in the Kripke structures in natural ways. For example, **S4** corresponds to the class

of Kripke structures in which the accessibility relation is reflexive and transitive. **S5** corresponds to the class of Kripke structures in which the accessibility relation is *universal*, which is to say that every world is accessible from every other; so $\Box\varphi$ holds if and only if φ holds in every world.

---

content/first-order-logic/beyond/other-logics.tex

24.7 Other Logics

fol:byd:oth:sec

As you may have gathered by now, it is not hard to design a new logic. You too can create your own a syntax, make up a deductive system, and fashion a semantics to go with it. You might have to be a bit clever if you want the derivation system to be complete for the semantics, and it might take some effort to convince the world at large that your logic is truly interesting. But, in return, you can enjoy hours of good, clean fun, exploring your logic's mathematical and computational properties.

Recent decades have witnessed a veritable explosion of formal logics. Fuzzy logic is designed to model reasoning about vague properties. Probabilistic logic is designed to model reasoning about uncertainty. Default logics and nonmonotonic logics are designed to model defeasible forms of reasoning, which is to say, "reasonable" inferences that can later be overturned in the face of new information. There are epistemic logics, designed to model reasoning about knowledge; causal logics, designed to model reasoning about causal relationships; and even "deontic" logics, which are designed to model reasoning about moral and ethical obligations. Depending on whether the primary motivation for introducing these systems is philosophical, mathematical, or computational, you may find such creatures studies under the rubric of mathematical logic, philosophical logic, artificial intelligence, cognitive science, or elsewhere.

The list goes on and on, and the possibilities seem endless. We may never attain Leibniz' dream of reducing all of human reason to calculation—but that can't stop us from trying.

Part IV

Model Theory

Material on model theory is incomplete and experimental. It is currently simply an adaptation of Aldo Antonelli's notes on model theory, less those topics covered in the part on first-order logic (theories, completeness, compactness). It requires much more introduction, motivation, and explanation, as well as exercises, to be useful for a textbook. Andy Arana is at planning to work on this part specifically (issue #65).

Chapter 25

Basics of Model Theory

content/model-theory/basics/reducts-and-expansions.tex

25.1 Reducts and Expansions

Often it is useful or necessary to compare languages which have symbols in common, as well as structures for these languages. The most common case is when all the symbols in a language $\mathcal{L}$ are also part of a language $\mathcal{L}'$, i.e., $\mathcal{L} \subseteq \mathcal{L}'$. An $\mathcal{L}$-structure $\mathfrak{M}$ can then always be expanded to an $\mathcal{L}'$-structure by adding interpretations of the additional symbols while leaving the interpretations of the common symbols the same. On the other hand, from an $\mathcal{L}'$-structure $\mathfrak{M}'$ we can obtain an $\mathcal{L}$-structure simply by "forgetting" the interpretations of the symbols that do not occur in $\mathcal{L}$.

mod:bas:red: defn:reduct

Definition 25.1. Suppose $\mathcal{L} \subseteq \mathcal{L}'$, $\mathfrak{M}$ is an $\mathcal{L}$-structure and $\mathfrak{M}'$ is an $\mathcal{L}'$-structure. $\mathfrak{M}$ is the *reduct* of $\mathfrak{M}'$ to $\mathcal{L}$, and $\mathfrak{M}'$ is an *expansion* of $\mathfrak{M}$ to $\mathcal{L}'$ iff

1. $|\mathfrak{M}| = |\mathfrak{M}'|$

2. For every constant symbol $c \in \mathcal{L}$, $c^{\mathfrak{M}} = c^{\mathfrak{M}'}$.
3. For every function symbol $f \in \mathcal{L}$, $f^{\mathfrak{M}} = f^{\mathfrak{M}'}$.
4. For every predicate symbol $P \in \mathcal{L}$, $P^{\mathfrak{M}} = P^{\mathfrak{M}'}$.

Proposition 25.2. *If an $\mathcal{L}$-structure $\mathfrak{M}$ is a reduct of an $\mathcal{L}'$-structure $\mathfrak{M}'$, then for all $\mathcal{L}$-sentences φ,* mod:bas:red: prop:reduct

$$\mathfrak{M} \vDash \varphi \text{ iff } \mathfrak{M}' \vDash \varphi.$$

Proof. Exercise. □

Problem 25.1. Prove Proposition 25.2.

Definition 25.3. When we have an $\mathcal{L}$-structure $\mathfrak{M}$, and $\mathcal{L}' = \mathcal{L} \cup \{P\}$ is the expansion of $\mathcal{L}$ obtained by adding a single n-place predicate symbol P, and $R \subseteq |\mathfrak{M}|^n$ is an n-place relation, then we write $(\mathfrak{M}, R)$ for the expansion $\mathfrak{M}'$ of $\mathfrak{M}$ with $P^{\mathfrak{M}'} = R$.

---

content/model-theory/basics/substructures.tex

25.2 Substructures

The domain of a structure $\mathfrak{M}$ may be a subset of another $\mathfrak{M}'$. But we should obviously only consider $\mathfrak{M}$ a "part" of $\mathfrak{M}'$ if not only $|\mathfrak{M}| \subseteq |\mathfrak{M}'|$, but $\mathfrak{M}$ and $\mathfrak{M}'$ "agree" in how they interpret the symbols of the language at least on the shared part $|\mathfrak{M}|$. mod:bas:sub: sec

Definition 25.4. Given structures $\mathfrak{M}$ and $\mathfrak{M}'$ for the same language $\mathcal{L}$, we say that $\mathfrak{M}$ is a *substructure* of $\mathfrak{M}'$, and $\mathfrak{M}'$ an *extension* of $\mathfrak{M}$, written $\mathfrak{M} \subseteq \mathfrak{M}'$, iff mod:bas:sub: defn:substructure

1. $|\mathfrak{M}| \subseteq |\mathfrak{M}'|$,
2. For each constant $c \in \mathcal{L}$, $c^{\mathfrak{M}} = c^{\mathfrak{M}'}$;
3. For each n-place function symbol $f \in \mathcal{L}$ $f^{\mathfrak{M}}(a_1, \ldots, a_n) = f^{\mathfrak{M}'}(a_1, \ldots, a_n)$ for all $a_1, \ldots, a_n \in |\mathfrak{M}|$.
4. For each n-place predicate symbol $R \in \mathcal{L}$, $\langle a_1, \ldots, a_n \rangle \in R^{\mathfrak{M}}$ iff $\langle a_1, \ldots, a_n \rangle \in R^{\mathfrak{M}'}$ for all $a_1, \ldots, a_n \in |\mathfrak{M}|$.

Remark 1. If the language contains no constant or function symbols, then any $N \subseteq |\mathfrak{M}|$ determines a substructure $\mathfrak{N}$ of $\mathfrak{M}$ with domain $|\mathfrak{N}| = N$ by putting $R^{\mathfrak{N}} = R^{\mathfrak{M}} \cap N^n$. mod:bas:sub: rem:substructure

---

content/model-theory/basics/overspill.tex

25.3 Overspill

mod:bas:ove:sec
mod:bas:ove:overspill

Theorem 25.5. *If a set Γ of sentences has arbitrarily large finite models, then it has an infinite model.*

Proof. Expand the language of Γ by adding countably many new constants c_0, c_1, ... and consider the set $\Gamma \cup \{c_i \neq c_j : i \neq j\}$. To say that Γ has arbitrarily large finite models means that for every $m > 0$ there is $n \geq m$ such that Γ has a model of cardinality n. This implies that $\Gamma \cup \{c_i \neq c_j : i \neq j\}$ is finitely satisfiable. By compactness, $\Gamma \cup \{c_i \neq c_j : i \neq j\}$ has a model $\mathfrak{M}$ whose domain must be infinite, since it satisfies all inequalities $c_i \neq c_j$. □

mod:bas:ove:inf-not-fo

Proposition 25.6. *There is no sentence φ of any first-order language that is true in a structure $\mathfrak{M}$ if and only if the domain $|\mathfrak{M}|$ of the structure is infinite.*

Proof. If there were such a φ, its negation $\neg\varphi$ would be true in all and only the finite structures, and it would therefore have arbitrarily large finite models but it would lack an infinite model, contradicting Theorem 25.5. □

---

content/model-theory/basics/isomorphism.tex

25.4 Isomorphic Structures

mod:bas:iso:sec

First-order structures can be alike in one of two ways. One way in which they can be alike is that they make the same sentences true. We call such structures *elementarily equivalent*. But structures can be very different and still make the same sentences true—for instance, one can be enumerable and the other not. This is because there are lots of features of a structure that cannot be expressed in first-order languages, either because the language is not rich enough, or because of fundamental limitations of first-order logic such as the Löwenheim–Skolem theorem. So another, stricter, aspect in which structures can be alike is if they are fundamentally the same, in the sense that they only differ in the objects that make them up, but not in their structural features. A way of making this precise is by the notion of an *isomorphism*.

mod:bas:iso:defn:elem-equiv

Definition 25.7. Given two structures $\mathfrak{M}$ and $\mathfrak{M}'$ for the same language $\mathcal{L}$, we say that $\mathfrak{M}$ is *elementarily equivalent to* $\mathfrak{M}'$, written $\mathfrak{M} \equiv \mathfrak{M}'$, if and only if for every sentence φ of $\mathcal{L}$, $\mathfrak{M} \vDash \varphi$ iff $\mathfrak{M}' \vDash \varphi$.

mod:bas:iso:defn:isomorphism

Definition 25.8. Given two structures $\mathfrak{M}$ and $\mathfrak{M}'$ for the same language $\mathcal{L}$, we say that $\mathfrak{M}$ is *isomorphic to* $\mathfrak{M}'$, written $\mathfrak{M} \simeq \mathfrak{M}'$, if and only if there is a function $h\colon |\mathfrak{M}| \to |\mathfrak{M}'|$ such that:

1. h is injective: if $h(x) = h(y)$ then $x = y$;
2. h is surjective: for every $y \in |\mathfrak{M}'|$ there is $x \in |\mathfrak{M}|$ such that $h(x) = y$;

3. for every constant symbol c: $h(c^{\mathfrak{M}}) = c^{\mathfrak{M}'}$; mod:bas:iso: defn:iso-const

4. for every n-place predicate symbol P: mod:bas:iso: defn:iso-pred

$$\langle a_1, \ldots, a_n \rangle \in P^{\mathfrak{M}} \quad \text{iff} \quad \langle h(a_1), \ldots, h(a_n) \rangle \in P^{\mathfrak{M}'};$$

5. for every n-place function symbol f: mod:bas:iso: defn:iso-func

$$h(f^{\mathfrak{M}}(a_1, \ldots, a_n)) = f^{\mathfrak{M}'}(h(a_1), \ldots, h(a_n)).$$

Theorem 25.9. *If $\mathfrak{M} \simeq \mathfrak{M}'$ then $\mathfrak{M} \equiv \mathfrak{M}'$.* *mod:bas:iso: thm:isom*

Proof. Let h be an isomorphism of $\mathfrak{M}$ onto $\mathfrak{M}'$. For any assignment s, $h \circ s$ is the composition of h and s, i.e., the assignment in $\mathfrak{M}'$ such that $(h \circ s)(x) = h(s(x))$. By induction on t and φ one can prove the stronger claims:

a. $h(\mathrm{Val}^{\mathfrak{M}}_s(t)) = \mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(t)$.

b. $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}', h \circ s \vDash \varphi$.

The first is proved by induction on the complexity of t.

1. If $t \equiv c$, then $\mathrm{Val}^{\mathfrak{M}}_s(c) = c^{\mathfrak{M}}$ and $\mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(c) = c^{\mathfrak{M}'}$. Thus, $h(\mathrm{Val}^{\mathfrak{M}}_s(t)) = h(c^{\mathfrak{M}}) = c^{\mathfrak{M}'}$ (by (3) of Definition 25.8) $= \mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(t)$.

2. If $t \equiv x$, then $\mathrm{Val}^{\mathfrak{M}}_s(x) = s(x)$ and $\mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(x) = h(s(x))$. Thus, $h(\mathrm{Val}^{\mathfrak{M}}_s(x)) = h(s(x)) = \mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(x)$.

3. If $t \equiv f(t_1, \ldots, t_n)$, then

$$\mathrm{Val}^{\mathfrak{M}}_s(t) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_s(t_n)) \quad \text{and}$$
$$\mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(t) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(t_n)).$$

The induction hypothesis is that for each i, $h(\mathrm{Val}^{\mathfrak{M}}_s(t_i)) = \mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(t_i)$. So,

$$\begin{aligned} h(\mathrm{Val}^{\mathfrak{M}}_s(t)) &= h(f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_s(t_n)) \\ &= f^{\mathfrak{M}'}(h(\mathrm{Val}^{\mathfrak{M}}_s(t_1)), \ldots, h(\mathrm{Val}^{\mathfrak{M}}_s(t_n))) && (25.1) \quad \text{mod:bas:iso: iso-1} \\ &= f^{\mathfrak{M}'}(\mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(t_n)) && (25.2) \quad \text{mod:bas:iso: iso-2} \\ &= \mathrm{Val}^{\mathfrak{M}'}_{h \circ s}(t) \end{aligned}$$

Here, eq. (25.1) follows by (5) of Definition 25.8 and eq. (25.2) by induction hypothesis.

Part (b) is left as an exercise.

If φ is a sentence, the assignments s and $h \circ s$ are irrelevant, and we have $\mathfrak{M} \vDash \varphi$ iff $\mathfrak{M}' \vDash \varphi$. □

Problem 25.2. Carry out the proof of (b) of Theorem 25.9 in detail. Make sure to note where each of the five properties characterizing isomorphisms of Definition 25.8 is used.

Definition 25.10. An *automorphism* of a structure $\mathfrak{M}$ is an isomorphism of $\mathfrak{M}$ onto itself.

Problem 25.3. Show that for any structure $\mathfrak{M}$, if X is a definable subset of $\mathfrak{M}$, and h is an automorphism of $\mathfrak{M}$, then $X = \{h(x) : x \in X\}$ (i.e., X is fixed under h).

---

content/model-theory/basics/theory-of-m.tex

25.5 The Theory of a Structure

Every structure $\mathfrak{M}$ makes some sentences true, and some false. The set of all the sentences it makes true is called its *theory*. That set is in fact a theory, since anything it entails must be true in all its models, including $\mathfrak{M}$.

Definition 25.11. Given a structure $\mathfrak{M}$, the *theory* of $\mathfrak{M}$ is the set $\mathrm{Th}(\mathfrak{M})$ of sentences that are true in $\mathfrak{M}$, i.e., $\mathrm{Th}(\mathfrak{M}) = \{\varphi : \mathfrak{M} \vDash \varphi\}$.

We also use the term "theory" informally to refer to sets of sentences having an intended interpretation, whether deductively closed or not.

Proposition 25.12. *For any $\mathfrak{M}$, $\mathrm{Th}(\mathfrak{M})$ is complete.*

Proof. For any sentence φ either $\mathfrak{M} \vDash \varphi$ or $\mathfrak{M} \vDash \neg\varphi$, so either $\varphi \in \mathrm{Th}(\mathfrak{M})$ or $\neg\varphi \in \mathrm{Th}(\mathfrak{M})$. □

mod:bas:thm: prop:equiv

Proposition 25.13. *If $\mathfrak{N} \models \varphi$ for every $\varphi \in \mathrm{Th}(\mathfrak{M})$, then $\mathfrak{M} \equiv \mathfrak{N}$.*

Proof. Since $\mathfrak{N} \vDash \varphi$ for all $\varphi \in \mathrm{Th}(\mathfrak{M})$, $\mathrm{Th}(\mathfrak{M}) \subseteq \mathrm{Th}(\mathfrak{N})$. If $\mathfrak{N} \vDash \varphi$, then $\mathfrak{N} \nvDash \neg\varphi$, so $\neg\varphi \notin \mathrm{Th}(\mathfrak{M})$. Since $\mathrm{Th}(\mathfrak{M})$ is complete, $\varphi \in \mathrm{Th}(\mathfrak{M})$. So, $\mathrm{Th}(\mathfrak{N}) \subseteq \mathrm{Th}(\mathfrak{M})$, and we have $\mathfrak{M} \equiv \mathfrak{N}$. □

mod:bas:thm: remark:R

Remark 2. Consider $\mathfrak{R} = \langle \mathbb{R}, < \rangle$, the structure whose domain is the set $\mathbb{R}$ of the real numbers, in the language comprising only a 2-place predicate symbol interpreted as the $<$ relation over the reals. Clearly $\mathfrak{R}$ is non-enumerable; however, since $\mathrm{Th}(\mathfrak{R})$ is obviously consistent, by the Löwenheim–Skolem theorem it has an enumerable model, say $\mathfrak{S}$, and by Proposition 25.13, $\mathfrak{R} \equiv \mathfrak{S}$. Moreover, since $\mathfrak{R}$ and $\mathfrak{S}$ are not isomorphic, this shows that the converse of Theorem 25.9 fails in general.

---

content/model-theory/basics/partial-iso.tex

25.6 Partial Isomorphisms

Definition 25.14. Given two structures $\mathfrak{M}$ and $\mathfrak{N}$, a *partial isomorphism* from $\mathfrak{M}$ to $\mathfrak{N}$ is a finite partial function p taking arguments in $|\mathfrak{M}|$ and returning values in $|\mathfrak{N}|$, which satisfies the isomorphism conditions from Definition 25.8 on its domain:

1. p is injective;
2. for every constant symbol c: if $p(c^{\mathfrak{M}})$ is defined, then $p(c^{\mathfrak{M}}) = c^{\mathfrak{N}}$;
3. for every n-place predicate symbol P: if $a_1, \ldots, a_n$ are in the domain of p, then $\langle a_1, \ldots, a_n\rangle \in P^{\mathfrak{M}}$ if and only if $\langle p(a_1), \ldots, p(a_n)\rangle \in P^{\mathfrak{N}}$;
4. for every n-place function symbol f: if $a_1, \ldots, a_n$ are in the domain of p, then $p(f^{\mathfrak{M}}(a_1, \ldots, a_n)) = f^{\mathfrak{N}}(p(a_1), \ldots, p(a_n))$.

That p is finite means that $\mathrm{dom}(p)$ is finite.

Notice that the empty function $\emptyset$ is always a partial isomorphism between any two structures.

Definition 25.15. Two structures $\mathfrak{M}$ and $\mathfrak{N}$, are *partially isomorphic*, written $\mathfrak{M} \simeq_p \mathfrak{N}$, if and only if there is a non-empty set I of partial isomorphisms between $\mathfrak{M}$ and $\mathfrak{N}$ satisfying the *back-and-forth* property: mod:bas:pis: defn:partialisom

1. (*Forth*) For every $p \in I$ and $a \in |\mathfrak{M}|$ there is $q \in I$ such that $p \subseteq q$ and a is in the domain of q;
2. (*Back*) For every $p \in I$ and $b \in |\mathfrak{N}|$ there is $q \in I$ such that $p \subseteq q$ and b is in the range of q.

Theorem 25.16. *If $\mathfrak{M} \simeq_p \mathfrak{N}$ and $\mathfrak{M}$ and $\mathfrak{N}$ are enumerable, then $\mathfrak{M} \simeq \mathfrak{N}$.* mod:bas:pis: thm:p-isom1

Proof. Since $\mathfrak{M}$ and $\mathfrak{N}$ are enumerable, let $|\mathfrak{M}| = \{a_0, a_1, \ldots\}$ and $|\mathfrak{N}| = \{b_0, b_1, \ldots\}$. Starting with an arbitrary $p_0 \in I$, we define an increasing sequence of partial isomorphisms $p_0 \subseteq p_1 \subseteq p_2 \subseteq \cdots$ as follows:

1. if $n+1$ is odd, say $n = 2r$, then using the Forth property find a $p_{n+1} \in I$ such that $p_n \subseteq p_{n+1}$ and a_r is in the domain of p_{n+1};
2. if $n+1$ is even, say $n+1 = 2r$, then using the Back property find a $p_{n+1} \in I$ such that $p_n \subseteq p_{n+1}$ and b_r is in the range of p_{n+1}.

If we now put:

$$p = \bigcup_{n \geq 0} p_n,$$

we have that p is a an isomorphism between $\mathfrak{M}$ and $\mathfrak{N}$. □

Problem 25.4. Show in detail that p as defined in Theorem 25.16 is in fact an isomorphism.

mod:bas:pis: thm:p-isom2

Theorem 25.17. *Suppose $\mathfrak{M}$ and $\mathfrak{N}$ are structures for a purely relational language (a language containing only predicate symbols, and no function symbols or constants). Then if $\mathfrak{M} \simeq_p \mathfrak{N}$, also $\mathfrak{M} \equiv \mathfrak{N}$.*

Proof. By induction on formulas, one shows that if $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ are such that there is a partial isomorphism p mapping each a_i to b_i and $s_1(x_i) = a_i$ and $s_2(x_i) = b_i$ (for $i = 1, \ldots, n$), then $\mathfrak{M}, s_1 \vDash \varphi$ if and only if $\mathfrak{N}, s_2 \vDash \varphi$. The case for $n = 0$ gives $\mathfrak{M} \equiv \mathfrak{N}$. □

Remark 3. If function symbols are present, the previous result is still true, but one needs to consider the isomorphism induced by p between the substructure of $\mathfrak{M}$ generated by $a_1, \ldots, a_n$ and the substructure of $\mathfrak{N}$ generated by $b_1, \ldots, b_n$.

The previous result can be "broken down" into stages by establishing a connection between the number of nested quantifiers in a formula and how many times the relevant partial isomorphisms can be extended.

Definition 25.18. For any formula φ, the *quantifier rank* of φ, denoted by $\mathrm{qr}(\varphi) \in \mathbb{N}$, is recursively defined as the highest number of nested quantifiers in φ. Two structures $\mathfrak{M}$ and $\mathfrak{N}$ are *n-equivalent*, written $\mathfrak{M} \equiv_n \mathfrak{N}$, if they agree on all sentences of quantifier rank less than or equal to n.

mod:bas:pis: prop:qr-finite

Proposition 25.19. *Let $\mathcal{L}$ be a finite purely relational language, i.e., a language containing finitely many predicate symbols and constant symbols, and no function symbols. Then for each $n \in \mathbb{N}$ there are only finitely many first-order sentences in the language $\mathcal{L}$ that have quantifier rank no greater than n, up to logical equivalence.*

Proof. By induction on n. □

Definition 25.20. Given a structure $\mathfrak{M}$, let $|\mathfrak{M}|^{<\omega}$ be the set of all finite sequences over $|\mathfrak{M}|$. We use $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$ to range over finite sequences of elements. If $\mathbf{a} \in |\mathfrak{M}|^{<\omega}$ and $a \in |\mathfrak{M}|$, then $\mathbf{a}a$ represents the *concatenation* of $\mathbf{a}$ with a.

Definition 25.21. Given structures $\mathfrak{M}$ and $\mathfrak{N}$, we define relations $I_n \subseteq |\mathfrak{M}|^{<\omega} \times |\mathfrak{N}|^{<\omega}$ between sequences of equal length, by recursion on n as follows:

1. $I_0(\mathbf{a}, \mathbf{b})$ if and only if $\mathbf{a}$ and $\mathbf{b}$ satisfy the same atomic formulas in $\mathfrak{M}$ and $\mathfrak{N}$; i.e., if $s_1(x_i) = a_i$ and $s_2(x_i) = b_i$ and φ is atomic with all variables among $x_1, \ldots, x_n$, then $\mathfrak{M}, s_1 \vDash \varphi$ if and only if $\mathfrak{N}, s_2 \vDash \varphi$.

2. $I_{n+1}(\mathbf{a}, \mathbf{b})$ if and only if for every $a \in |\mathfrak{M}|$ there is a $b \in |\mathfrak{N}|$ such that $I_n(\mathbf{a}a, \mathbf{b}b)$, and vice-versa.

Definition 25.22. Write $\mathfrak{M} \approx_n \mathfrak{N}$ if $I_n(\Lambda, \Lambda)$ holds of $\mathfrak{M}$ and $\mathfrak{N}$ (where Λ is the empty sequence).

Theorem 25.23. *Let $\mathcal{L}$ be a purely relational language. Then $I_n(\mathbf{a}, \mathbf{b})$ implies that for every φ such that $\mathrm{qr}(\varphi) \leq n$, we have $\mathfrak{M}, \mathbf{a} \vDash \varphi$ if and only if $\mathfrak{N}, \mathbf{b} \vDash \varphi$ (where again $\mathbf{a}$ satisfies φ if any s such that $s(x_i) = a_i$ satisfies φ). Moreover, if $\mathcal{L}$ is finite, the converse also holds.* mod:bas:pis: thm:b-n-f

Proof. The proof that $I_n(\mathbf{a}, \mathbf{b})$ implies that $\mathbf{a}$ and $\mathbf{b}$ satisfy the same formulas of quantifier rank no greater than n is by an easy induction on φ. For the converse we proceed by induction on n, using Proposition 25.19, which ensures that for each n there are at most finitely many non-equivalent formulas of that quantifier rank.

For $n = 0$ the hypothesis that $\mathbf{a}$ and $\mathbf{b}$ satisfy the same quantifier-free formulas gives that they satisfy the same atomic ones, so that $I_0(\mathbf{a}, \mathbf{b})$.

For the $n + 1$ case, suppose that $\mathbf{a}$ and $\mathbf{b}$ satisfy the same formulas of quantifier rank no greater than $n + 1$; in order to show that $I_{n+1}(\mathbf{a}, \mathbf{b})$ suffices to show that for each $a \in |\mathfrak{M}|$ there is a $b \in |\mathfrak{N}|$ such that $I_n(\mathbf{a}a, \mathbf{b}b)$, and by the inductive hypothesis again suffices to show that for each $a \in |\mathfrak{M}|$ there is a $b \in |\mathfrak{N}|$ such that $\mathbf{a}a$ and $\mathbf{b}b$ satisfy the same formulas of quantifier rank no greater than n.

Given $a \in |\mathfrak{M}|$, let τ^a_n be set of formulas $\psi(x, \mathbf{y})$ of rank no greater than n satisfied by $\mathbf{a}a$ in $\mathfrak{M}$; τ^a_n is finite, so we can assume it is a single first-order formula. It follows that $\mathbf{a}$ satisfies $\exists x\, \tau^a_n(x, \mathbf{y})$, which has quantifier rank no greater than $n + 1$. By hypothesis $\mathbf{b}$ satisfies the same formula in $\mathfrak{N}$, so that there is a $b \in |\mathfrak{N}|$ such that $\mathbf{b}b$ satisfies τ^a_n; in particular, $\mathbf{b}b$ satisfies the same formulas of quantifier rank no greater than n as $\mathbf{a}a$. Similarly one shows that for every $b \in |\mathfrak{N}|$ there is $a \in |\mathfrak{M}|$ such that $\mathbf{a}a$ and $\mathbf{b}b$ satisfy the same formulas of quantifier rank no greater than n, which completes the proof. □

Corollary 25.24. *If $\mathfrak{M}$ and $\mathfrak{N}$ are purely relational structures in a finite language, then $\mathfrak{M} \approx_n \mathfrak{N}$ if and only if $\mathfrak{M} \equiv_n \mathfrak{N}$. In particular $\mathfrak{M} \equiv \mathfrak{N}$ if and only if for each n, $\mathfrak{M} \approx_n \mathfrak{N}$.* mod:bas:pis: cor:b-n-f

---

content/model-theory/basics/dlo.tex

25.7 Dense Linear Orders

Definition 25.25. A *dense linear ordering without endpoints* is a structure $\mathfrak{M}$ for the language containing a single 2-place predicate symbol $<$ satisfying the following sentences:

1. $\forall x \neg x < x$;
2. $\forall x \forall y \forall z\, (x < y \to (y < z \to x < z))$;

3. $\forall x \forall y\, (x < y \vee x = y \vee y < x)$;

4. $\forall x \exists y\, x < y$;

5. $\forall x \exists y\, y < x$;

6. $\forall x \forall y\, (x < y \to \exists z\, (x < z \wedge z < y))$.

mod:bas:dlo: thm:cantorQ

Theorem 25.26. *Any two enumerable dense linear orderings without endpoints are isomorphic.*

Proof. Let $\mathfrak{M}_1$ and $\mathfrak{M}_2$ be enumerable dense linear orderings without endpoints, with $<_1 = <^{\mathfrak{M}_1}$ and $<_2 = <^{\mathfrak{M}_2}$, and let $\mathcal{I}$ be the set of all partial isomorphisms between them. $\mathcal{I}$ is not empty since at least $\emptyset \in \mathcal{I}$. We show that $\mathcal{I}$ satisfies the Back-and-Forth property. Then $\mathfrak{M}_1 \simeq_p \mathfrak{M}_2$, and the theorem follows by Theorem 25.16.

To show $\mathcal{I}$ satisfies the Forth property, let $p \in \mathcal{I}$ and let $p(a_i) = b_i$ for $i = 1, \ldots, n$, and without loss of generality suppose $a_1 <_1 a_2 <_1 \cdots <_1 a_n$. Given $a \in |\mathfrak{M}_1|$, find $b \in |\mathfrak{M}_2|$ as follows:

1. if $a <_1 a_1$ let $b \in |\mathfrak{M}_2|$ be such that $b <_2 b_1$;

2. if $a_n <_1 a$ let $b \in |\mathfrak{M}_2|$ be such that $b_n <_2 b$;

3. if $a_i <_1 a <_1 a_{i+1}$ for some i, then let $b \in |\mathfrak{M}_2|$ be such that $b_i <_2 b <_2 b_{i+1}$.

It is always possible to find a b with the desired property since $\mathfrak{M}_2$ is a dense linear ordering without endpoints. Define $q = p \cup \{\langle a, b\rangle\}$ so that $q \in \mathcal{I}$ is the desired extension of p. This establishes the Forth property. The Back property is similar. So $\mathfrak{M}_1 \simeq_p \mathfrak{M}_2$; by Theorem 25.16, $\mathfrak{M}_1 \simeq \mathfrak{M}_2$. □

Problem 25.5. Complete the proof of Theorem 25.26 by verifying that $\mathcal{I}$ satisfies the Back property.

Remark 4. Let $\mathfrak{S}$ be any enumerable dense linear ordering without endpoints. Then (by Theorem 25.26) $\mathfrak{S} \simeq \mathfrak{Q}$, where $\mathfrak{Q} = (\mathbb{Q}, <)$ is the enumerable dense linear ordering having the set $\mathbb{Q}$ of the rational numbers as its domain. Now consider again the structure $\mathfrak{R} = (\mathbb{R}, <)$ from Remark 2. We saw that there is an enumerable structure $\mathfrak{S}$ such that $\mathfrak{R} \equiv \mathfrak{S}$. But $\mathfrak{S}$ is an enumerable dense linear ordering without endpoints, and so it is isomorphic (and hence elementarily equivalent) to the structure $\mathfrak{Q}$. By transitivity of elementary equivalence, $\mathfrak{R} \equiv \mathfrak{Q}$. (We could have shown this directly by establishing $\mathfrak{R} \simeq_p \mathfrak{Q}$ by the same back-and-forth argument.)

Chapter 26

Models of Arithmetic

content/model-theory/models-of-arithmetic/introduction.tex

26.1 Introduction

The *standard model* of arithmetic is the structure $\mathfrak{N}$ with $|\mathfrak{N}| = \mathbb{N}$ in which o, $\mathsf{'}$, $+$, $\times$, and $<$ are interpreted as you would expect. That is, o is 0, $\mathsf{'}$ is the successor function, $+$ is interpreted as addition and $\times$ as multiplication of the numbers in $\mathbb{N}$. Specifically,

$$\begin{aligned} \mathsf{o}^{\mathfrak{N}} &= 0 \\ \mathsf{'}^{\mathfrak{N}}(n) &= n + 1 \\ +^{\mathfrak{N}}(n, m) &= n + m \\ \times^{\mathfrak{N}}(n, m) &= nm \end{aligned}$$

Of course, there are structures for $\mathcal{L}_A$ that have domains other than $\mathbb{N}$. For instance, we can take $\mathfrak{M}$ with domain $|\mathfrak{M}| = \{a\}^*$ (the finite sequences of the single symbol a, i.e., $\emptyset$, a, aa, aaa, ...), and interpretations

$$\begin{aligned} \mathsf{o}^{\mathfrak{M}} &= \emptyset \\ \mathsf{'}^{\mathfrak{M}}(s) &= s \frown a \\ +^{\mathfrak{M}}(n, m) &= a^{n+m} \\ \times^{\mathfrak{M}}(n, m) &= a^{nm} \end{aligned}$$

These two structures are "essentially the same" in the sense that the only difference is the elements of the domains but not how the elements of the domains are related among each other by the interpretation functions. We say that the two structures are *isomorphic*.

It is an easy consequence of the compactness theorem that any theory true in $\mathfrak{N}$ also has models that are not isomorphic to $\mathfrak{N}$. Such structures are called

non-standard. The interesting thing about them is that while the elements of a standard model (i.e., $\mathfrak{N}$, but also all structures isomorphic to it) are exhausted by the values of the standard numerals $\overline{n}$, i.e.,

$$|\mathfrak{N}| = \{\mathrm{Val}^{\mathfrak{N}}(\overline{n}) : n \in \mathbb{N}\}$$

that isn't the case in non-standard models: if $\mathfrak{M}$ is non-standard, then there is at least one $x \in |\mathfrak{M}|$ such that $x \neq \mathrm{Val}^{\mathfrak{M}}(\overline{n})$ for all n.

These non-standard elements are pretty neat: they are "infinite natural numbers." But their existence also explains, in a sense, the incompleteness phenomena. Consider an example, e.g., the consistency statement for Peano arithmetic, $\mathsf{Con}_{\mathbf{PA}}$, i.e., $\neg\exists x\, \mathsf{Prf}_{\mathbf{PA}}(x, \ulcorner\bot\urcorner)$. Since **PA** neither proves $\mathsf{Con}_{\mathbf{PA}}$ nor $\neg\mathsf{Con}_{\mathbf{PA}}$, either can be consistently added to **PA**. Since **PA** is consistent, $\mathfrak{N} \vDash \mathsf{Con}_{\mathbf{PA}}$, and consequently $\mathfrak{N} \nvDash \neg\mathsf{Con}_{\mathbf{PA}}$. So $\mathfrak{N}$ is *not* a model of $\mathbf{PA} \cup \{\neg\mathsf{Con}_{\mathbf{PA}}\}$, and all its models must be nonstandard. Models of $\mathbf{PA} \cup \{\neg\mathsf{Con}_{\mathbf{PA}}\}$ must contain some element that serves as the witness that makes $\exists x\, \mathsf{Prf}_{\mathbf{PA}}(\ulcorner\bot\urcorner)$ true, i.e., a Gödel number of a derivation of a contradiction from **PA**. Such an element can't be standard—since $\mathbf{PA} \vdash \neg\mathsf{Prf}_{\mathbf{PA}}(\overline{n}, \ulcorner\bot\urcorner)$ for every n.

---

content/model-theory/models-of-arithmetic/standard-models.tex

26.2 Standard Models of Arithmetic

mod:mar:stm: sec

The language of arithmetic $\mathcal{L}_A$ is obviously intended to be about numbers, specifically, about natural numbers. So, "the" standard model $\mathfrak{N}$ is special: it is the model we want to talk about. But in logic, we are often just interested in structural properties, and any two structures that are isomorphic share those. So we can be a bit more liberal, and consider any structure that is isomorphic to $\mathfrak{N}$ "standard."

Definition 26.1. A structure for $\mathcal{L}_A$ is *standard* if it is isomorphic to $\mathfrak{N}$.

mod:mar:stm: prop:standard-domain

Proposition 26.2. *If a structure $\mathfrak{M}$ is standard, then its domain is the set of values of the standard numerals, i.e.,*

$$|\mathfrak{M}| = \{\mathrm{Val}^{\mathfrak{M}}(\overline{n}) : n \in \mathbb{N}\}$$

Proof. Clearly, every $\mathrm{Val}^{\mathfrak{M}}(\overline{n}) \in |\mathfrak{M}|$. We just have to show that every $x \in |\mathfrak{M}|$ is equal to $\mathrm{Val}^{\mathfrak{M}}(\overline{n})$ for some n. Since $\mathfrak{M}$ is standard, it is isomorphic to $\mathfrak{N}$. Suppose $g\colon \mathbb{N} \to |\mathfrak{M}|$ is an isomorphism. Then $g(n) = g(\mathrm{Val}^{\mathfrak{N}}(\overline{n})) = \mathrm{Val}^{\mathfrak{M}}(\overline{n})$. But for every $x \in |\mathfrak{M}|$, there is an $n \in \mathbb{N}$ such that $g(n) = x$, since g is surjective. □

If a structure $\mathfrak{M}$ for $\mathcal{L}_A$ is standard, the elements of its domain can all be named by the standard numerals $\overline{0}$, $\overline{1}$, $\overline{2}$, ..., i.e., the terms o, o', o'', etc. Of course, this does not mean that the elements of $|\mathfrak{M}|$ *are* the numbers, just that we can pick them out the same way we can pick out the numbers in $|\mathfrak{N}|$. explanation

Problem 26.1. Show that the converse of Proposition 26.2 is false, i.e., give an example of a structure $\mathfrak{M}$ with $|\mathfrak{M}| = \{\mathrm{Val}^{\mathfrak{M}}(\overline{n}) : n \in \mathbb{N}\}$ that is not isomorphic to $\mathfrak{N}$.

Proposition 26.3. *If $\mathfrak{M} \vDash \mathbf{Q}$, and $|\mathfrak{M}| = \{\mathrm{Val}^{\mathfrak{M}}(\overline{n}) : n \in \mathbb{N}\}$, then $\mathfrak{M}$ is standard.* mod:mar:stm: prop:thq-standard

Proof. We have to show that $\mathfrak{M}$ is isomorphic to $\mathfrak{N}$. Consider the function $g\colon \mathbb{N} \to |\mathfrak{M}|$ defined by $g(n) = \mathrm{Val}^{\mathfrak{M}}(\overline{n})$. By the hypothesis, g is surjective. It is also injective: $\mathbf{Q} \vdash \overline{n} \neq \overline{m}$ whenever $n \neq m$. Thus, since $\mathfrak{M} \vDash \mathbf{Q}$, $\mathfrak{M} \vDash \overline{n} \neq \overline{m}$, whenever $n \neq m$. Thus, if $n \neq m$, then $\mathrm{Val}^{\mathfrak{M}}(\overline{n}) \neq \mathrm{Val}^{\mathfrak{M}}(\overline{m})$, i.e., $g(n) \neq g(m)$.

We also have to verify that g is an isomorphism.

1. We have $g(\mathsf{o}^{\mathfrak{N}}) = g(0)$ since, $\mathsf{o}^{\mathfrak{N}} = 0$. By definition of g, $g(0) = \mathrm{Val}^{\mathfrak{M}}(\overline{0})$. But $\overline{0}$ is just o, and the value of a term which happens to be a constant symbol is given by what the structure assigns to that constant symbol, i.e., $\mathrm{Val}^{\mathfrak{M}}(\mathsf{o}) = \mathsf{o}^{\mathfrak{M}}$. So we have $g(\mathsf{o}^{\mathfrak{N}}) = \mathsf{o}^{\mathfrak{M}}$ as required.

2. $g(\mathsf{f}^{\mathfrak{N}}(n)) = g(n+1)$, since f in $\mathfrak{N}$ is the successor function on $\mathbb{N}$. Then, $g(n+1) = \mathrm{Val}^{\mathfrak{M}}(\overline{n+1})$ by definition of g. But $\overline{n+1}$ is the same term as $\overline{n}'$, so $\mathrm{Val}^{\mathfrak{M}}(\overline{n+1}) = \mathrm{Val}^{\mathfrak{M}}(\overline{n}')$. By the definition of the value function, this is $= \mathsf{f}^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(\overline{n}))$. Since $\mathrm{Val}^{\mathfrak{M}}(\overline{n}) = g(n)$ we get $g(\mathsf{f}^{\mathfrak{N}}(n)) = \mathsf{f}^{\mathfrak{M}}(g(n))$.

3. $g(+^{\mathfrak{N}}(n,m)) = g(n+m)$, since $+$ in $\mathfrak{N}$ is the addition function on $\mathbb{N}$. Then, $g(n+m) = \mathrm{Val}^{\mathfrak{M}}(\overline{n+m})$ by definition of g. But $\mathbf{Q} \vdash \overline{n+m} = (\overline{n} + \overline{m})$, so $\mathrm{Val}^{\mathfrak{M}}(\overline{n+m}) = \mathrm{Val}^{\mathfrak{M}}(\overline{n} + \overline{m})$. By the definition of the value function, this is $= +^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(\overline{n}), \mathrm{Val}^{\mathfrak{M}}(\overline{m}))$. Since $\mathrm{Val}^{\mathfrak{M}}(\overline{n}) = g(n)$ and $\mathrm{Val}^{\mathfrak{M}}(\overline{m}) = g(m)$, we get $g(+^{\mathfrak{N}}(n,m)) = +^{\mathfrak{M}}(g(n), g(m))$.

4. $g(\times^{\mathfrak{N}}(n,m)) = \times^{\mathfrak{M}}(g(n), g(m))$: Exercise.

5. $\langle n, m\rangle \in <^{\mathfrak{N}}$ iff $n < m$. If $n < m$, then $\mathbf{Q} \vdash \overline{n} < \overline{m}$, and also $\mathfrak{M} \vDash \overline{n} < \overline{m}$. Thus $\langle \mathrm{Val}^{\mathfrak{M}}(\overline{n}), \mathrm{Val}^{\mathfrak{M}}(\overline{m})\rangle \in <^{\mathfrak{M}}$, i.e., $\langle g(n), g(m)\rangle \in <^{\mathfrak{M}}$. If $n \not< m$, then $\mathbf{Q} \vdash \neg\overline{n} < \overline{m}$, and consequently $\mathfrak{M} \nvDash \overline{n} < \overline{m}$. Thus, as before, $\langle g(n), g(m)\rangle \notin <^{\mathfrak{M}}$. Together, we get: $\langle n, m\rangle \in <^{\mathfrak{N}}$ iff $\langle g(n), g(m)\rangle \in <^{\mathfrak{M}}$. □

explanation The function g is the most obvious way of defining a mapping from $\mathbb{N}$ to the domain of any other structure $\mathfrak{M}$ for $\mathcal{L}_A$, since every such $\mathfrak{M}$ contains elements named by $\overline{0}$, $\overline{1}$, $\overline{2}$, etc. So it isn't surprising that if $\mathfrak{M}$ makes at least some basic statements about the $\overline{n}$'s true in the same way that $\mathfrak{N}$ does, and g is also bijective, then g will turn into an isomorphism. In fact, if $|\mathfrak{M}|$ contains no elements other than what the $\overline{n}$'s name, it's the only one.

Proposition 26.4. *If $\mathfrak{M}$ is standard, then g from the proof of Proposition 26.3 is the only isomorphism from $\mathfrak{N}$ to $\mathfrak{M}$.* mod:mar:stm: prop:thq-unique-iso

Proof. Suppose $h\colon \mathbb{N} \to |\mathfrak{M}|$ is an isomorphism between $\mathfrak{N}$ and $\mathfrak{M}$. We show that $g = h$ by induction on n. If $n = 0$, then $g(0) = \mathsf{o}^{\mathfrak{M}}$ by definition of g. But since h is an isomorphism, $h(0) = h(\mathsf{o}^{\mathfrak{N}}) = \mathsf{o}^{\mathfrak{M}}$, so $g(0) = h(0)$.

Now consider the case for $n + 1$. We have

$$\begin{aligned}
g(n+1) &= \mathrm{Val}^{\mathfrak{M}}(\overline{n+1}) \text{ by definition of } g \\
&= \mathrm{Val}^{\mathfrak{M}}(\overline{n}') \text{ since } \overline{n+1} \equiv \overline{n}' \\
&= \mathit{\prime}^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(\overline{n})) \text{ by definition of } \mathrm{Val}^{\mathfrak{M}}(t') \\
&= \mathit{\prime}^{\mathfrak{M}}(g(n)) \text{ by definition of } g \\
&= \mathit{\prime}^{\mathfrak{M}}(h(n)) \text{ by induction hypothesis} \\
&= h(\mathit{\prime}^{\mathfrak{N}}(n)) \text{ since } h \text{ is an isomorphism} \\
&= h(n+1)
\end{aligned}$$

□

For any denumerable set M, there's a bijection between $\mathbb{N}$ and M, so every such set M is potentially the domain of a standard model $\mathfrak{M}$. In fact, once you pick an object $z \in M$ and a suitable function s as $\mathsf{o}^{\mathfrak{M}}$ and $\mathit{\prime}^{\mathfrak{M}}$, the interpretations of $+$, $\times$, and $<$ is already fixed. Only functions $s\colon M \to M \setminus \{z\}$ that are both injective and surjective are suitable in a standard model as $\mathit{\prime}^{\mathfrak{M}}$. The range of s cannot contain z, since otherwise $\forall x\, \mathsf{o} \neq x'$ would be false. That sentence is true in $\mathfrak{N}$, and so $\mathfrak{M}$ also has to make it true. The function s has to be injective, since the successor function $\mathit{\prime}^{\mathfrak{N}}$ in $\mathfrak{N}$ is, and that $\mathit{\prime}^{\mathfrak{N}}$ is injective is expressed by a sentence true in $\mathfrak{N}$. It has to be surjective because otherwise there would be some $x \in M \setminus \{z\}$ not in the domain of s, i.e., the sentence $\forall x\, (x = \mathsf{o} \vee \exists y\, y' = x)$ would be false in $\mathfrak{M}$—but it is true in $\mathfrak{N}$. explanation

---

content/model-theory/models-of-arithmetic/non-standard-models.tex

26.3 Non-Standard Models

We call a structure for $\mathcal{L}_A$ standard if it is isomorphic to $\mathfrak{N}$. If a structure isn't isomorphic to $\mathfrak{N}$, it is called non-standard. explanation

Definition 26.5. A structure $\mathfrak{M}$ for $\mathcal{L}_A$ is *non-standard* if it is not isomorphic to $\mathfrak{N}$. The elements $x \in |\mathfrak{M}|$ which are equal to $\mathrm{Val}^{\mathfrak{M}}(\overline{n})$ for some $n \in \mathbb{N}$ are called *standard numbers* (of $\mathfrak{M}$), and those not, *non-standard numbers*.

By Proposition 26.2, any standard structure for $\mathcal{L}_A$ contains only standard elements. Consequently, a non-standard structure must contain at least one non-standard element. In fact, the existence of a non-standard element guarantees that the structure is non-standard. explanation

Proposition 26.6. *If a structure $\mathfrak{M}$ for $\mathcal{L}_A$ contains a non-standard number, $\mathfrak{M}$ is non-standard.*

Proof. Suppose not, i.e., suppose $\mathfrak{M}$ standard but contains a non-standard number x. Let $g\colon \mathbb{N} \to |\mathfrak{M}|$ be an isomorphism. It is easy to see (by induction on n) that $g(\mathrm{Val}^{\mathfrak{N}}(\overline{n})) = \mathrm{Val}^{\mathfrak{M}}(\overline{n})$. In other words, g maps standard numbers of $\mathfrak{N}$ to standard numbers of $\mathfrak{M}$. If $\mathfrak{M}$ contains a non-standard number, g cannot be surjective, contrary to hypothesis. □

Problem 26.2. Recall that **Q** contains the axioms

$$\forall x\, \forall y\, (x' = y' \to x = y) \tag{Q_1}$$
$$\forall x\, \mathsf{o} \neq x' \tag{Q_2}$$
$$\forall x\, (x = \mathsf{o} \vee \exists y\, x = y') \tag{Q_3}$$

Give structures $\mathfrak{M}_1$, $\mathfrak{M}_2$, $\mathfrak{M}_3$ such that

1. $\mathfrak{M}_1 \vDash Q_1$, $\mathfrak{M}_1 \vDash Q_2$, $\mathfrak{M}_1 \nvDash Q_3$;
2. $\mathfrak{M}_2 \vDash Q_1$, $\mathfrak{M}_2 \nvDash Q_2$, $\mathfrak{M}_2 \vDash Q_3$; and
3. $\mathfrak{M}_3 \nvDash Q_1$, $\mathfrak{M}_3 \vDash Q_2$, $\mathfrak{M}_3 \vDash Q_3$;

Obviously, you just have to specify $\mathsf{o}^{\mathfrak{M}_i}$ and $\mathsf{'}^{\mathfrak{M}_i}$ for each.

explanation

It is easy enough to specify non-standard structures for $\mathcal{L}_A$. For instance, take the structure with domain $\mathbb{Z}$ and interpret all non-logical symbols as usual. Since negative numbers are not values of $\overline{n}$ for any n, this structure is non-standard. Of course, it will not be a *model* of arithmetic in the sense that it makes the same sentences true as $\mathfrak{N}$. For instance, $\forall x\, x' \neq \mathsf{o}$ is false. However, we can prove that non-standard models of arithmetic exist easily enough, using the compactness theorem.

Proposition 26.7. *Let* $\mathbf{TA} = \{\varphi : \mathfrak{N} \vDash \varphi\}$ *be the theory of* $\mathfrak{N}$. **TA** *has an enumerable non-standard model.*

Proof. Expand $\mathcal{L}_A$ by a new constant symbol c and consider the set of sentences

$$\Gamma = \mathbf{TA} \cup \{c \neq \overline{0}, c \neq \overline{1}, c \neq \overline{2}, \dots\}$$

Any model $\mathfrak{M}^c$ of Γ would contain an element $x = c^{\mathfrak{M}}$ which is non-standard, since $x \neq \mathrm{Val}^{\mathfrak{M}}(\overline{n})$ for all $n \in \mathbb{N}$. Also, obviously, $\mathfrak{M}^c \vDash \mathbf{TA}$, since $\mathbf{TA} \subseteq \Gamma$. If we turn $\mathfrak{M}^c$ into a structure $\mathfrak{M}$ for $\mathcal{L}_A$ simply by forgetting about c, its domain still contains the non-standard x, and also $\mathfrak{M} \vDash \mathbf{TA}$. The latter is guaranteed since c does not occur in **TA**. So, it suffices to show that Γ has a model.

We use the compactness theorem to show that Γ has a model. If every finite subset of Γ is satisfiable, so is Γ. Consider any finite subset $\Gamma_0 \subseteq \Gamma$. Γ_0 includes some sentences of **TA** and some of the form $c \neq \overline{n}$, but only finitely many. Suppose k is the largest number so that $c \neq \overline{k} \in \Gamma_0$. Define $\mathfrak{N}_k$ by expanding $\mathfrak{N}$ to include the interpretation $c^{\mathfrak{N}_k} = k + 1$. $\mathfrak{N}_k \vDash \Gamma_0$: if $\varphi \in \mathbf{TA}$, $\mathfrak{N}_k \vDash \varphi$ since $\mathfrak{N}_k$ is just like $\mathfrak{N}$ in all respects except c, and c does not occur in φ. And $\mathfrak{N}_k \vDash c \neq \overline{n}$, since $n \leq k$, and $\mathrm{Val}^{\mathfrak{N}_k}(c) = k + 1$. Thus, every finite subset of Γ is satisfiable. □

content/model-theory/models-of-arithmetic/models-of-q.tex

26.4 Models of Q

We know that there are non-standard structures that make the same sentences true as $\mathfrak{N}$ does, i.e., is a model of **TA**. Since $\mathfrak{N} \vDash \mathbf{Q}$, any model of **TA** is also a model of **Q**. **Q** is much weaker than **TA**, e.g., $\mathbf{Q} \nvdash \forall x \forall y\, (x + y) = (y + x)$. Weaker theories are easier to satisfy: they have more models. E.g., **Q** has models which make $\forall x \forall y\, (x + y) = (y + x)$ false, but those cannot also be models of **TA**, or **PA** for that matter. Models of **Q** are also relatively simple: we can specify them explicitly.

explanation

mod:mar:mdq: ex:model-K-of-Q

Example 26.8. Consider the structure $\mathfrak{K}$ with domain $|\mathfrak{K}| = \mathbb{N} \cup \{a\}$ and interpretations

$$\mathsf{0}^{\mathfrak{K}} = 0$$

$$\prime^{\mathfrak{K}}(x) = \begin{cases} x + 1 & \text{if } x \in \mathbb{N} \\ a & \text{if } x = a \end{cases}$$

$$+^{\mathfrak{K}}(x, y) = \begin{cases} x + y & \text{if } x, y \in \mathbb{N} \\ a & \text{otherwise} \end{cases}$$

$$\times^{\mathfrak{K}}(x, y) = \begin{cases} xy & \text{if } x, y \in \mathbb{N} \\ 0 & \text{if } x = 0 \text{ or } y = 0 \\ a & \text{otherwise} \end{cases}$$

$$<^{\mathfrak{K}} = \{\langle x, y\rangle : x, y \in \mathbb{N} \text{ and } x < y\} \cup \{\langle x, a\rangle : x \in |\mathfrak{K}|\}$$

To show that $\mathfrak{K} \vDash \mathbf{Q}$ we have to verify that all axioms of **Q** are true in $\mathfrak{K}$. For convenience, let's write x^* for $\prime^{\mathfrak{K}}(x)$ (the "successor" of x in $\mathfrak{K}$), $x \oplus y$ for $+^{\mathfrak{K}}(x, y)$ (the "sum" of x and y in $\mathfrak{K}$, $x \otimes y$ for $\times^{\mathfrak{K}}(x, y)$ (the "product" of x and y in $\mathfrak{K}$), and $x \lessdot y$ for $\langle x, y\rangle \in <^{\mathfrak{K}}$. With these abbreviations, we can give the operations in $\mathfrak{K}$ more perspicuously as

x	x^*
n	$n+1$
a	a

$x \oplus y$	0	m	a
0	0	m	a
n	n	$n+m$	a
a	a	a	a

$x \otimes y$	0	m	a
0	0	0	0
n	0	nm	a
a	0	a	a

We have $n \lessdot m$ iff $n < m$ for $n, m \in \mathbb{N}$ and $x \lessdot a$ for all $x \in |\mathfrak{K}|$.

$\mathfrak{K} \vDash \forall x \forall y\, (x' = y' \to x = y)$ since $*$ is injective. $\mathfrak{K} \vDash \forall x\, \mathsf{0} \neq x'$ since 0 is not a $*$-successor in $\mathfrak{K}$. $\mathfrak{K} \vDash \forall x\, (x = \mathsf{0} \vee \exists y\, x = y')$ since for every $n > 0$, $n = (n-1)^*$, and $a = a^*$.

$\mathfrak{K} \vDash \forall x\, (x + \mathsf{0}) = x$ since $n \oplus 0 = n + 0 = n$, and $a \oplus 0 = a$ by definition of $\oplus$. $\mathfrak{K} \vDash \forall x \forall y\, (x + y') = (x + y)'$ is a bit trickier. If n, m are both standard,

we have:

$$(n \oplus m^*) = (n + (m + 1)) = (n + m) + 1 = (n \oplus m)^*$$

since $\oplus$ and * agree with $+$ and $'$ on standard numbers. Now suppose $x \in |\mathfrak{K}|$. Then

$$(x \oplus a^*) = (x \oplus a) = a = a^* = (x \oplus a)^*$$

The remaining case is if $y \in |\mathfrak{K}|$ but $x = a$. Here we also have to distinguish cases according to whether $y = n$ is standard or $y = b$:

$$(a \oplus n^*) = (a \oplus (n + 1)) = a = a^* = (a \oplus n)^*$$
$$(a \oplus a^*) = (a \oplus a) = a = a^* = (a \oplus a)^*$$

This is of course a bit more detailed than needed. For instance, since $a \oplus z = a$ whatever z is, we can immediately conclude $a \oplus a^* = a$. The remaining axioms can be verified the same way.

$\mathfrak{K}$ is thus a model of $\mathbf{Q}$. Its "addition" $\oplus$ is also commutative. But there are other sentences true in $\mathfrak{N}$ but false in $\mathfrak{K}$, and vice versa. For instance, $a \trianglelefteq a$, so $\mathfrak{K} \vDash \exists x\, x < x$ and $\mathfrak{K} \nvDash \forall x \neg x < x$. This shows that $\mathbf{Q} \nvdash \forall x \neg x < x$.

Problem 26.3. Prove that $\mathfrak{K}$ from Example 26.8 satisfies the remaining axioms of $\mathbf{Q}$,

$$\forall x\, (x \times \mathsf{0}) = \mathsf{0} \tag{Q_6}$$
$$\forall x \forall y\, (x \times y') = ((x \times y) + x) \tag{Q_7}$$
$$\forall x \forall y\, (x < y \leftrightarrow \exists z\, (z' + x) = y) \tag{Q_8}$$

Find a sentence only involving $'$ true in $\mathfrak{N}$ but false in $\mathfrak{K}$.

Example 26.9. Consider the structure $\mathfrak{L}$ with domain $|\mathfrak{L}| = \mathbb{N} \cup \{a, b\}$ and interpretations $'^{\mathfrak{L}} = *$, $+^{\mathfrak{L}} = \oplus$ given by mod:mar:mdq: ex:model-L-of-Q

x	x^*
n	$n+1$
a	a
b	b

$x \oplus y$	m	a	b
n	$n+m$	b	a
a	a	b	a
b	b	b	a

Since $*$ is injective, 0 is not in its range, and every $x \in |\mathfrak{L}|$ other than 0 is, axioms Q_1–Q_3 are true in $\mathfrak{L}$. For any x, $x \oplus 0 = x$, so Q_4 is true as well. For Q_5, consider $x \oplus y^*$ and $(x \oplus y)^*$. They are equal if x and y are both standard, since then $*$ and $\oplus$ agree with $'$ and $+$. If x is non-standard, and y is standard, we have $x \oplus y^* = x = x^* = (x \oplus y)^*$. If x and y are both non-standard, we have four cases:

$$a \oplus a^* = b = b^* = (a \oplus a)^*$$
$$b \oplus b^* = a = a^* = (b \oplus b)^*$$
$$b \oplus a^* = b = b^* = (b \oplus y)^*$$
$$a \oplus b^* = a = a^* = (a \oplus b)^*$$

If x is standard, but y is non-standard, we have

$$n \oplus a^* = n \oplus a = b = b^* = (n \oplus a)^*$$
$$n \oplus b^* = n \oplus b = a = a^* = (n \oplus b)^*$$

So, $\mathfrak{L} \vDash Q_5$. However, $a \oplus 0 \neq 0 \oplus a$, so $\mathfrak{L} \nvDash \forall x \forall y\, (x + y) = (y + x)$.

Problem 26.4. Expand $\mathfrak{L}$ of Example 26.9 to include $\otimes$ and $\Subset$ that interpret $\times$ and $<$. Show that your structure satisfies the remaining axioms of **Q**,

$$\forall x\, (x \times \mathsf{o}) = \mathsf{o} \tag{Q_6}$$
$$\forall x \forall y\, (x \times y') = ((x \times y) + x) \tag{Q_7}$$
$$\forall x \forall y\, (x < y \leftrightarrow \exists z\, (z' + x) = y) \tag{Q_8}$$

Problem 26.5. In $\mathfrak{L}$ of Example 26.9, $a^* = a$ and $b^* = b$. Is there a model of **Q** in which $a^* = b$ and $b^* = a$?

We've explicitly constructed models of **Q** in which the non-standard elements live "beyond" the standard elements. In fact, that much is required by the axioms. A non-standard element x cannot be $\Subset 0$, since $\mathbf{Q} \vdash \forall x \neg x < 0$ (see Lemma 35.23). Also, for every n, $\mathbf{Q} \vdash \forall x\, (x < \overline{n}' \rightarrow (x = \overline{0} \vee x = \overline{1} \vee \dots \vee x = \overline{n}))$ (Lemma 35.24), so we can't have $a \Subset n$ for any $n > 0$. explanation

---

content/model-theory/models-of-arithmetic/models-of-pa.tex

26.5 Models of PA

Any non-standard model of **TA** is also one of **PA**. We know that non-standard models of **TA** and hence of **PA** exist. We also know that such non-standard models contain non-standard "numbers," i.e., elements of the domain that are "beyond" all the standard "numbers." But how are they arranged? How many are there? We've seen that models of the weaker theory **Q** can contain as few as a single non-standard number. But these simple structures are not models of **PA** or **TA**. explanation

The key to understanding the structure of models of **PA** or **TA** is to see what facts are derivable in these theories. For instance, already **PA** proves that $\forall x\, x \neq x'$ and $\forall x \forall y\, (x + y) = (y + x)$, so this rules out simple structures (in which these sentences are false) as models of **PA**.

Suppose $\mathfrak{M}$ is a model of **PA**. Then if $\mathbf{PA} \vdash \varphi$, $\mathfrak{M} \vDash \varphi$. Let's again use $\mathbf{z}$ for $\mathsf{o}^{\mathfrak{M}}$, $*$ for $\prime^{\mathfrak{M}}$, $\oplus$ for $+^{\mathfrak{M}}$, $\otimes$ for $\times^{\mathfrak{M}}$, and $\Subset$ for $<^{\mathfrak{M}}$. Any sentence φ then states some condition about $\mathbf{z}$, $*$, $\oplus$, $\otimes$, and $\Subset$, and if $\mathfrak{M} \vDash \varphi$ that condition must be satisfied. For instance, if $\mathfrak{M} \vDash Q_1$, i.e., $\mathfrak{M} \vDash \forall x \forall y\, (x' = y' \rightarrow x = y)$, then $*$ must be injective.

Proposition 26.10. *In $\mathfrak{M}$, $\Subset$ is a linear strict order, i.e., it satisfies:*

1. *Not $x \Subset x$ for any $x \in |\mathfrak{M}|$.*

2. *If $x \trianglelefteq y$ and $y \trianglelefteq z$ then $x \trianglelefteq z$.*

3. *For any $x \neq y$, $x \trianglelefteq y$ or $y \trianglelefteq x$*

Proof. **PA** proves:

1. $\forall x \neg x < x$

2. $\forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z)$

3. $\forall x \forall y ((x < y \vee y < x) \vee x = y))$ □

Proposition 26.11. $\mathbf{z}$ *is the least element of $|\mathfrak{M}|$ in the $\trianglelefteq$-ordering. For any x, $x \trianglelefteq x^*$, and x^* is the $\trianglelefteq$-least element with that property. For any x, there is a unique y such that $y^* = x$. (We call y the "predecessor" of x in $\mathfrak{M}$, and denote it by ${}^*x$.)* mod:mar:mpa: prop:M-discrete

Proof. Exercise. □

Problem 26.6. Find sentences in $\mathcal{L}_A$ derivable in **PA** (and hence true in $\mathfrak{N}$) which guarantee the properties of $\mathbf{z}$, $*$, and $\trianglelefteq$ in Proposition 26.11

Proposition 26.12. *All standard elements of $\mathfrak{M}$ are less than (according to $\trianglelefteq$) all non-standard elements.*

Proof. We'll use n as short for $\mathrm{Val}^{\mathfrak{M}}(\overline{n})$, a standard element of $\mathfrak{M}$. Already **Q** proves that, for any $n \in \mathbb{N}$, $\forall x (x < \overline{n}' \rightarrow (x = \overline{0} \vee x = \overline{1} \vee \cdots \vee x = \overline{n}))$. There are no elements that are $\trianglelefteq \mathbf{z}$. So if n is standard and x is non-standard, we cannot have $x \trianglelefteq n$. By definition, a non-standard element is one that isn't $\mathrm{Val}^{\mathfrak{M}}(\overline{n})$ for any $n \in \mathbb{N}$, so $x \neq n$ as well. Since $\trianglelefteq$ is a linear order, we must have $n \trianglelefteq x$. □

Proposition 26.13. *Every nonstandard element x of $|\mathfrak{M}|$ is an element of the subset*

$$\ldots^{***} x \trianglelefteq^{**} x \trianglelefteq^{*} x \trianglelefteq x \trianglelefteq x^* \trianglelefteq x^{**} \trianglelefteq x^{***} \trianglelefteq \ldots$$

We call this subset the block *of x and write it as $[x]$. It has no least and no greatest element. It can be characterized as the set of those $y \in |\mathfrak{M}|$ such that, for some standard n, $x \oplus n = y$ or $y \oplus n = x$.*

Proof. Clearly, such a set $[x]$ always exists since every element y of $|\mathfrak{M}|$ has a unique successor y^* and unique predecessor ${}^*y$. For successive elements y, y^* we have $y \trianglelefteq y^*$ and y^* is the $\trianglelefteq$-least element of $|\mathfrak{M}|$ such that y is $\trianglelefteq$-less than it. Since always ${}^*y \trianglelefteq y$ and $y \trianglelefteq y^*$, $[x]$ has no least or greatest element. If $y \in [x]$ then $x \in [y]$, for then either $y^{*\cdots*} = x$ or $x^{*\cdots*} = y$. If $y^{*\cdots*} = x$ (with n $*$'s), then $y \oplus n = x$ and conversely, since $\mathbf{PA} \vdash \forall x\, x'^{\cdots'} = (x + \overline{n})$ (if n is the number of $'$'s). □

Proposition 26.14. *If $[x] \neq [y]$ and $x \trianglelefteq y$, then for any $u \in [x]$ and any $v \in [y]$, $u \trianglelefteq v$.*

Proof. Note that $\mathbf{PA} \vdash \forall x \forall y\,(x < y \to (x' < y \vee x' = y))$. Thus, if $u \trianglelefteq v$, we also have $u \oplus n^* \trianglelefteq v$ for any n if $[u] \neq [v]$.

Any $u \in [x]$ is $\trianglelefteq y$: $x \trianglelefteq y$ by assumption. If $u \trianglelefteq x$, $u \trianglelefteq y$ by transitivity. And if $x \trianglelefteq u$ but $u \in [x]$, we have $u = x \oplus n^*$ for some n, and so $u \trianglelefteq y$ by the fact just proved.

Now suppose that $v \in [y]$ is $\trianglelefteq y$, i.e., $v \oplus m^* = y$ for some standard m. This rules out $v \trianglelefteq x$, otherwise $y = v \oplus m^* \trianglelefteq x$. Clearly also, $x \neq v$, otherwise $x \oplus m^* = v \oplus m^* = y$ and we would have $[x] = [y]$. So, $x \trianglelefteq v$. But then also $x \oplus n^* \trianglelefteq v$ for any n. Hence, if $x \trianglelefteq u$ and $u \in [x]$, we have $u \trianglelefteq v$. If $u \trianglelefteq x$ then $u \trianglelefteq v$ by transitivity.

Lastly, if $y \trianglelefteq v$, $u \trianglelefteq v$ since, as we've shown, $u \trianglelefteq y$ and $y \trianglelefteq v$. □

Corollary 26.15. *If $[x] \neq [y]$, $[x] \cap [y] = \emptyset$.*

Proof. Suppose $z \in [x]$ and $x \trianglelefteq y$. Then $z \trianglelefteq u$ for all $u \in [y]$. If $z \in [y]$, we would have $z \trianglelefteq z$. Similarly if $y \trianglelefteq x$. □

This means that the blocks themselves can be ordered in a way that respects $\trianglelefteq$: $[x] \trianglelefteq [y]$ iff $x \trianglelefteq y$, or, equivalently, if $u \trianglelefteq v$ for any $u \in [x]$ and $v \in [y]$. Clearly, the standard block $[0]$ is the least block. It intersects with no non-standard block, and no two non-standard blocks intersect either. Specifically, you cannot "reach" a different block by taking repeated successors or predecessors. explanation

Proposition 26.16. *If x and y are non-standard, then $x \trianglelefteq x \oplus y$ and $x \oplus y \notin [x]$.*

Proof. If y is nonstandard, then $y \neq \mathbf{z}$. $\mathbf{PA} \vdash \forall x\,(y \neq \mathsf{o} \to x < (x + y))$. Now suppose $x \oplus y \in [x]$. Since $x \trianglelefteq x \oplus y$, we would have $x \oplus n^* = x \oplus y$. But $\mathbf{PA} \vdash \forall x \forall y \forall z\,((x + y) = (x + z) \to y = z)$ (the cancellation law for addition). This would mean $y = n^*$ for some standard n; but y is assumed to be non-standard. □

Proposition 26.17. *There is no least non-standard block.*

Proof. $\mathbf{PA} \vdash \forall x\,\exists y\,((y+y) = x \vee (y+y)' = x)$, i.e., that every x is divisible by 2 (possibly with remainder 1). If x is non-standard, so is y. By the preceding proposition, $y \trianglelefteq y \oplus y$ and $y \oplus y \notin [y]$. Then also $y \trianglelefteq (y \oplus y)^*$ and $(y \oplus y)^* \notin [y]$. But $x = y \oplus y$ or $x = (y \oplus y)^*$, so $y \trianglelefteq x$ and $y \notin [x]$. □

Proposition 26.18. *There is no largest block.*

Proof. Exercise. □

Problem 26.7. Show that in a non-standard model of **PA**, there is no largest block.

Proposition 26.19. *The ordering of the blocks is dense. That is, if $x \oslash y$ and $[x] \neq [y]$, then there is a block $[z]$ distinct from both that is between them.* mod:mar:mpa: prop:blocks-dense

Proof. Suppose $x \oslash y$. As before, $x \oplus y$ is divisible by two (possibly with remainder): there is a $z \in |\mathfrak{M}|$ such that either $x \oplus y = z \oplus z$ or $x \oplus y = (z \oplus z)^*$. The element z is the "average" of x and y, and $x \oslash z$ and $z \oslash y$. □

Problem 26.8. Write out a detailed proof of Proposition 26.19. Which sentence must **PA** derive in order to guarantee the existence of z? Why is $x \oslash z$ and $z \oslash y$, and why is $[x] \neq [z]$ and $[z] \neq [y]$?

explanation The non-standard blocks are therefore ordered like the rationals: they form a denumerable dense linear ordering without endpoints. One can show that any two such denumerable orderings are isomorphic. It follows that for any two enumerable non-standard models $\mathfrak{M}_1$ and $\mathfrak{M}_2$ of true arithmetic, their reducts to the language containing $<$ and $=$ only are isomorphic. Indeed, an isomorphism h can be defined as follows: the standard parts of $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are isomorphic to the standard model $\mathfrak{N}$ and hence to each other. The blocks making up the non-standard part are themselves ordered like the rationals and therefore isomorphic; an isomorphism of the blocks can be extended to an isomorphism *within* the blocks by matching up arbitrary elements in each, and then taking the image of the successor of x in $\mathfrak{M}_1$ to be the successor of the image of x in $\mathfrak{M}_2$. Note that it does *not* follow that $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are isomorphic in the full language of arithmetic (indeed, isomorphism is always relative to a language), as there are non-isomorphic ways to define addition and multiplication over $|\mathfrak{M}_1|$ and $|\mathfrak{M}_2|$. (This also follows from a famous theorem due to Vaught that the number of countable models of a complete theory cannot be 2.)

---

content/model-theory/models-of-arithmetic/computable-models.tex

26.6 Computable Models of Arithmetic

explanation The standard model $\mathfrak{N}$ has two nice features. Its domain is the natural numbers $\mathbb{N}$, i.e., its elements are just the kinds of things we want to talk about using the language of arithmetic, and the standard numeral $\overline{n}$ actually picks out n. The other nice feature is that the interpretations of the non-logical symbols of $\mathcal{L}_A$ are all *computable*. The successor, addition, and multiplication functions which serve as $'^{\mathfrak{N}}$, $+^{\mathfrak{N}}$, and $\times^{\mathfrak{N}}$ are computable functions of numbers. (Computable by Turing machines, or definable by primitive recursion, say.) And the less-than relation on $\mathfrak{N}$, i.e., $<^{\mathfrak{N}}$, is decidable.

Non-standard models of arithmetical theories such as **Q** and **PA** must contain non-standard elements. Thus their domains typically include elements in addition to $\mathbb{N}$. However, any countable structure can be built on any denumerable set, including $\mathbb{N}$. So there are also non-standard models with domain $\mathbb{N}$.

In such models $\mathfrak{M}$, of course, at least some numbers cannot play the roles they usually play, since some k must be different from $\mathrm{Val}^{\mathfrak{M}}(\overline{n})$ for all $n \in \mathbb{N}$.

Definition 26.20. A structure $\mathfrak{M}$ for $\mathcal{L}_A$ is *computable* iff $|\mathfrak{M}| = \mathbb{N}$ and $\prime^{\mathfrak{M}}$, $+^{\mathfrak{M}}$, $\times^{\mathfrak{M}}$ are computable functions and $<^{\mathfrak{M}}$ is a decidable relation.

mod:mar:cmp: ex:comp-model-q

Example 26.21. Recall the structure $\mathfrak{K}$ from Example 26.8. Its domain was $|\mathfrak{K}| = \mathbb{N} \cup \{a\}$ and interpretations

$$\mathsf{o}^{\mathfrak{K}} = 0$$

$$\prime^{\mathfrak{K}}(x) = \begin{cases} x+1 & \text{if } x \in \mathbb{N} \\ a & \text{if } x = a \end{cases}$$

$$+^{\mathfrak{K}}(x,y) = \begin{cases} x+y & \text{if } x, y \in \mathbb{N} \\ a & \text{otherwise} \end{cases}$$

$$\times^{\mathfrak{K}}(x,y) = \begin{cases} xy & \text{if } x, y \in \mathbb{N} \\ 0 & \text{if } x = 0 \text{ or } y = 0 \\ a & \text{otherwise} \end{cases}$$

$$<^{\mathfrak{K}} = \{\langle x, y\rangle : x, y \in \mathbb{N} \text{ and } x < y\} \cup \{\langle x, a\rangle : n \in |\mathfrak{K}|\}$$

But $|\mathfrak{K}|$ is denumerable and so is equinumerous with $\mathbb{N}$. For instance, $g\colon \mathbb{N} \to |\mathfrak{K}|$ with $g(0) = a$ and $g(n) = n+1$ for $n > 0$ is a bijection. We can turn it into an isomorphism between a new model $\mathfrak{K}'$ of $\mathbf{Q}$ and $\mathfrak{K}$. In $\mathfrak{K}'$, we have to assign different functions and relations to the symbols of $\mathcal{L}_A$, since different elements of $\mathbb{N}$ play the roles of standard and non-standard numbers.

Specifically, 0 now plays the role of a, not of the smallest standard number. The smallest standard number is now 1. So we assign $\mathsf{o}^{\mathfrak{K}'} = 1$. The successor function is also different now: given a standard number, i.e., an $n > 0$, it still returns $n+1$. But 0 now plays the role of a, which is its own successor. So $\prime^{\mathfrak{K}'}(0) = 0$. For addition and multiplication we likewise have

$$+^{\mathfrak{K}'}(x,y) = \begin{cases} x+y-1 & \text{if } x, y > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\times^{\mathfrak{K}'}(x,y) = \begin{cases} 1 & \text{if } x = 1 \text{ or } y = 1 \\ xy - x - y + 2 & \text{if } x, y > 1 \\ 0 & \text{otherwise} \end{cases}$$

And we have $\langle x, y\rangle \in <^{\mathfrak{K}'}$ iff $x < y$ and $x > 0$ and $y > 0$, or if $y = 0$.

All of these functions are computable functions of natural numbers and $<^{\mathfrak{K}'}$ is a decidable relation on $\mathbb{N}$—but they are not the same functions as successor, addition, and multiplication on $\mathbb{N}$, and $<^{\mathfrak{K}'}$ is not the same relation as $<$ on $\mathbb{N}$.

Problem 26.9. Give a structure $\mathfrak{L}'$ with $|\mathfrak{L}'| = \mathbb{N}$ isomorphic to $\mathfrak{L}$ of Example 26.9.

explanation

Example 26.21 shows that **Q** has computable non-standard models with domain $\mathbb{N}$. However, the following result shows that this is not true for models of **PA** (and thus also for models of **TA**).

Theorem 26.22 (Tennenbaum's Theorem). *$\mathfrak{N}$ is the only computable model of* **PA**.

Chapter 27

The Interpolation Theorem

---

content/model-theory/interpolation/introduction.tex

27.1 Introduction

mod:int:int:sec

The interpolation theorem is the following result: Suppose $\vDash \varphi \to \psi$. Then there is a sentence χ such that $\vDash \varphi \to \chi$ and $\vDash \chi \to \psi$. Moreover, every constant symbol, function symbol, and predicate symbol (other than $=$) in χ occurs both in φ and ψ. The sentence χ is called an *interpolant* of φ and ψ.

The interpolation theorem is interesting in its own right, but its main importance lies in the fact that it can be used to prove results about definability in a theory, and the conditions under which combining two consistent theories results in a consistent theory. The first result is known as the Beth definability theorem; the second, Robinson's joint consistency theorem.

---

content/model-theory/interpolation/separation.tex

27.2 Separation of Sentences

mod:int:sep:sec

A bit of groundwork is needed before we can proceed with the proof of the interpolation theorem. An interpolant for φ and ψ is a sentence χ such that $\varphi \vDash \chi$ and $\chi \vDash \psi$. By contraposition, the latter is true iff $\neg\psi \vDash \neg\chi$. A sentence χ with this property is said to *separate* φ and $\neg\psi$. So finding an interpolant for φ and ψ amounts to finding a sentence that separates φ and $\neg\psi$. As so often, it will be useful to consider a generalization: a sentence that separates two *sets* of sentences.

Definition 27.1. A sentence χ *separates* sets of sentences Γ and Δ if and only if $\Gamma \vDash \chi$ and $\Delta \vDash \neg\chi$. If no such sentence exists, then Γ and Δ are *inseparable*.

The inclusion relations between the classes of models of Γ, Δ and χ are represented below:

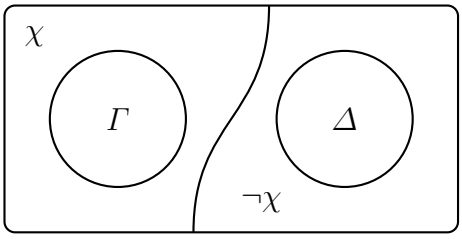


Figure 27.1: χ separates Γ and Δ

mod:int:sep: mod:int:sep:lg:sep lem:sep1

Lemma 27.2. *Suppose $\mathcal{L}_0$ is the language containing every constant symbol, function symbol and predicate symbol (other than $\doteq$) that occurs in* both *Γ and Δ, and let $\mathcal{L}_0'$ be obtained by the addition of infinitely many new constant symbols c_n for $n \geq 0$. Then if Γ and Δ are inseparable in $\mathcal{L}_0$, they are also inseparable in $\mathcal{L}_0'$.*

Proof. We proceed indirectly: suppose by way of contradiction that Γ and Δ are separated in $\mathcal{L}_0'$. Then $\Gamma \vDash \chi[c/x]$ and $\Delta \vDash \neg\chi[c/x]$ for some $\chi \in \mathcal{L}_0$ (where c is a new constant symbol—the case where χ contains more than one such new constant symbol is similar). By compactness, there are *finite* subsets Γ_0 of Γ and Δ_0 of Δ such that $\Gamma_0 \vDash \chi[c/x]$ and $\Delta_0 \vDash \neg\chi[c/x]$. Let γ be the conjunction of all formulas in Γ_0 and δ the conjunction of all formulas in Δ_0. Then

$$\gamma \vDash \chi[c/x], \qquad \delta \vDash \neg\chi[c/x].$$

From the former, by Generalization, we have $\gamma \vDash \forall x\, \chi$, and from the latter by contraposition, $\chi[c/x] \vDash \neg\delta$, whence also $\forall x\, \chi \vDash \neg\delta$. Contraposition again gives $\delta \vDash \neg\forall x\, \chi$. By monotonicity,

$$\Gamma \vDash \forall x\, \chi, \qquad \Delta \vDash \neg\forall x\, \chi,$$

so that $\forall x\, \chi$ separates Γ and Δ in $\mathcal{L}_0$. □

mod:int:sep: lem:sep2

Lemma 27.3. *Suppose that $\Gamma \cup \{\exists x\, \sigma\}$ and Δ are inseparable, and c is a new constant symbol not in Γ, Δ, or σ. Then $\Gamma \cup \{\exists x\, \sigma, \sigma[c/x]\}$ and Δ are also inseparable.*

Proof. Suppose for contradiction that χ separates $\Gamma \cup \{\exists x\, \sigma, \sigma[c/x]\}$ and Δ, while at the same time $\Gamma \cup \{\exists x \sigma\}$ and Δ are inseparable. We distinguish two cases:

1. c does not occur in χ: in this case $\Gamma \cup \{\exists x\, \sigma, \neg\chi\}$ is satisfiable (otherwise χ separates $\Gamma \cup \{\exists x\, \sigma\}$ and Δ). It remains so if $\sigma[c/x]$ is added, so χ does not separate $\Gamma \cup \{\exists x\, \sigma, \sigma[c/x]\}$ and Δ after all.

2. c does occur in χ so that χ has the form $\chi[c/x]$. Then we have that

$$\Gamma \cup \{\exists x\, \sigma, \sigma[c/x]\} \vDash \chi[c/x],$$

whence $\Gamma, \exists x\, \sigma \vDash \forall x\, (\sigma \to \chi)$ by the Deduction Theorem and Generalization, and finally $\Gamma \cup \{\exists x\, \sigma\} \vDash \exists x\, \chi$. On the other hand, $\Delta \vDash \neg\chi[c/x]$ and hence by Generalization $\Delta \vDash \neg\exists x\, \chi$. So $\Gamma \cup \{\exists x\, \sigma\}$ and Δ are separable, a contradiction. □

---

content/model-theory/interpolation/interpolation-proof.tex

27.3 Craig's Interpolation Theorem

mod:int:prf: sec
mod:int:prf: thm:interpol

Theorem 27.4 (Craig's Interpolation Theorem). *If $\vDash \varphi \to \psi$, then there is a sentence χ such that $\vDash \varphi \to \chi$ and $\vDash \chi \to \psi$, and every constant symbol, function symbol, and predicate symbol (other than $=$) in χ occurs both in φ and ψ. The sentence χ is called an* interpolant *of φ and ψ.*

Proof. Suppose $\mathcal{L}_1$ is the language of φ and $\mathcal{L}_2$ is the language of ψ. Let $\mathcal{L}_0 = \mathcal{L}_1 \cap \mathcal{L}_2$. For each $i \in \{0, 1, 2\}$, let $\mathcal{L}_i'$ be obtained from $\mathcal{L}_i$ by adding the infinitely many new constant symbols $\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2, \ldots$.

If φ is unsatisfiable, $\exists x\, x \neq x$ is an interpolant. If $\neg\psi$ is unsatisfiable (and hence ψ is valid), $\exists x\, x = x$ is an interpolant. So we may assume also that both φ and $\neg\psi$ are satisfiable.

In order to prove the contrapositive of the Interpolation Theorem, assume that there is no interpolant for φ and ψ. In other words, assume that $\{\varphi\}$ and $\{\neg\psi\}$ are inseparable in $\mathcal{L}_0$.

Our goal is to extend the pair $(\{\varphi\}, \{\neg\psi\})$ to a maximally inseparable pair (Γ^*, Δ^*). Let φ_0, φ_1, φ_2, ... enumerate the sentences of $\mathcal{L}_1$, and ψ_0, ψ_1, ψ_2, ... enumerate the sentences of $\mathcal{L}_2$. We define two increasing sequences of sets of sentences (Γ_n, Δ_n), for $n \geq 0$, as follows. Put $\Gamma_0 = \{\varphi\}$ and $\Delta_0 = \{\neg\psi\}$. Assuming (Γ_n, Δ_n) are already defined, define Γ_{n+1} and Δ_{n+1} by:

1. If $\Gamma_n \cup \{\varphi_n\}$ and Δ_n are inseparable in $\mathcal{L}_0'$, put φ_n in Γ_{n+1}. Moreover, if φ_n is an existential formula $\exists x\, \sigma$ then pick a new constant symbol c not occurring in Γ_n, Δ_n, φ_n or ψ_n, and put $\sigma[c/x]$ in Γ_{n+1}.

2. If Γ_{n+1} and $\Delta_n \cup \{\psi_n\}$ are inseparable in $\mathcal{L}_0'$, put ψ_n in Δ_{n+1}. Moreover, if ψ_n is an existential formula $\exists x\, \sigma$, then pick a new constant symbol c not occurring in Γ_{n+1}, Δ_n, φ_n or ψ_n, and put $\sigma[c/x]$ in Δ_{n+1}.

Finally, define:

$$\Gamma^* = \bigcup_{n \geq 0} \Gamma_n, \qquad \Delta^* = \bigcup_{n \geq 0} \Delta_n.$$

By simultaneous induction on n we can now prove:

mod:int:prf: part-a
1. Γ_n and Δ_n are inseparable in $\mathcal{L}'_0$;

mod:int:prf: part-b
2. Γ_{n+1} and Δ_n are inseparable in $\mathcal{L}'_0$.

The basis for (1) is given by Lemma 27.2. For part (2), we need to distinguish three cases:

1. If $\Gamma_0 \cup \{\varphi_0\}$ and Δ_0 are separable, then $\Gamma_1 = \Gamma_0$ and (2) is just (1);
2. If $\Gamma_1 = \Gamma_0 \cup \{\varphi_0\}$, then Γ_1 and Δ_0 are inseparable by construction.
3. It remains to consider the case where φ_0 is existential, so that $\Gamma_1 = \Gamma_0 \cup \{\exists x\, \sigma, \sigma[c/x]\}$. By construction, $\Gamma_0 \cup \{\exists x\, \sigma\}$ and Δ_0 are inseparable, so that by Lemma 27.3 also $\Gamma_0 \cup \{\exists x\, \sigma, \sigma[c/x]\}$ and Δ_0 are inseparable.

This completes the basis of the induction for (1) and (2) above. Now for the inductive step. For (1), if $\Delta_{n+1} = \Delta_n \cup \{\psi_n\}$ then Γ_{n+1} and Δ_{n+1} are inseparable by construction (even when ψ_n is existential, by Lemma 27.3); if $\Delta_{n+1} = \Delta_n$ (because Γ_{n+1} and $\Delta_n \cup \{\psi_n\}$ are separable), then we use the induction hypothesis on (2). For the inductive step for (2), if $\Gamma_{n+2} = \Gamma_{n+1} \cup \{\varphi_{n+1}\}$ then Γ_{n+2} and Δ_{n+1} are inseparable by construction (even when φ_{n+1} is existential, by Lemma 27.3); and if $\Gamma_{n+2} = \Gamma_{n+1}$ then we use the inductive case for (1) just proved. This concludes the induction on (1) and (2).

It follows that Γ^* and Δ^* are inseparable; if not, by compactness, there is $n \geq 0$ that separates Γ_n and Δ_n, against (1). In particular, Γ^* and Δ^* are consistent: for if the former or the latter is inconsistent, then they are separated by $\exists x\, x \neq x$ or $\forall x\, x = x$, respectively.

We now show that Γ^* is maximally consistent in $\mathcal{L}'_1$ and likewise Δ^* in $\mathcal{L}'_2$. For the former, suppose that $\varphi_n \notin \Gamma^*$ and $\neg\varphi_n \notin \Gamma^*$, for some $n \geq 0$. If $\varphi_n \notin \Gamma^*$ then $\Gamma_n \cup \{\varphi_n\}$ is separable from Δ_n, and so there is $\chi \in \mathcal{L}'_0$ such that both:

$$\Gamma^* \vDash \varphi_n \to \chi, \qquad \Delta^* \vDash \neg\chi.$$

Likewise, if $\neg\varphi_n \notin \Gamma^*$, there is $\chi' \in \mathcal{L}'_0$ such that both:

$$\Gamma^* \vDash \neg\varphi_n \to \chi', \qquad \Delta^* \vDash \neg\chi'.$$

By propositional logic, $\Gamma^* \vDash \chi \vee \chi'$ and $\Delta^* \vDash \neg(\chi \vee \chi')$, so $\chi \vee \chi'$ separates $\Gamma^*$and Δ^*. A similar argument establishes that Δ^* is maximal.

Finally, we show that $\Gamma^* \cap \Delta^*$ is maximally consistent in $\mathcal{L}'_0$. It is obviously consistent, since it is the intersection of consistent sets. To show maximality, let $\sigma \in \mathcal{L}'_0$. Now, Γ^* is maximal in $\mathcal{L}'_1 \supseteq \mathcal{L}'_0$, and similarly Δ^* is maximal in $\mathcal{L}'_2 \supseteq \mathcal{L}'_0$. It follows that either $\sigma \in \Gamma^*$ or $\neg\sigma \in \Gamma^*$, and either $\sigma \in \Delta^*$ or $\neg\sigma \in \Delta^*$. If $\sigma \in \Gamma^*$ and $\neg\sigma \in \Delta^*$ then σ would separate Γ^* and Δ^*; and if $\neg\sigma \in \Gamma^*$ and $\sigma \in \Delta^*$ then Γ^* and Δ^* would be separated by $\neg\sigma$. Hence, either $\sigma \in \Gamma^* \cap \Delta^*$ or $\neg\sigma \in \Gamma^* \cap \Delta^*$, and $\Gamma^* \cap \Delta^*$ is maximal.

Since Γ^* is maximally consistent, it has a model $\mathfrak{M}'_1$ whose domain $|\mathfrak{M}'_1|$ comprises all and only the elements $c^{\mathfrak{M}'_1}$ interpreting the constant symbols—just like in the proof of the completeness theorem (Theorem 23.20). Similarly, Δ^* has a model $\mathfrak{M}'_2$ whose domain $|\mathfrak{M}'_2|$ is given by the interpretations $c^{\mathfrak{M}'_2}$ of the constant symbols.

Let $\mathfrak{M}_1$ be obtained from $\mathfrak{M}'_1$ by dropping interpretations for constant symbols, function symbols, and predicate symbols in $\mathcal{L}'_1 \setminus \mathcal{L}'_0$, and similarly for $\mathfrak{M}_2$. Then the map $h\colon M_1 \to M_2$ defined by $h(c^{\mathfrak{M}'_1}) = c^{\mathfrak{M}'_2}$ is an isomorphism in $\mathcal{L}'_0$, because $\Gamma^* \cap \Delta^*$ is maximally consistent in $\mathcal{L}'_0$, as shown. This follows because any $\mathcal{L}'_0$-sentence either belongs to both Γ^* and Δ^*, or to neither: so $c^{\mathfrak{M}'_1} \in P^{\mathfrak{M}'_1}$ if and only if $P(c) \in \Gamma^*$ if and only if $P(c) \in \Delta^*$ if and only if $c^{\mathfrak{M}'_2} \in P^{\mathfrak{M}'_2}$. The other conditions satisfied by isomorphisms can be established similarly.

Let us now define a model $\mathfrak{M}$ for the language $\mathcal{L}_1 \cup \mathcal{L}_2$ as follows:

1. The domain $|\mathfrak{M}|$ is just $|\mathfrak{M}_2|$, i.e., the set of all elements $c^{\mathfrak{M}'_2}$;
2. If a predicate symbol P is in $\mathcal{L}_2 \setminus \mathcal{L}_1$ then $P^{\mathfrak{M}} = P^{\mathfrak{M}'_2}$;
3. If a predicate P is in $\mathcal{L}_1 \setminus \mathcal{L}_2$ then $P^{\mathfrak{M}} = h(P^{\mathfrak{M}'_2})$, i.e., $\langle c_1^{\mathfrak{M}'_2}, \dots, c_n^{\mathfrak{M}'_2}\rangle \in P^{\mathfrak{M}}$ if and only if $\langle c_1^{\mathfrak{M}'_1}, \dots, c_n^{\mathfrak{M}'_1}\rangle \in P^{\mathfrak{M}'_1}$.
4. If a predicate symbol P is in $\mathcal{L}_0$ then $P^{\mathfrak{M}} = P^{\mathfrak{M}'_2} = h(P^{\mathfrak{M}'_1})$.
5. Function symbols of $\mathcal{L}_1 \cup \mathcal{L}_2$, including constant symbols, are handled similarly.

Finally, one shows by induction on formulas that $\mathfrak{M}$ agrees with $\mathfrak{M}'_1$ on all formulas of $\mathcal{L}'_1$ and with $\mathfrak{M}'_2$ on all formulas of $\mathcal{L}'_2$. In particular, $\mathfrak{M} \vDash \Gamma^* \cup \Delta^*$, whence $\mathfrak{M} \vDash \varphi$ and $\mathfrak{M} \vDash \neg\psi$, and $\nvdash \varphi \to \psi$. This concludes the proof of Craig's Interpolation Theorem. □

---

content/model-theory/interpolation/definability.tex

27.4 The Definability Theorem

mod:int:def: sec

One important application of the interpolation theorem is Beth's definability theorem. To define an n-place relation R we can give a formula χ with n free variables which does not involve R. This would be an *explicit* definition of R in terms of χ. We can then say also that a theory $\Sigma(P)$ in a language containing the n-place predicate symbol P explicitly defines P if it contains (or at least entails) a formalized explicit definition, i.e.,

$$\Sigma(P) \vDash \forall x_1 \dots \forall x_n\, (P(x_1, \dots, x_n) \leftrightarrow \chi(x_1, \dots, x_n)).$$

But an explicit definition is only one way of defining—in the sense of determining completely—a relation. A theory may also be such that the interpretation

of P is fixed by the interpretation of the rest of the language in any model. The definability theorem states that whenever a theory fixes the interpretation of P in this way—whenever it *implicitly defines* P—then it also explicitly defines it.

Definition 27.5. Suppose $\mathcal{L}$ is a language not containing the predicate symbol P. A set $\Sigma(P)$ of sentences of $\mathcal{L} \cup \{P\}$ *explicitly defines* P if and only if there is a formula $\chi(x_1, \ldots, x_n)$ of $\mathcal{L}$ such that

$$\Sigma(P) \vDash \forall x_1 \ldots \forall x_n \, (P(x_1, \ldots, x_n) \leftrightarrow \chi(x_1, \ldots, x_n)).$$

Definition 27.6. Suppose $\mathcal{L}$ is a language not containing the predicate symbols P and P'. A set $\Sigma(P)$ of sentences of $\mathcal{L} \cup \{P\}$ *implicitly defines* P if and only if

$$\Sigma(P) \cup \Sigma(P') \vDash \forall x_1 \ldots \forall x_n \, (P(x_1, \ldots, x_n) \leftrightarrow P'(x_1, \ldots, x_n)),$$

where $\Sigma(P')$ is the result of uniformly replacing P with P' in $\Sigma(P)$.

In other words, for any model $\mathfrak{M}$ and $R, R' \subseteq |\mathfrak{M}|^n$, if both $(\mathfrak{M}, R) \vDash \Sigma(P)$ and $(\mathfrak{M}, R') \vDash \Sigma(P')$, then $R = R'$; where $(\mathfrak{M}, R)$ is the structure $\mathfrak{M}'$ for the expansion of $\mathcal{L}$ to $\mathcal{L} \cup \{P\}$ such that $P^{\mathfrak{M}'} = R$, and similarly for $(\mathfrak{M}, R')$.

Theorem 27.7 (Beth Definability Theorem). *A set $\Sigma(P)$ of $\mathcal{L} \cup \{P\}$-formulas implicitly defines P if and only $\Sigma(P)$ explicitly defines P.*

Proof. If $\Sigma(P)$ explicitly defines P then both

$$\begin{aligned} \Sigma(P) \vDash \quad & \forall x_1 \ldots \forall x_n \, (P(x_1, \ldots, x_n) \leftrightarrow \chi(x_1, \ldots, x_n)) \\ \Sigma(P') \vDash \quad & \forall x_1 \ldots \forall x_n \, (P'(x_1, \ldots, x_n) \leftrightarrow \chi(x_1, \ldots, x_n)) \end{aligned}$$

and the conclusion follows. For the converse: assume that $\Sigma(P)$ implicitly defines P. First, we add constant symbols $c_1, \ldots, c_n$ to $\mathcal{L}$. Then

$$\Sigma(P) \cup \Sigma(P') \vDash P(c_1, \ldots, c_n) \to P'(c_1, \ldots, c_n).$$

By compactness, there are finite sets $\Delta_0 \subseteq \Sigma(P)$ and $\Delta_1 \subseteq \Sigma(P')$ such that

$$\Delta_0 \cup \Delta_1 \vDash P(c_1, \ldots, c_n) \to P'(c_1, \ldots, c_n).$$

Let $\theta(P)$ be the conjunction of all sentences $\varphi(P)$ such that either $\varphi(P) \in \Delta_0$ or $\varphi(P') \in \Delta_1$ and let $\theta(P')$ be the conjunction of all sentences $\varphi(P')$ such that either $\varphi(P) \in \Delta_0$ or $\varphi(P') \in \Delta_1$. Then $\theta(P) \wedge \theta(P') \vDash P(c_1, \ldots, c_n) \to P' c_1 \ldots c_n$. We can re-arrange this so that each predicate symbol occurs on one side of $\vDash$:

$$\theta(P) \wedge P(c_1, \ldots, c_n) \vDash \theta(P') \to P'(c_1, \ldots, c_n).$$

By Craig's Interpolation Theorem there is a sentence $\chi(c_1, \ldots, c_n)$ not containing P or P' such that:

$$\begin{aligned} \theta(P) \wedge P(c_1, \ldots, c_n) &\vDash \chi(c_1, \ldots, c_n); \\ \chi(c_1, \ldots, c_n) &\vDash \theta(P') \to P'(c_1, \ldots, c_n). \end{aligned}$$

From the former of these two entailments we have: $\theta(P) \vDash P(c_1, \ldots, c_n) \to \chi(c_1, \ldots, c_n)$. And from the latter, since an $\mathcal{L} \cup \{P\}$-model $(\mathfrak{M}, R) \vDash \varphi(P)$ if and only if the corresponding $\mathcal{L} \cup \{P'\}$-model $(\mathfrak{M}, R) \models \varphi(P')$, we have $\chi(c_1, \ldots, c_n) \vDash \theta(P) \to P(c_1, \ldots, c_n)$, from which:

$$\theta(P) \vDash \chi(c_1, \ldots, c_n) \to P(c_1, \ldots, c_n).$$

Putting the two together, $\theta(P) \vDash P(c_1, \ldots, c_n) \leftrightarrow \chi(c_1, \ldots, c_n)$, and by monotonicity and generalization also

$$\Sigma(P) \vDash \forall x_1 \ldots \forall x_n \, (P(x_1, \ldots, x_n) \leftrightarrow \chi(x_1, \ldots, x_n)). \qquad \square$$

Chapter 28

Lindström's Theorem

---

content/model-theory/lindstrom/introduction.tex

28.1 Introduction

In this chapter we aim to prove Lindström's characterization of first-order logic as the maximal logic for which (given certain further constraints) the Compactness and the Downward Löwenheim–Skolem theorems hold (Theorem 23.23 and Theorem 23.32). First, we need a more general characterization of the general class of logics to which the theorem applies. We will restrict ourselves to *relational* languages, i.e., languages which only contain predicate symbols and individual constants, but no function symbols.

---

content/model-theory/lindstrom/abstract-logics.tex

28.2 Abstract Logics

mod:lin:alg:sec

Definition 28.1. An *abstract logic* is a pair $\langle L, \models_L \rangle$, where L is a function that assigns to each language $\mathcal{L}$ a set $L(\mathcal{L})$ of sentences, and $\models_L$ is a relation between structures for the language $\mathcal{L}$ and elements of $L(\mathcal{L})$. In particular, $\langle F, \models \rangle$ is

ordinary first-order logic, i.e., F is the function assigning to the language $\mathcal{L}$ the set of first-order sentences built from the constants in $\mathcal{L}$, and $\models$ is the satisfaction relation of first-order logic.

Notice that we are still employing the same notion of structure for a given language as for first-order logic, but we do not presuppose that sentences are build up from the basic symbols in $\mathcal{L}$ in the usual way, nor that the relation $\models_L$ is recursively defined in the same way as for first-order logic. So for instance the definition, being completely general, is intended to capture the case where sentences in $\langle L, \models_L \rangle$ contain infinitely long conjunctions or disjunction, or quantifiers other than $\exists$ and $\forall$ (e.g., "there are infinitely many x such that ..."), or perhaps infinitely long quantifier prefixes. To emphasize that "sentences" in $L(\mathcal{L})$ need not be ordinary sentences of first-order logic, in this chapter we use variables α, β, ... to range over them, and reserve φ, ψ, ... for ordinary first-order formulas.

Definition 28.2. Let $\mathrm{Mod}_L(\alpha)$ denote the class $\{\mathfrak{M} : \mathfrak{M} \models_L \alpha\}$. If the language needs to be made explicit, we write $\mathrm{Mod}_L^{\mathcal{L}}(\alpha)$. Two structures $\mathfrak{M}$ and $\mathfrak{N}$ for $\mathcal{L}$ are *elementarily equivalent in* $\langle L, \models_L \rangle$, written $\mathfrak{M} \equiv_L \mathfrak{N}$, if the same sentences from $L(\mathcal{L})$ are true in each.

Definition 28.3. An abstract logic $\langle L, \models_L \rangle$ for the language $\mathcal{L}$ is *normal* if it satisfies the following properties:

1. (*L-Monotonicity*) For languages $\mathcal{L}$ and $\mathcal{L}'$, if $\mathcal{L} \subseteq \mathcal{L}'$, then $L(\mathcal{L}) \subseteq L(\mathcal{L}')$.

2. (*Expansion Property*) For each $\alpha \in L(\mathcal{L})$ there is a *finite* subset $\mathcal{L}'$ of $\mathcal{L}$ such that the relation $\mathfrak{M} \models_L \alpha$ depends only on the reduct of $\mathfrak{M}$ to $\mathcal{L}'$; i.e., if $\mathfrak{M}$ and $\mathfrak{N}$ have the same reduct to $\mathcal{L}'$ then $\mathfrak{M} \models_L \alpha$ if and only if $\mathfrak{N} \models_L \alpha$.

3. (*Isomorphism Property*) If $\mathfrak{M} \models_L \alpha$ and $\mathfrak{M} \simeq \mathfrak{N}$ then also $\mathfrak{N} \models_L \alpha$.

4. (*Renaming Property*) The relation $\models_L$ is preserved under renaming: if the language $\mathcal{L}'$ is obtained from $\mathcal{L}$ by replacing each symbol P by a symbol P' of the same arity and each constant c by a distinct constant c', then for each structure $\mathfrak{M}$ and sentence α, $\mathfrak{M} \models_L \alpha$ if and only if $\mathfrak{M}' \models_L \alpha'$, where $\mathfrak{M}'$ is the $\mathcal{L}'$-structure corresponding to $\mathcal{L}$ and $\alpha' \in L(\mathcal{L}')$.

5. (*Boolean Property*) The abstract logic $\langle L, \models_L \rangle$ is closed under the Boolean connectives in the sense that for each $\alpha \in L(\mathcal{L})$ there is a $\beta \in L(\mathcal{L})$ such that $\mathfrak{M} \models_L \beta$ if and only if $\mathfrak{M} \not\models_L \alpha$, and for each α and β there is a γ such that $\mathrm{Mod}_L(\gamma) = \mathrm{Mod}_L(\alpha) \cap \mathrm{Mod}_L(\beta)$. Similarly for atomic formulas and the other connectives.

6. (*Quantifier Property*) For each constant c in $\mathcal{L}$ and $\alpha \in L(\mathcal{L})$ there is a $\beta \in L(\mathcal{L})$ such that

$$\mathrm{Mod}_L^{\mathcal{L}'}(\beta) = \{\mathfrak{M} : (\mathfrak{M}, a) \in \mathrm{Mod}_L^{\mathcal{L}}(\alpha) \text{ for some } a \in |\mathfrak{M}|\},$$

where $\mathcal{L}' = \mathcal{L} \setminus \{c\}$ and $(\mathfrak{M}, a)$ is the expansion of $\mathfrak{M}$ to $\mathcal{L}$ assigning a to c.

7. (*Relativization Property*) Given a sentence $\alpha \in L(\mathcal{L})$ and symbols R, c_1, $\ldots$, c_n not in $\mathcal{L}$, there is a sentence $\beta \in L(\mathcal{L} \cup \{R, c_1, \ldots, c_n\})$ called the *relativization* of α to $R(x, c_1, \ldots c_n)$, such that for each structure $\mathfrak{M}$:

$$(\mathfrak{M}, X, b_1, \ldots, b_n) \models_L \beta \text{ if and only if } \mathfrak{N} \models_L \alpha,$$

where $\mathfrak{N}$ is the substructure of $\mathfrak{M}$ with domain $|\mathfrak{N}| = \{a \in |\mathfrak{M}| : R^{\mathfrak{M}}(a, b_1, \ldots, b_n)\}$ (see Remark 1), and $(\mathfrak{M}, X, b_1, \ldots, b_n)$ is the expansion of $\mathfrak{M}$ interpreting R, c_1, $\ldots$, c_n by X, b_1, $\ldots$, b_n, respectively (with $X \subseteq M^{n+1}$).

Definition 28.4. Given two abstract logics $\langle L_1, \models_{L_1}\rangle$ and $\langle L_2, \models_{L_2}\rangle$ we say that the latter is *at least as expressive* as the former, written $\langle L_1, \models_{L_1}\rangle \leq \langle L_2, \models_{L_2}\rangle$, if for each language $\mathcal{L}$ and sentence $\alpha \in L_1(\mathcal{L})$ there is a sentence $\beta \in L_2(\mathcal{L})$ such that $\mathrm{Mod}^{\mathcal{L}}_{L_1}(\alpha) = \mathrm{Mod}^{\mathcal{L}}_{L_2}(\beta)$. The logics $\langle L_1, \models_{L_1}\rangle$ and $\langle L_2, \models_{L_2}\rangle$ are *equivalent* if $\langle L_1, \models_{L_1}\rangle \leq \langle L_2, \models_{L_2}\rangle$ and $\langle L_2, \models_{L_2}\rangle \leq \langle L_1, \models_{L_1}\rangle$.

Remark 5. First-order logic, i.e., the abstract logic $\langle F, \models\rangle$, is normal. In fact, the above properties are mostly straightforward for first-order logic. We just remark that the expansion property comes down to extensionality, and that the relativization of a sentence α to $R(x, c_1, \ldots, c_n)$ is obtained by replacing each subformula $\forall x\, \beta$ by $\forall x\, (R(x, c_1, \ldots, c_n) \to \beta)$. Moreover, if $\langle L, \models_L\rangle$ is normal, then $\langle F, \models\rangle \leq \langle L, \models_L\rangle$, as can be can shown by induction on first-order formulas. Accordingly, with no loss in generality, we can assume that every first-order sentence belongs to every normal logic.

---

content/model-theory/lindstrom/ls-property.tex

28.3 Compactness and Löwenheim–Skolem Properties

mod:lin:lsp:sec

We now give the obvious extensions of compactness and Löwenheim–Skolem to the case of abstract logics.

Definition 28.5. An abstract logic $\langle L, \models_L\rangle$ has the *Compactness Property* if each set Γ of $L(\mathcal{L})$-sentences is satisfiable whenever each finite $\Gamma_0 \subseteq \Gamma$ is satisfiable.

Definition 28.6. $\langle L, \models_L\rangle$ has the *Downward Löwenheim–Skolem property* if any satisfiable Γ has an enumerable model.

The notion of partial isomorphism from Definition 25.15 is purely "algebraic" (i.e., given without reference to the sentences of the language but only to the constants provided by the language $\mathcal{L}$ of the structures), and hence it

applies to the case of abstract logics. In case of first-order logic, we know from Theorem 25.17 that if two structures are partially isomorphic then they are elementarily equivalent. That proof does not carry over to abstract logics, for induction on formulas need not be available for arbitrary $\alpha \in L(\mathcal{L})$, but the theorem is true nonetheless, provided the Löwenheim–Skolem property holds.

mod:lin:lsp: thm:abstract-p-isom

Theorem 28.7. *Suppose $\langle L, \models_L \rangle$ is a normal logic with the Löwenheim–Skolem property. Then any two structures that are partially isomorphic are elementarily equivalent in $\langle L, \models_L \rangle$.*

Proof. Suppose $\mathfrak{M} \simeq_p \mathfrak{N}$, but for some α also $\mathfrak{M} \models_L \alpha$ while $\mathfrak{N} \not\models_L \alpha$. By the Isomorphism Property we can assume that $|\mathfrak{M}|$ and $|\mathfrak{N}|$ are disjoint, and by the Expansion Property we can assume that $\alpha \in L(\mathcal{L})$ for a finite language $\mathcal{L}$. Let $\mathcal{I}$ be a set of partial isomorphisms between $\mathfrak{M}$ and $\mathfrak{N}$, and with no loss of generality also assume that if $p \in \mathcal{I}$ and $q \subseteq p$ then also $q \in \mathcal{I}$.

$|\mathfrak{M}|^{<\omega}$ is the set of finite sequences of elements of $|\mathfrak{M}|$. Let S be the ternary relation over $|\mathfrak{M}|^{<\omega}$ representing concatenation, i.e., if $\mathbf{a}, \mathbf{b}, \mathbf{c} \in |\mathfrak{M}|^{<\omega}$ then $S(\mathbf{a}, \mathbf{b}, \mathbf{c})$ holds if and only if $\mathbf{c}$ is the concatenation of $\mathbf{a}$ and $\mathbf{b}$; and let T be the ternary relation such that $T(\mathbf{a}, b, \mathbf{c})$ holds for $b \in M$ and $\mathbf{a}, \mathbf{c} \in |\mathfrak{M}|^{<\omega}$ if and only if $\mathbf{a} = a_1, \dots a_n$ and $\mathbf{c} = a_1, \dots a_n, b$. Pick new 3-place predicate symbols P and Q and form the structure $\mathfrak{M}^*$ having the universe $|\mathfrak{M}| \cup |\mathfrak{M}|^{<\omega}$, having $\mathfrak{M}$ as a substructure, and interpreting P and Q by the concatenation relations S and T (so $\mathfrak{M}^*$ is in the language $\mathcal{L} \cup \{P, Q\}$).

Define $|\mathfrak{N}|^{<\omega}$, S', T', P', Q' and $\mathfrak{N}^*$ analogously. Since by hypothesis $\mathfrak{M} \simeq_p \mathfrak{N}$, there is a relation I between $|\mathfrak{M}|^{<\omega}$ and $|\mathfrak{N}|^{<\omega}$ such that $I(\mathbf{a}, \mathbf{b})$ holds if and only if $\mathbf{a}$ and $\mathbf{b}$ are isomorphic and satisfy the back-and-forth condition of Definition 25.15. Now, let $\mathfrak{M}$ be the structure whose domain is the union of the domains of $\mathfrak{M}^*$ and $\mathfrak{N}^*$, having $\mathfrak{M}^*$ and $\mathfrak{N}^*$ as substructures, in the language with one extra binary predicate symbol R interpreted by the relation I and predicate symbols denoting the domains $|\mathfrak{M}|^*$ and $|\mathfrak{N}|*$.

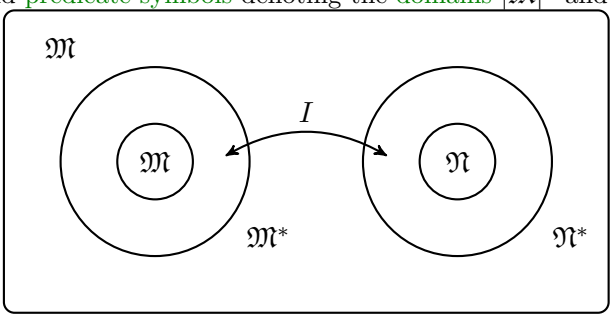


Figure 28.1: The structure $\mathfrak{M}$ with the internal partial isomorphism.

The crucial observation is that in the language of the structure $\mathfrak{M}$ there is a *first-order* sentence θ_1 true in $\mathfrak{M}$ saying that $\mathfrak{M} \models_L \alpha$ and $\mathfrak{N} \not\models_L \alpha$ (this requires the Relativization Property), as well as a *first-order* sentence θ_2 true in $\mathfrak{M}$ saying that $\mathfrak{M} \simeq_p \mathfrak{N}$ via the partial isomorphism I. By the Löwenheim–Skolem Property, θ_1 and θ_2 are jointly true in an enumerable model $\mathfrak{M}_0$ con-

taining partially isomorphic substructures $\mathfrak{M}_0$ and $\mathfrak{N}_0$ such that $\mathfrak{M}_0 \models_L \alpha$ and $\mathfrak{N}_0 \not\models_L \alpha$. But enumerable partially isomorphic structures are in fact isomorphic by Theorem 25.16, contradicting the Isomorphism Property of normal abstract logics. □

---

content/model-theory/lindstrom/lindstrom-proof.tex

28.4 Lindström's Theorem

mod:lin:prf: sec
mod:lin:prf: lem:lindstrom

Lemma 28.8. *Suppose $\alpha \in L(\mathcal{L})$, with $\mathcal{L}$ finite, and assume also that there is an $n \in \mathbb{N}$ such that for any two structures $\mathfrak{M}$ and $\mathfrak{N}$, if $\mathfrak{M} \equiv_n \mathfrak{N}$ and $\mathfrak{M} \models_L \alpha$ then also $\mathfrak{N} \models_L \alpha$. Then α is equivalent to a first-order sentence, i.e., there is a first-order θ such that $\mathrm{Mod}_L(\alpha) = \mathrm{Mod}_L(\theta)$.*

Proof. Let n be such that any two n-equivalent structures $\mathfrak{M}$ and $\mathfrak{N}$ agree on the value assigned to α. Recall Proposition 25.19: there are only finitely many first-order sentences in a finite language that have quantifier rank no greater than n, up to logical equivalence. Now, for each fixed structure $\mathfrak{M}$ let $\theta_{\mathfrak{M}}$ be the conjunction of all first-order sentences α true in $\mathfrak{M}$ with $\mathrm{qr}(\alpha) \leq n$ (this conjunction is finite), so that $\mathfrak{N} \models \theta_{\mathfrak{M}}$ if and only if $\mathfrak{N} \equiv_n \mathfrak{M}$. Then put $\theta = \bigvee\{\theta_{\mathfrak{M}} : \mathfrak{M} \models_L \alpha\}$; this disjunction is also finite (up to logical equivalence).

The conclusion $\mathrm{Mod}_L(\alpha) = \mathrm{Mod}_L(\theta)$ follows. In fact, if $\mathfrak{N} \models_L \theta$ then for some $\mathfrak{M} \models_L \alpha$ we have $\mathfrak{N} \models \theta_{\mathfrak{M}}$, whence also $\mathfrak{N} \models_L \alpha$ (by the hypothesis of the lemma). Conversely, if $\mathfrak{N} \models_L \alpha$ then $\theta_{\mathfrak{N}}$ is a disjunct in θ, and since $\mathfrak{N} \models \theta_{\mathfrak{N}}$, also $\mathfrak{N} \models_L \theta$. □

mod:lin:prf: thm:lindstrom

Theorem 28.9 (Lindström's Theorem). *Suppose $\langle L, \models_L\rangle$ has the Compactness and the Löwenheim–Skolem Properties. Then $\langle L, \models_L\rangle \leq \langle F, \models\rangle$ (so $\langle L, \models_L\rangle$ is equivalent to first-order logic).*

Proof. By Lemma 28.8, it suffices to show that for any $\alpha \in L(\mathcal{L})$, with $\mathcal{L}$ finite, there is $n \in \mathbb{N}$ such that for any two structures $\mathfrak{M}$ and $\mathfrak{N}$: if $\mathfrak{M} \equiv_n \mathfrak{N}$ then $\mathfrak{M}$ and $\mathfrak{N}$ agree on α. For then α is equivalent to a first-order sentence, from which $\langle L, \models_L\rangle \leq \langle F, \models\rangle$ follows. Since we are working in a finite, purely relational language, by Theorem 25.23 we can replace the statement that $\mathfrak{M} \equiv_n \mathfrak{N}$ by the corresponding algebraic statement that $I_n(\emptyset, \emptyset)$.

Given α, suppose towards a contradiction that for each n there are structures $\mathfrak{M}_n$ and $\mathfrak{N}_n$ such that $I_n(\emptyset, \emptyset)$, but (say) $\mathfrak{M}_n \models_L \alpha$ whereas $\mathfrak{N}_n \not\models_L \alpha$. By the Isomorphism Property we can assume that all the $\mathfrak{M}_n$'s interpret the constants of the language by the same objects; furthermore, since there are only finitely many atomic sentences in the language, we may also assume that they satisfy the same atomic sentences (we can take a subsequence of the $\mathfrak{M}$'s otherwise). Let $\mathfrak{M}$ be the union of all the $\mathfrak{M}_n$'s, i.e., the unique minimal structure having each $\mathfrak{M}_n$ as a substructure. As in the proof of Theorem 28.7, let

$\mathfrak{M}^*$ be the extension of $\mathfrak{M}$ with domain $|\mathfrak{M}| \cup |\mathfrak{M}|^{<\omega}$, in the expanded language comprising the concatenation predicates P and Q.

Similarly, define $\mathfrak{N}_n$, $\mathfrak{N}$ and $\mathfrak{N}^*$. Now let $\mathfrak{M}$ be the structure whose domain comprises the domains of $\mathfrak{M}^*$ and $\mathfrak{N}^*$ as well as the natural numbers $\mathbb{N}$ along with their natural ordering $\leq$, in the language with extra predicates representing the domains $|\mathfrak{M}|$, $|\mathfrak{N}|$, $|\mathfrak{M}|^{<\omega}$ and $|\mathfrak{N}|^{<\omega}$ as well as predicates coding the domains of $\mathfrak{M}_n$ and $\mathfrak{N}_n$ in the sense that:

$$|\mathfrak{M}_n| = \{a \in |\mathfrak{M}| : R(a,n)\}; \qquad |\mathfrak{N}_n| = \{a \in |\mathfrak{N}| : S(a,n)\};$$
$$|\mathfrak{M}|_n^{<\omega} = \{a \in |\mathfrak{M}|^{<\omega} : R(a,n)\}; \qquad |\mathfrak{N}|_n^{<\omega} = \{a \in |\mathfrak{N}|^{<\omega} : S(a,n)\}.$$

The structure $\mathfrak{M}$ also has a ternary relation J such that $J(n, \mathbf{a}, \mathbf{b})$ holds if and only if $I_n(\mathbf{a}, \mathbf{b})$.

Now there is a sentence θ in the language $\mathcal{L}$ augmented by R, S, J, etc., saying that $\leq$ is a discrete linear ordering with first but no last element and such that $\mathfrak{M}_n \models \alpha$, $\mathfrak{N}_n \not\models \alpha$, and for each n in the ordering, $J(n, \mathbf{a}, \mathbf{b})$ holds if and only if $I_n(\mathbf{a}, \mathbf{b})$.

Using the Compactness Property, we can find a model $\mathfrak{M}^*$ of θ in which the ordering contains a non-standard element n^*. In particular then $\mathfrak{M}^*$ will contain substructures $\mathfrak{M}_{n^*}$ and $\mathfrak{N}_{n^*}$ such that $\mathfrak{M}_{n^*} \models_L \alpha$ and $\mathfrak{N}_{n^*} \not\models_L \alpha$. But now we can define a set $\mathcal{I}$ of pairs of k-tuples from $|\mathfrak{M}_{n^*}|$ and $|\mathfrak{N}_{n^*}|$ by putting $\langle \mathbf{a}, \mathbf{b} \rangle \in \mathcal{I}$ if and only if $J(n^* - k, \mathbf{a}, \mathbf{b})$, where k is the length of $\mathbf{a}$ and $\mathbf{b}$. Since n^* is non-standard, for each standard k we have that $n^* - k > 0$, and the set $\mathcal{I}$ witnesses the fact that $\mathfrak{M}_{n^*} \simeq_p \mathfrak{N}_{n^*}$. But by Theorem 28.7, $\mathfrak{M}_{n^*}$ is L-equivalent to $\mathfrak{N}_{n^*}$, a contradiction. □

Part V

Computability

This part is based on Jeremy Avigad's notes on computability theory. Only the chapter on recursive functions contains exercises yet, and everything could stand to be expanded with motivation, examples, details, and exercises.

Chapter 29

Recursive Functions

These are Jeremy Avigad's notes on recursive functions, revised and expanded by Richard Zach. This chapter does contain some exercises, and can be included independently to provide the basis for a discussion of arithmetization of syntax.

---

content/computability/recursive-functions/introduction.tex

29.1 Introduction

cmp:rec:int:sec

In order to develop a mathematical theory of computability, one has to, first of all, develop a *model* of computability. We now think of computability as the kind of thing that computers do, and computers work with symbols. But at the beginning of the development of theories of computability, the paradigmatic example of computation was *numerical* computation. Mathematicians were always interested in number-theoretic functions, i.e., functions $f\colon \mathbb{N}^n \to \mathbb{N}$ that can be computed. So it is not surprising that at the beginning of the theory of computability, it was such functions that were studied. The most familiar

examples of computable numerical functions, such as addition, multiplication, exponentiation (of natural numbers) share an interesting feature: they can be defined *recursively*. It is thus quite natural to attempt a general definition of *computable function* on the basis of recursive definitions. Among the many possible ways to define number-theoretic functions recursively, one particularly simple pattern of definition here becomes central: so-called *primitive recursion*.

In addition to computable functions, we might be interested in computable sets and relations. A set is computable if we can compute the answer to whether or not a given number is an element of the set, and a relation is computable iff we can compute whether or not a tuple $\langle n_1, \ldots, n_k \rangle$ is an element of the relation. By considering the *characteristic function* of a set or relation, discussion of computable sets and relations can be subsumed under that of computable functions. Thus we can define primitive recursive relations as well, e.g., the relation "n evenly divides m" is a primitive recursive relation.

Primitive recursive functions—those that can be defined using just primitive recursion—are not, however, the only computable number-theoretic functions. Many generalizations of primitive recursion have been considered, but the most powerful and widely-accepted additional way of computing functions is by unbounded search. This leads to the definition of *partial recursive functions*, and a related definition to *general recursive functions*. General recursive functions are computable and total, and the definition characterizes exactly the partial recursive functions that happen to be total. Recursive functions can simulate every other model of computation (Turing machines, lambda calculus, etc.) and so represent one of the many accepted models of computation.

---

content/computability/recursive-functions/primitive-recursion.tex

29.2 Primitive Recursion

cmp:rec:pre:sec

A characteristic of the natural numbers is that every natural number can be reached from 0 by applying the successor operation $+1$ finitely many times—any natural number is either 0 or the successor of ... the successor of 0. One way to specify a function $h\colon \mathbb{N} \to \mathbb{N}$ that makes use of this fact is this: (a) specify what the value of h is for argument 0, and (b) also specify how to, given the value of $h(x)$, compute the value of $h(x+1)$. For (a) tells us directly what $h(0)$ is, so h is defined for 0. Now, using the instruction given by (b) for $x = 0$, we can compute $h(1) = h(0+1)$ from $h(0)$. Using the same instructions for $x = 1$, we compute $h(2) = h(1+1)$ from $h(1)$, and so on. For every natural number x, we'll eventually reach the step where we define $h(x)$ from $h(x+1)$, and so $h(x)$ is defined for all $x \in \mathbb{N}$.

For instance, suppose we specify $h\colon \mathbb{N} \to \mathbb{N}$ by the following two equations:

$$h(0) = 1$$
$$h(x+1) = 2 \cdot h(x)$$

If we already know how to multiply, then these equations give us the information required for (a) and (b) above. By successively applying the second equation, we get that

$$\begin{aligned} h(1) &= 2 \cdot h(0) = 2, \\ h(2) &= 2 \cdot h(1) = 2 \cdot 2, \\ h(3) &= 2 \cdot h(2) = 2 \cdot 2 \cdot 2, \\ &\vdots \end{aligned}$$

We see that the function h we have specified is $h(x) = 2^x$.

The characteristic feature of the natural numbers guarantees that there is only one function h that meets these two criteria. A pair of equations like these is called a *definition by primitive recursion* of the function h. It is so-called because we define h "recursively," i.e., the definition, specifically the second equation, involves h itself on the right-hand-side. It is "primitive" because in defining $h(x+1)$ we only use the value $h(x)$, i.e., the immediately preceding value. This is the simplest way of defining a function on $\mathbb{N}$ recursively.

We can define even more fundamental functions like addition and multiplication by primitive recursion. In these cases, however, the functions in question are 2-place. We fix one of the argument places, and use the other for the recursion. E.g, to define $\text{add}(x, y)$ we can fix x and define the value first for $y = 0$ and then for $y + 1$ in terms of y. Since x is fixed, it will appear on the left and on the right side of the defining equations.

$$\begin{aligned} \text{add}(x, 0) &= x \\ \text{add}(x, y + 1) &= \text{add}(x, y) + 1 \end{aligned}$$

These equations specify the value of add for all x *and* y. To find $\text{add}(2, 3)$, for instance, we apply the defining equations for $x = 2$, using the first to find $\text{add}(2, 0) = 2$, then using the second to successively find $\text{add}(2, 1) = 2 + 1 = 3$, $\text{add}(2, 2) = 3 + 1 = 4$, $\text{add}(2, 3) = 4 + 1 = 5$.

In the definition of add we used $+$ on the right-hand-side of the second equation, but only to add 1. In other words, we used the successor function $\text{succ}(z) = z+1$ and applied it to the previous value $\text{add}(x, y)$ to define $\text{add}(x, y+1)$. So we can think of the recursive definition as given in terms of a single function which we apply to the previous value. However, it doesn't hurt—and sometimes is necessary—to allow the function to depend not just on the previous value but also on x and y. Consider:

$$\begin{aligned} \text{mult}(x, 0) &= 0 \\ \text{mult}(x, y + 1) &= \text{add}(\text{mult}(x, y), x) \end{aligned}$$

This is a primitive recursive definition of a function mult by applying the function add to both the preceding value $\text{mult}(x, y)$ and the first argument x. It also defines the function $\text{mult}(x, y)$ for all arguments x and y. For instance,

mult$(2,3)$ is determined by successively computing mult$(2,0)$, mult$(2,1)$, mult$(2,2)$, and mult$(2,3)$:

$$\begin{aligned}
&\text{mult}(2,0) = 0\\
&\text{mult}(2,1) = \text{mult}(2,0+1) = \text{add}(\text{mult}(2,0),2) = \text{add}(0,2) = 2\\
&\text{mult}(2,2) = \text{mult}(2,1+1) = \text{add}(\text{mult}(2,1),2) = \text{add}(2,2) = 4\\
&\text{mult}(2,3) = \text{mult}(2,2+1) = \text{add}(\text{mult}(2,2),2) = \text{add}(4,2) = 6
\end{aligned}$$

The general pattern then is this: to give a primitive recursive definition of a function $h(x_0,\ldots,x_{k-1},y)$, we provide two equations. The first defines the value of $h(x_0,\ldots,x_{k-1},0)$ without reference to h. The second defines the value of $h(x_0,\ldots,x_{k-1},y+1)$ in terms of $h(x_0,\ldots,x_{k-1},y)$, the other arguments x_0, $\ldots$, x_{k-1}, and y. Only the immediately preceding value of h may be used in that second equation. If we think of the operations given by the right-hand-sides of these two equations as themselves being functions f and g, then the general pattern to define a new function h by primitive recursion is this:

$$\begin{aligned}
h(x_0,\ldots,x_{k-1},0) &= f(x_0,\ldots,x_{k-1})\\
h(x_0,\ldots,x_{k-1},y+1) &= g(x_0,\ldots,x_{k-1},y,h(x_0,\ldots,x_{k-1},y))
\end{aligned}$$

In the case of add, we have $k=1$ and $f(x_0)=x_0$ (the identity function), and $g(x_0,y,z)=z+1$ (the 3-place function that returns the successor of its third argument):

$$\begin{aligned}
\text{add}(x_0,0) &= f(x_0) = x_0\\
\text{add}(x_0,y+1) &= g(x_0,y,\text{add}(x_0,y)) = \text{succ}(\text{add}(x_0,y))
\end{aligned}$$

In the case of mult, we have $f(x_0)=0$ (the constant function always returning 0) and $g(x_0,y,z)=\text{add}(z,x_0)$ (the 3-place function that returns the sum of its last and first argument):

$$\begin{aligned}
\text{mult}(x_0,0) &= f(x_0) = 0\\
\text{mult}(x_0,y+1) &= g(x_0,y,\text{mult}(x_0,y)) = \text{add}(\text{mult}(x_0,y),x_0)
\end{aligned}$$

---

content/computability/recursive-functions/composition.tex

29.3 Composition

cmp:rec:com:sec

If f and g are two one-place functions of natural numbers, we can compose them: $h(x)=f(g(x))$. The new function $h(x)$ is then defined by *composition* from the functions f and g. We'd like to generalize this to functions of more than one argument.

Here's one way of doing this: suppose f is a k-place function, and g_0, $\ldots$, g_{k-1} are k functions which are all n-place. Then we can define a new n-place function h as follows:

$$h(x_0,\ldots,x_{n-1}) = f(g_0(x_0,\ldots,x_{n-1}),\ldots,g_{k-1}(x_0,\ldots,x_{n-1}))$$

If f and all g_i are computable, so is h: To compute $h(x_0, \ldots, x_{n-1})$, first compute the values $y_i = g_i(x_0, \ldots, x_{n-1})$ for each $i = 0, \ldots, k-1$. Then feed these values into f to compute $h(x_0, \ldots, x_{k-1}) = f(y_0, \ldots, y_{k-1})$.

This may seem like an overly restrictive characterization of what happens when we compute a new function using some existing ones. For one thing, sometimes we do not use all the arguments of a function, as when we defined $g(x, y, z) = \text{succ}(z)$ for use in the primitive recursive definition of add. Suppose we are allowed use of the following functions:

$$P_i^n(x_0, \ldots, x_{n-1}) = x_i$$

The functions P_i^k are called *projection* functions: P_i^n is an n-place function. Then g can be defined by

$$g(x, y, z) = \text{succ}(P_2^3(x, y, z)).$$

Here the role of f is played by the 1-place function succ, so $k = 1$. And we have one 3-place function P_2^3 which plays the role of g_0. The result is a 3-place function that returns the successor of the third argument.

The projection functions also allow us to define new functions by reordering or identifying arguments. For instance, the function $h(x) = \text{add}(x, x)$ can be defined by

$$h(x_0) = \text{add}(P_0^1(x_0), P_0^1(x_0)).$$

Here $k = 2$, $n = 1$, the role of $f(y_0, y_1)$ is played by add, and the roles of $g_0(x_0)$ and $g_1(x_0)$ are both played by $P_0^1(x_0)$, the one-place projection function (aka the identity function).

If $f(y_0, y_1)$ is a function we already have, we can define the function $h(x_0, x_1) = f(x_1, x_0)$ by

$$h(x_0, x_1) = f(P_1^2(x_0, x_1), P_0^2(x_0, x_1)).$$

Here $k = 2$, $n = 2$, and the roles of g_0 and g_1 are played by P_1^2 and P_0^2, respectively.

You may also worry that $g_0, \ldots, g_{k-1}$ are all required to have the same arity n. (Remember that the *arity* of a function is the number of arguments; an n-place function has arity n.) But adding the projection functions provides the desired flexibility. For example, suppose f and g are 3-place functions and h is the 2-place function defined by

$$h(x, y) = f(x, g(x, x, y), y).$$

The definition of h can be rewritten with the projection functions, as

$$h(x, y) = f(P_0^2(x, y), g(P_0^2(x, y), P_0^2(x, y), P_1^2(x, y)), P_1^2(x, y)).$$

Then h is the composition of f with P_0^2, l, and P_1^2, where

$$l(x, y) = g(P_0^2(x, y), P_0^2(x, y), P_1^2(x, y)),$$

i.e., l is the composition of g with P_0^2, P_0^2, and P_1^2.

29.4 Primitive Recursion Functions

cmp:rec:prf: sec

Let us record again how we can define new functions from existing ones using primitive recursion and composition.

cmp:rec:prf: defn:primitive-recursion

Definition 29.1. Suppose f is a k-place function ($k \geq 1$) and g is a $(k+2)$-place function. The function defined by *primitive recursion from f and g* is the $(k+1)$-place function h defined by the equations

$$\begin{aligned} h(x_0, \ldots, x_{k-1}, 0) &= f(x_0, \ldots, x_{k-1}) \\ h(x_0, \ldots, x_{k-1}, y+1) &= g(x_0, \ldots, x_{k-1}, y, h(x_0, \ldots, x_{k-1}, y)) \end{aligned}$$

cmp:rec:prf: defn:composition

Definition 29.2. Suppose f is a k-place function, and $g_0, \ldots, g_{k-1}$ are k functions which are all n-place. The function defined by *composition from f and $g_0, \ldots, g_{k-1}$* is the n-place function h defined by

$$h(x_0, \ldots, x_{n-1}) = f(g_0(x_0, \ldots, x_{n-1}), \ldots, g_{k-1}(x_0, \ldots, x_{n-1})).$$

In addition to succ and the projection functions

$$P^n_i(x_0, \ldots, x_{n-1}) = x_i,$$

for each natural number n and $i < n$, we will include among the primitive recursive functions the function $\mathrm{zero}(x) = 0$.

Definition 29.3. The set of primitive recursive functions is the set of functions from $\mathbb{N}^n$ to $\mathbb{N}$, defined inductively by the following clauses:

1. zero is primitive recursive.
2. succ is primitive recursive.
3. Each projection function P^n_i is primitive recursive.
4. If f is a k-place primitive recursive function and $g_0, \ldots, g_{k-1}$ are n-place primitive recursive functions, then the composition of f with $g_0, \ldots, g_{k-1}$ is primitive recursive.
5. If f is a k-place primitive recursive function and g is a $k+2$-place primitive recursive function, then the function defined by primitive recursion from f and g is primitive recursive.

Put more concisely, the set of primitive recursive functions is the smallest set containing zero, succ, and the projection functions P^n_j, and which is closed under composition and primitive recursion. explanation

Another way of describing the set of primitive recursive functions is by defining it in terms of "stages." Let S_0 denote the set of starting functions: zero, succ, and the projections. These are the primitive recursive functions of stage 0. Once a stage S_i has been defined, let S_{i+1} be the set of all functions

you get by applying a single instance of composition or primitive recursion to functions already in S_i. Then

$$S = \bigcup_{i \in \mathbb{N}} S_i$$

is the set of all primitive recursive functions

Let us verify that add is a primitive recursive function.

Proposition 29.4. *The addition function* $\mathrm{add}(x, y) = x + y$ *is primitive recursive.*

Proof. We already have a primitive recursive definition of add in terms of two functions f and g which matches the format of Definition 29.1:

$$\begin{aligned} \mathrm{add}(x_0, 0) &= f(x_0) = x_0 \\ \mathrm{add}(x_0, y+1) &= g(x_0, y, \mathrm{add}(x_0, y)) = \mathrm{succ}(\mathrm{add}(x_0, y)) \end{aligned}$$

So add is primitive recursive provided f and g are as well. $f(x_0) = x_0 = P^1_0(x_0)$, and the projection functions count as primitive recursive, so f is primitive recursive. The function g is the three-place function $g(x_0, y, z)$ defined by

$$g(x_0, y, z) = \mathrm{succ}(z).$$

This does not yet tell us that g is primitive recursive, since g and succ are not quite the same function: succ is one-place, and g has to be three-place. But we can define g "officially" by composition as

$$g(x_0, y, z) = \mathrm{succ}(P^3_2(x_0, y, z))$$

Since succ and P^3_2 count as primitive recursive functions, g does as well, since it can be defined by composition from primitive recursive functions. □

Proposition 29.5. *The multiplication function* $\mathrm{mult}(x, y) = x \cdot y$ *is primitive recursive.* *cmp:rec:prf: prop:mult-pr*

Proof. Exercise. □

Problem 29.1. Prove Proposition 29.5 by showing that the primitive recursive definition of mult can be put into the form required by Definition 29.1 and showing that the corresponding functions f and g are primitive recursive.

Example 29.6. Here's our very first example of a primitive recursive definition:

$$\begin{aligned} h(0) &= 1 \\ h(y+1) &= 2 \cdot h(y). \end{aligned}$$

This function cannot fit into the form required by Definition 29.1, since $k = 0$. The definition also involves the constants 1 and 2. To get around the first problem, let's introduce a dummy argument and define the function h':

$$h'(x_0, 0) = f(x_0) = 1$$
$$h'(x_0, y + 1) = g(x_0, y, h'(x_0, y)) = 2 \cdot h'(x_0, y).$$

The function $f(x_0) = 1$ can be defined from succ and zero by composition: $f(x_0) = \text{succ}(\text{zero}(x_0))$. The function g can be defined by composition from $g'(z) = 2 \cdot z$ and projections:

$$g(x_0, y, z) = g'(P^3_2(x_0, y, z))$$

and g' in turn can be defined by composition as

$$g'(z) = \text{mult}(g''(z), P^1_0(z))$$

and

$$g''(z) = \text{succ}(f(z)),$$

where f is as above: $f(z) = \text{succ}(\text{zero}(z))$. Now that we have h', we can use composition again to let $h(y) = h'(P^1_0(y), P^1_0(y))$. This shows that h can be defined from the basic functions using a sequence of compositions and primitive recursions, so h is primitive recursive.

---

content/computability/recursive-functions/notation-pr-functions.tex

29.5 Primitive Recursion Notations

cmp:rec:not:sec

One advantage to having the precise inductive description of the primitive recursive functions is that we can be systematic in describing them. For example, we can assign a "notation" to each such function, as follows. Use symbols zero, succ, and P^n_i for zero, successor, and the projections. Now suppose h is defined by composition from a k-place function f and n-place functions g_0, $\ldots$, g_{k-1}, and we have assigned notations F, G_0, $\ldots$, G_{k-1} to the latter functions. Then, using a new symbol $\text{Comp}_{k,n}$, we can denote the function h by $\text{Comp}_{k,n}[F, G_0, \ldots, G_{k-1}]$.

For functions defined by primitive recursion, we can use analogous notations. Suppose the $(k + 1)$-ary function h is defined by primitive recursion from the k-ary function f and the $(k + 2)$-ary function g, and the notations assigned to f and g are F and G, respectively. Then the notation assigned to h is $\text{Rec}_k[F, G]$.

Recall that the addition function is defined by primitive recursion as

$$\text{add}(x_0, 0) = P^1_0(x_0) = x_0$$
$$\text{add}(x_0, y + 1) = \text{succ}(P^3_2(x_0, y, \text{add}(x_0, y))) = \text{add}(x_0, y) + 1$$

Here the role of f is played by P^1_0, and the role of g is played by $\text{succ}(P^3_2(x_0, y, z))$, which is assigned the notation $\text{Comp}_{1,3}[\text{succ}, P^3_2]$ as it is the result of defining a function by composition from the 1-ary function succ and the 3-ary function P^3_2. With this setup, we can denote the addition function by

$$\text{Rec}_1[P^1_0, \text{Comp}_{1,3}[\text{succ}, P^3_2]].$$

Having these notations sometimes proves useful, e.g., when enumerating primitive recursive functions.

Problem 29.2. Give the complete primitive recursive notation for mult.

---

content/computability/recursive-functions/pr-functions-computable.tex

29.6 Primitive Recursive Functions are Computable

cmp:rec:cmp:sec

Suppose a function h is defined by primitive recursion

$$\begin{aligned} h(\vec{x}, 0) &= f(\vec{x}) \\ h(\vec{x}, y+1) &= g(\vec{x}, y, h(\vec{x}, y)) \end{aligned}$$

and suppose the functions f and g are computable. (We use $\vec{x}$ to abbreviate x_0, $\ldots$, x_{k-1}.) Then $h(\vec{x}, 0)$ can obviously be computed, since it is just $f(\vec{x})$ which we assume is computable. $h(\vec{x}, 1)$ can then also be computed, since $1 = 0 + 1$ and so $h(\vec{x}, 1)$ is just

$$h(\vec{x}, 1) = g(\vec{x}, 0, h(\vec{x}, 0)) = g(\vec{x}, 0, f(\vec{x})).$$

We can go on in this way and compute

$$\begin{aligned} h(\vec{x}, 2) &= g(\vec{x}, 1, h(\vec{x}, 1)) = g(\vec{x}, 1, g(\vec{x}, 0, f(\vec{x}))) \\ h(\vec{x}, 3) &= g(\vec{x}, 2, h(\vec{x}, 2)) = g(\vec{x}, 2, g(\vec{x}, 1, g(\vec{x}, 0, f(\vec{x})))) \\ h(\vec{x}, 4) &= g(\vec{x}, 3, h(\vec{x}, 3)) = g(\vec{x}, 3, g(\vec{x}, 2, g(\vec{x}, 1, g(\vec{x}, 0, f(\vec{x}))))) \\ &\vdots \end{aligned}$$

Thus, to compute $h(\vec{x}, y)$ in general, successively compute $h(\vec{x}, 0)$, $h(\vec{x}, 1)$, $\ldots$, until we reach $h(\vec{x}, y)$.

Thus, a primitive recursive definition yields a new computable function if the functions f and g are computable. Composition of functions also results in a computable function if the functions f and g_i are computable.

Since the basic functions zero, succ, and P^n_i are computable, and composition and primitive recursion yield computable functions from computable functions, this means that every primitive recursive function is computable.

---

content/computability/recursive-functions/examples.tex

29.7 Examples of Primitive Recursive Functions

cmp:rec:exa:sec

We already have some examples of primitive recursive functions: the addition and multiplication functions add and mult. The identity function $\mathrm{id}(x) = x$ is primitive recursive, since it is just P^1_0. The constant functions $\mathrm{const}_n(x) = n$ are primitive recursive since they can be defined from zero and succ by successive composition. This is useful when we want to use constants in primitive recursive definitions, e.g., if we want to define the function $f(x) = 2 \cdot x$ can obtain it by composition from $\mathrm{const}_n(x)$ and multiplication as $f(x) = \mathrm{mult}(\mathrm{const}_2(x), P^1_0(x))$. We'll make use of this trick from now on.

Proposition 29.7. *The exponentiation function* $\exp(x, y) = x^y$ *is primitive recursive.*

Proof. We can define exp primitive recursively as

$$\begin{aligned} \exp(x, 0) &= 1 \\ \exp(x, y+1) &= \mathrm{mult}(x, \exp(x, y)). \end{aligned}$$

Strictly speaking, this is not a recursive definition from primitive recursive functions. Officially, though, we have:

$$\begin{aligned} \exp(x, 0) &= f(x) \\ \exp(x, y+1) &= g(x, y, \exp(x, y)). \end{aligned}$$

where

$$\begin{aligned} f(x) &= \mathrm{succ}(\mathrm{zero}(x)) = 1 \\ g(x, y, z) &= \mathrm{mult}(P^3_0(x, y, z), P^3_2(x, y, z)) = x \cdot z \end{aligned}$$

and so f and g are defined from primitive recursive functions by composition. □

Proposition 29.8. *The predecessor function* $\mathrm{pred}(y)$ *defined by*

$$\mathrm{pred}(y) = \begin{cases} 0 & \text{if } y = 0 \\ y - 1 & \text{otherwise} \end{cases}$$

is primitive recursive.

Proof. Note that

$$\begin{aligned} \mathrm{pred}(0) &= 0 \text{ and} \\ \mathrm{pred}(y+1) &= y. \end{aligned}$$

This is almost a primitive recursive definition. It does not, strictly speaking, fit into the pattern of definition by primitive recursion, since that pattern requires

at least one extra argument x. It is also odd in that it does not actually use $\text{pred}(y)$ in the definition of $\text{pred}(y+1)$. But we can first define $\text{pred}'(x,y)$ by

$$\begin{aligned}\text{pred}'(x,0) &= \text{zero}(x) = 0,\\ \text{pred}'(x,y+1) &= P^3_1(x,y,\text{pred}'(x,y)) = y.\end{aligned}$$

and then define pred from it by composition, e.g., as $\text{pred}(x) = \text{pred}'(\text{zero}(x), P^1_0(x))$.□

Proposition 29.9. *The factorial function* $\text{fac}(x) = x! = 1 \cdot 2 \cdot 3 \cdot \cdots \cdot x$ *is primitive recursive.*

Proof. The obvious primitive recursive definition is

$$\begin{aligned}\text{fac}(0) &= 1\\ \text{fac}(y+1) &= \text{fac}(y) \cdot (y+1).\end{aligned}$$

Officially, we have to first define a two-place function h

$$\begin{aligned}h(x,0) &= \text{const}_1(x)\\ h(x,y+1) &= g(x,y,h(x,y))\end{aligned}$$

where $g(x,y,z) = \text{mult}(P^3_2(x,y,z), \text{succ}(P^3_1(x,y,z)))$ and then let

$$\text{fac}(y) = h(P^1_0(y), P^1_0(y)) = h(y,y).$$

From now on we'll be a bit more laissez-faire and not give the official definitions by composition and primitive recursion. □

Proposition 29.10. *Truncated subtraction,* $x \mathbin{\dot{-}} y$*, defined by*

$$x \mathbin{\dot{-}} y = \begin{cases} 0 & \text{if } x < y\\ x - y & \text{otherwise}\end{cases}$$

is primitive recursive.

Proof. We have:

$$\begin{aligned}x \mathbin{\dot{-}} 0 &= x\\ x \mathbin{\dot{-}} (y+1) &= \text{pred}(x \mathbin{\dot{-}} y)\end{aligned}$$ □

Proposition 29.11. *The distance between* x *and* y*,* $|x-y|$*, is primitive recursive.*

Proof. We have $|x-y| = (x \mathbin{\dot{-}} y) + (y \mathbin{\dot{-}} x)$, so the distance can be defined by composition from $+$ and $\mathbin{\dot{-}}$, which are primitive recursive. □

Proposition 29.12. *The maximum of x and y, $\max(x, y)$, is primitive recursive.*

Proof. We can define $\max(x, y)$ by composition from $+$ and $\dot{-}$ by

$$\max(x, y) = x + (y \dot{-} x).$$

If x is the maximum, i.e., $x \geq y$, then $y \dot{-} x = 0$, so $x + (y \dot{-} x) = x + 0 = x$. If y is the maximum, then $y \dot{-} x = y - x$, and so $x + (y \dot{-} x) = x + (y - x) = y$. □

cmp:rec:exa: prop:min-pr

Proposition 29.13. *The minimum of x and y, $\min(x, y)$, is primitive recursive.*

Proof. Exercise. □

Problem 29.3. Prove Proposition 29.13.

Problem 29.4. Show that

$$f(x, y) = 2^{(2^{\cdot^{\cdot^{\cdot^{2^x}}}})} \Big\} y \text{ 2's}$$

is primitive recursive.

Problem 29.5. Show that integer division $d(x, y) = \lfloor x/y \rfloor$ (i.e., division, where you disregard everything after the decimal point) is primitive recursive. When $y = 0$, we stipulate $d(x, y) = 0$. Give an explicit definition of d using primitive recursion and composition.

Proposition 29.14. *The set of primitive recursive functions is closed under the following two operations:*

1. *Finite sums: if $f(\vec{x}, z)$ is primitive recursive, then so is the function*

$$g(\vec{x}, y) = \sum_{z=0}^{y} f(\vec{x}, z).$$

2. *Finite products: if $f(\vec{x}, z)$ is primitive recursive, then so is the function*

$$h(\vec{x}, y) = \prod_{z=0}^{y} f(\vec{x}, z).$$

Proof. For example, finite sums are defined recursively by the equations

$$\begin{aligned} g(\vec{x}, 0) &= f(\vec{x}, 0) \\ g(\vec{x}, y + 1) &= g(\vec{x}, y) + f(\vec{x}, y + 1). \end{aligned}$$

□

29.8 Primitive Recursive Relations

cmp:rec:prr:sec

Definition 29.15. A relation $R(\vec{x})$ is said to be primitive recursive if its characteristic function,

$$\chi_R(\vec{x}) = \begin{cases} 1 & \text{if } R(\vec{x}) \\ 0 & \text{otherwise} \end{cases}$$

is primitive recursive.

In other words, when one speaks of a primitive recursive relation $R(\vec{x})$, one is referring to a relation of the form $\chi_R(\vec{x}) = 1$, where χ_R is a primitive recursive function which, on any input, returns either 1 or 0. For example, the relation IsZero(x), which holds if and only if $x = 0$, corresponds to the function χ_{IsZero}, defined using primitive recursion by

$$\chi_{\text{IsZero}}(0) = 1,$$
$$\chi_{\text{IsZero}}(x+1) = 0.$$

It should be clear that one can compose relations with other primitive recursive functions. So the following are also primitive recursive:

1. The equality relation, $x = y$, defined by IsZero$(|x - y|)$
2. The less-than relation, $x \leq y$, defined by IsZero$(x \dot{-} y)$

Proposition 29.16. *The set of primitive recursive relations is closed under Boolean operations, that is, if $P(\vec{x})$ and $Q(\vec{x})$ are primitive recursive, so are*

1. $\neg P(\vec{x})$
2. $P(\vec{x}) \wedge Q(\vec{x})$
3. $P(\vec{x}) \vee Q(\vec{x})$
4. $P(\vec{x}) \to Q(\vec{x})$

Proof. Suppose $P(\vec{x})$ and $Q(\vec{x})$ are primitive recursive, i.e., their characteristic functions χ_P and χ_Q are. We have to show that the characteristic functions of $\neg P(\vec{x})$, etc., are also primitive recursive.

$$\chi_{\neg P}(\vec{x}) = \begin{cases} 0 & \text{if } \chi_P(\vec{x}) = 1 \\ 1 & \text{otherwise} \end{cases}$$

We can define $\chi_{\neg P}(\vec{x})$ as $1 \dot{-} \chi_P(\vec{x})$.

$$\chi_{P \wedge Q}(\vec{x}) = \begin{cases} 1 & \text{if } \chi_P(\vec{x}) = \chi_Q(\vec{x}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

We can define $\chi_{P \wedge Q}(\vec{x})$ as $\chi_P(\vec{x}) \cdot \chi_Q(\vec{x})$ or as $\min(\chi_P(\vec{x}), \chi_Q(\vec{x}))$. Similarly,

$$\chi_{P \vee Q}(\vec{x}) = \max(\chi_P(\vec{x}), \chi_Q(\vec{x})) \text{ and}$$
$$\chi_{P \to Q}(\vec{x}) = \max(1 \dot{-} \chi_P(\vec{x}), \chi_Q(\vec{x})).$$

□

Proposition 29.17. *The set of primitive recursive relations is closed under bounded quantification, i.e., if $R(\vec{x}, z)$ is a primitive recursive relation, then so are the relations*

$$(\forall z < y)\ R(\vec{x}, z) \text{ and}$$
$$(\exists z < y)\ R(\vec{x}, z).$$

$(\forall z < y)\ R(\vec{x}, z)$ holds of $\vec{x}$ and y if and only if $R(\vec{x}, z)$ holds for every z less than y, and similarly for $(\exists z < y)\ R(\vec{x}, z)$.

Proof. By convention, we take $(\forall z < 0)\ R(\vec{x}, z)$ to be true (for the trivial reason that there are no z less than 0) and $(\exists z < 0)\ R(\vec{x}, z)$ to be false. A bounded universal quantifier functions just like a finite product or iterated minimum, i.e., if $P(\vec{x}, y) \Leftrightarrow (\forall z < y)\ R(\vec{x}, z)$ then $\chi_P(\vec{x}, y)$ can be defined by

$$\chi_P(\vec{x}, 0) = 1$$
$$\chi_P(\vec{x}, y+1) = \min(\chi_P(\vec{x}, y), \chi_R(\vec{x}, y)).$$

Bounded existential quantification can similarly be defined using max. Alternatively, it can be defined from bounded universal quantification, using the equivalence $(\exists z < y)\ R(\vec{x}, z) \leftrightarrow \neg(\forall z < y)\ \neg R(\vec{x}, z)$. Note that, for example, a bounded quantifier of the form $(\exists x \leq y)\ \dots x \dots$ is equivalent to $(\exists x < y + 1)\ \dots x \dots$. □

Problem 29.6. Show that the three place relation $x \equiv y \mod n$ (congruence modulo n) is primitive recursive.

Another useful primitive recursive function is the conditional function, $\mathrm{cond}(x, y, z)$, defined by

$$\mathrm{cond}(x, y, z) = \begin{cases} y & \text{if } x = 0 \\ z & \text{otherwise.} \end{cases}$$

This is defined recursively by

$$\mathrm{cond}(0, y, z) = y,$$
$$\mathrm{cond}(x+1, y, z) = z.$$

One can use this to justify definitions of primitive recursive functions by cases from primitive recursive relations:

Proposition 29.18. *If $g_0(\vec{x})$, ..., $g_m(\vec{x})$ are primitive recursive functions, and $R_0(\vec{x})$, ..., $R_{m-1}(\vec{x})$ are primitive recursive relations, then the function f defined by*

$$f(\vec{x}) = \begin{cases} g_0(\vec{x}) & \text{if } R_0(\vec{x}) \\ g_1(\vec{x}) & \text{if } R_1(\vec{x}) \text{ and not } R_0(\vec{x}) \\ \vdots & \\ g_{m-1}(\vec{x}) & \text{if } R_{m-1}(\vec{x}) \text{ and none of the previous hold} \\ g_m(\vec{x}) & \text{otherwise} \end{cases}$$

is also primitive recursive.

Proof. When $m = 1$, this is just the function defined by

$$f(\vec{x}) = \text{cond}(\chi_{\neg R_0}(\vec{x}), g_0(\vec{x}), g_1(\vec{x})).$$

For m greater than 1, one can just compose definitions of this form. □

---

content/computability/recursive-functions/bounded-minimization.tex

29.9 Bounded Minimization

explanation

It is often useful to define a function as the least number satisfying some property or relation P. If P is decidable, we can compute this function simply by trying out all the possible numbers, 0, 1, 2, ..., until we find the least one satisfying P. This kind of unbounded search takes us out of the realm of primitive recursive functions. However, if we're only interested in the least number *less than some independently given bound*, we stay primitive recursive. In other words, and a bit more generally, suppose we have a primitive recursive relation $R(x, z)$. Consider the function that maps x and y to the least $z < y$ such that $R(x, z)$. It, too, can be computed, by testing whether $R(x, 0)$, $R(x, 1)$, ..., $R(x, y - 1)$. But why is it primitive recursive? cmp:rec:bmi:sec

Proposition 29.19. *If $R(\vec{x}, z)$ is primitive recursive, so is the function $m_R(\vec{x}, y)$ which returns the least z less than y such that $R(\vec{x}, z)$ holds, if there is one, and y otherwise. We will write the function m_R as*

$$(\min z < y)\, R(\vec{x}, z),$$

Proof. Note than there can be no $z < 0$ such that $R(\vec{x}, z)$ since there is no $z < 0$ at all. So $m_R(\vec{x}, 0) = 0$.

In case the bound is of the form $y + 1$ we have three cases:

1. There is a $z < y$ such that $R(\vec{x}, z)$, in which case $m_R(\vec{x}, y+1) = m_R(\vec{x}, y)$.
2. There is no such $z < y$ but $R(\vec{x}, y)$ holds, then $m_R(\vec{x}, y + 1) = y$.
3. There is no $z < y + 1$ such that $R(\vec{x}, z)$, then $m_R(\vec{z}, y + 1) = y + 1$.

So we can define $m_R(\vec{x}, 0)$ by primitive recursion as follows:

$$m_R(\vec{x}, 0) = 0$$

$$m_R(\vec{x}, y+1) = \begin{cases} m_R(\vec{x}, y) & \text{if } m_R(\vec{x}, y) \neq y \\ y & \text{if } m_R(\vec{x}, y) = y \text{ and } R(\vec{x}, y) \\ y + 1 & \text{otherwise.} \end{cases}$$

Note that there is a $z < y$ such that $R(\vec{x}, z)$ iff $m_R(\vec{x}, y) \neq y$. □

Problem 29.7. Suppose $R(\vec{x}, z)$ is primitive recursive. Define the function $m'_R(\vec{x}, y)$ which returns the least z less than y such that $R(\vec{x}, z)$ holds, if there is one, and 0 otherwise, by primitive recursion from χ_R.

---

content/computability/recursive-functions/primes.tex

29.10 Primes

cmp:rec:pri:sec

Bounded quantification and bounded minimization provide us with a good deal of machinery to show that natural functions and relations are primitive recursive. For example, consider the relation "x divides y", written $x \mid y$. The relation $x \mid y$ holds if division of y by x is possible without remainder, i.e., if y is an integer multiple of x. (If it doesn't hold, i.e., the remainder when dividing x by y is > 0, we write $x \nmid y$.) In other words, $x \mid y$ iff for some z, $x \cdot z = y$. Obviously, any such z, if it exists, must be $\leq y$. So, we have that $x \mid y$ iff for some $z \leq y$, $x \cdot z = y$. We can define the relation $x \mid y$ by bounded existential quantification from $=$ and multiplication by

$$x \mid y \Leftrightarrow (\exists z \leq y)\ (x \cdot z) = y.$$

We've thus shown that $x \mid y$ is primitive recursive.

A natural number x is *prime* if it is neither 0 nor 1 and is only divisible by 1 and itself. In other words, prime numbers are such that, whenever $y \mid x$, either $y = 1$ or $y = x$. To test if x is prime, we only have to check if $y \mid x$ for all $y \leq x$, since if $y > x$, then automatically $y \nmid x$. So, the relation $\mathrm{Prime}(x)$, which holds iff x is prime, can be defined by

$$\mathrm{Prime}(x) \Leftrightarrow x \geq 2 \wedge (\forall y \leq x)\ (y \mid x \rightarrow y = 1 \vee y = x)$$

and is thus primitive recursive.

The primes are 2, 3, 5, 7, 11, etc. Consider the function $p(x)$ which returns the xth prime in that sequence, i.e., $p(0) = 2$, $p(1) = 3$, $p(2) = 5$, etc. (For convenience we will often write $p(x)$ as p_x ($p_0 = 2$, $p_1 = 3$, etc.)

If we had a function nextPrime(x), which returns the first prime number larger than x, p can be easily defined using primitive recursion:

$$\begin{aligned} p(0) &= 2 \\ p(x+1) &= \mathrm{nextPrime}(p(x)) \end{aligned}$$

Since $\mathrm{nextPrime}(x)$ is the least y such that $y > x$ and y is prime, it can be easily computed by unbounded search. But it can also be defined by bounded minimization, thanks to a result due to Euclid: there is always a prime number between x and $x! + 1$.

$$\mathrm{nextPrime(x)} = (\min y \leq x! + 1)\ (y > x \wedge \mathrm{Prime}(y)).$$

This shows, that nextPrime(x) and hence $p(x)$ are (not just computable but) primitive recursive.

(If you're curious, here's a quick proof of Euclid's theorem. Suppose p_n is the largest prime $\leq x$ and consider the product $p = p_0 \cdot p_1 \cdot \cdots \cdot p_n$ of all primes $\leq x$. Either $p+1$ is prime or there is a prime between x and $p+1$. Why? Suppose $p+1$ is not prime. Then some prime number $q \mid p+1$ where $q < p+1$. None of the primes $\leq x$ divide $p+1$. (By definition of p, each of the primes $p_i \leq x$ divides p, i.e., with remainder 0. So, each of the primes $p_i \leq x$ divides $p+1$ with remainder 1, and so $p_i \nmid p+1$.) Hence, q is a prime $> x$ and $< p+1$. And $p \leq x\,!$, so there is a prime $> x$ and $\leq x\,! + 1$.)

Problem 29.8. Define integer division $d(x, y)$ using bounded minimization.

---

content/computability/recursive-functions/sequences.tex

29.11 Sequences

cmp:rec:seq:sec

The set of primitive recursive functions is remarkably robust. But we will be able to do even more once we have developed a adequate means of handling *sequences*. We will identify finite sequences of natural numbers with natural numbers in the following way: the sequence $\langle a_0, a_1, a_2, \ldots, a_k \rangle$ corresponds to the number

$$p_0^{a_0+1} \cdot p_1^{a_1+1} \cdot p_2^{a_2+1} \cdot \cdots \cdot p_k^{a_k+1}.$$

We add one to the exponents to guarantee that, for example, the sequences $\langle 2, 7, 3 \rangle$ and $\langle 2, 7, 3, 0, 0 \rangle$ have distinct numeric codes. We can take both 0 and 1 to code the empty sequence; for concreteness, let Λ denote 0.

The reason that this coding of sequences works is the so-called Fundamental Theorem of Arithmetic: every natural number $n \geq 2$ can be written in one and only one way in the form

$$n = p_0^{a_0} \cdot p_1^{a_1} \cdot \cdots \cdot p_k^{a_k}$$

with $a_k \geq 1$. This guarantees that the mapping $\langle\rangle(a_0, \ldots, a_k) = \langle a_0, \ldots, a_k \rangle$ is injective: different sequences are mapped to different numbers; to each number only at most one sequence corresponds.

We'll now show that the operations of determining the length of a sequence, determining its ith element, appending an element to a sequence, and concatenating two sequences, are all primitive recursive.

Proposition 29.20. *The function* len(s)*, which returns the length of the sequence s, is primitive recursive.*

Proof. Let $R(i, s)$ be the relation defined by

$$R(i, s) \text{ iff } p_i \mid s \wedge p_{i+1} \nmid s.$$

R is clearly primitive recursive. Whenever s is the code of a non-empty sequence, i.e.,

$$s = p_0^{a_0+1} \cdot \dots \cdot p_k^{a_k+1},$$

$R(i, s)$ holds if p_i is the largest prime such that $p_i \mid s$, i.e., $i = k$. The length of s thus is $i + 1$ iff p_i is the largest prime that divides s, so we can let

$$\text{len}(s) = \begin{cases} 0 & \text{if } s = 0 \text{ or } s = 1 \\ 1 + (\min i < s)\, R(i, s) & \text{otherwise} \end{cases}$$

We can use bounded minimization here, since there is only one i that satisfies $R(i, s)$ when s is a code of a sequence, and if i exists it is less than s itself. □

Proposition 29.21. *The function* $\text{append}(s, a)$*, which returns the result of appending* a *to the sequence* s*, is primitive recursive.*

Proof. append can be defined by:

$$\text{append}(s, a) = \begin{cases} 2^{a+1} & \text{if } s = 0 \text{ or } s = 1 \\ s \cdot p_{\text{len}(s)}^{a+1} & \text{otherwise.} \end{cases}$$

□

Proposition 29.22. *The function* $\text{element}(s, i)$*, which returns the* i*th element of* s *(where the initial element is called the* 0*th), or* 0 *if* i *is greater than or equal to the length of* s*, is primitive recursive.*

Proof. Note that a is the ith element of s iff p_i^{a+1} is the largest power of p_i that divides s, i.e., $p_i^{a+1} \mid s$ but $p_i^{a+2} \nmid s$. So:

$$\text{element}(s, i) = \begin{cases} 0 & \text{if } i \geq \text{len}(s) \\ (\min a < s)\, (p_i^{a+2} \nmid s) & \text{otherwise.} \end{cases}$$

□

Instead of using the official names for the functions defined above, we introduce a more compact notation. We will use $(s)_i$ instead of $\text{element}(s, i)$, and $\langle s_0, \dots, s_k \rangle$ to abbreviate

$$\text{append}(\text{append}(\dots \text{append}(\Lambda, s_0) \dots), s_k).$$

Note that if s has length k, the elements of s are $(s)_0, \dots, (s)_{k-1}$.

Proposition 29.23. *The function* $\text{concat}(s, t)$*, which concatenates two sequences, is primitive recursive.*

Proof. We want a function concat with the property that

$$\text{concat}(\langle a_0, \dots, a_k \rangle, \langle b_0, \dots, b_l \rangle) = \langle a_0, \dots, a_k, b_0, \dots, b_l \rangle.$$

We'll use a "helper" function $\text{hconcat}(s, t, n)$ which concatenates the first n symbols of t to s. This function can be defined by primitive recursion as follows:

$$\begin{aligned}\text{hconcat}(s, t, 0) &= s \\ \text{hconcat}(s, t, n+1) &= \text{append}(\text{hconcat}(s, t, n), (t)_n)\end{aligned}$$

Then we can define concat by

$$\text{concat}(s, t) = \text{hconcat}(s, t, \text{len}(t)). \qquad \square$$

We will write $s \frown t$ instead of $\text{concat}(s, t)$.

It will be useful for us to be able to bound the numeric code of a sequence in terms of its length and its largest element. Suppose s is a sequence of length k, each element of which is less than or equal to some number x. Then s has at most k prime factors, each at most p_{k-1}, and each raised to at most $x + 1$ in the prime factorization of s. In other words, if we define

$$\text{sequenceBound}(x, k) = p_{k-1}^{k \cdot (x+1)},$$

then the numeric code of the sequence s described above is at most $\text{sequenceBound}(x, k)$.

Having such a bound on sequences gives us a way of defining new functions using bounded search. For example, we can define concat using bounded search. All we need to do is write down a primitive recursive *specification* of the object (number of the concatenated sequence) we are looking for, and a bound on how far to look. The following works:

$$\begin{aligned}\text{concat}(s, t) = (&\min v < \text{sequenceBound}(s + t, \text{len}(s) + \text{len}(t))) \\ &(\text{len}(v) = \text{len}(s) + \text{len}(t) \wedge \\ &\quad (\forall i < \text{len}(s))\ ((v)_i = (s)_i) \wedge \\ &\quad (\forall j < \text{len}(t))\ ((v)_{\text{len}(s)+j} = (t)_j))\end{aligned}$$

Problem 29.9. Show that there is a primitive recursive function $\text{sconcat}(s)$ with the property that

$$\text{sconcat}(\langle s_0, \ldots, s_k \rangle) = s_0 \frown \ldots \frown s_k.$$

Problem 29.10. Show that there is a primitive recursive function $\text{tail}(s)$ with the property that

$$\begin{aligned}\text{tail}(\Lambda) &= 0 \text{ and} \\ \text{tail}(\langle s_0, \ldots, s_k \rangle) &= \langle s_1, \ldots, s_k \rangle.\end{aligned}$$

Proposition 29.24. *The function* $\text{subseq}(s, i, n)$ *which returns the subsequence of* s *of length* n *beginning at the* i*th element, is primitive recursive.* *cmp:rec:seq: prop:subseq*

Proof. Exercise. $\square$

Problem 29.11. Prove Proposition 29.24.

---

content/computability/recursive-functions/trees.tex

29.12 Trees

cmp:rec:tre:sec

Sometimes it is useful to represent trees as natural numbers, just like we can represent sequences by numbers and properties of and operations on them by primitive recursive relations and functions on their codes. We'll use sequences and their codes to do this. A tree can be either a single node (possibly with a label) or else a node (possibly with a label) connected to a number of subtrees. The node is called the *root* of the tree, and the subtrees it is connected to its *immediate subtrees*.

We code trees recursively as a sequence $\langle k, d_1, \ldots, d_k \rangle$, where k is the number of immediate subtrees and d_1, …, d_k the codes of the immediate subtrees. If the nodes have labels, they can be included after the immediate subtrees. So a tree consisting just of a single node with label l would be coded by $\langle 0, l \rangle$, and a tree consisting of a root (labelled l_1) connected to two single nodes (labelled l_2, l_3) would be coded by $\langle 2, \langle 0, l_2 \rangle, \langle 0, l_3 \rangle, l_1 \rangle$.

cmp:rec:tre:prop:subtreeseq

Proposition 29.25. *The function* $\mathrm{SubtreeSeq}(t)$*, which returns the code of a sequence the elements of which are the codes of all subtrees of the tree with code* t*, is primitive recursive.*

Proof. First note that $\mathrm{ISubtrees}(t) = \mathrm{subseq}(t, 1, (t)_0)$ is primitive recursive and returns the codes of the immediate subtrees of a tree t. Now we can define a helper function $\mathrm{hSubtreeSeq}(t, n)$ which computes the sequence of all subtrees which are n nodes removed from the root. The sequence of subtrees of t which is 0 nodes removed from the root—in other words, begins at the root of t—is the sequence consisting just of t. To obtain a sequence of all level $n+1$ subtrees of t, we concatenate the level n subtrees with a sequence consisting of all immediate subtrees of the level n subtrees. To get a list of all these, note that if $f(x)$ is a primitive recursive function returning codes of sequences, then $g_f(s, k) = f((s)_0) \frown \ldots \frown f((s)_k)$ is also primitive recursive:

$$\begin{aligned} g(s, 0) &= f((s)_0) \\ g(s, k+1) &= g(s, k) \frown f((s)_{k+1}) \end{aligned}$$

For instance, if s is a sequence of trees, then $h(s) = g_{\mathrm{ISubtrees}}(s, \mathrm{len}(s))$ gives the sequence of the immediate subtrees of the elements of s. We can use it to define hSubtreeSeq by

$$\begin{aligned} \mathrm{hSubtreeSeq}(t, 0) &= \langle t \rangle \\ \mathrm{hSubtreeSeq}(t, n+1) &= \mathrm{hSubtreeSeq}(t, n) \frown h(\mathrm{hSubtreeSeq}(t, n)). \end{aligned}$$

The maximum level of subtrees in a tree coded by t, i.e., the maximum distance between the root and a leaf node, is bounded by the code t. So a sequence of codes of all subtrees of the tree coded by t is given by $\mathrm{hSubtreeSeq}(t, t)$. □

Problem 29.12. The definition of hSubtreeSeq in the proof of Proposition 29.25 in general includes repetitions. Give an alternative definition which guarantees that the code of a subtree occurs only once in the resulting list.

---

content/computability/recursive-functions/other-recursions.tex

29.13 Other Recursions

cmp:rec:ore:sec

Using pairing and sequencing, we can justify more exotic (and useful) forms of primitive recursion. For example, it is often useful to define two functions simultaneously, such as in the following definition:

$$\begin{aligned} h_0(\vec{x}, 0) &= f_0(\vec{x}) \\ h_1(\vec{x}, 0) &= f_1(\vec{x}) \\ h_0(\vec{x}, y+1) &= g_0(\vec{x}, y, h_0(\vec{x}, y), h_1(\vec{x}, y)) \\ h_1(\vec{x}, y+1) &= g_1(\vec{x}, y, h_0(\vec{x}, y), h_1(\vec{x}, y)) \end{aligned}$$

This is an instance of *simultaneous recursion*. Another useful way of defining functions is to give the value of $h(\vec{x}, y+1)$ in terms of *all* the values $h(\vec{x}, 0)$, $\ldots$, $h(\vec{x}, y)$, as in the following definition:

$$\begin{aligned} h(\vec{x}, 0) &= f(\vec{x}) \\ h(\vec{x}, y+1) &= g(\vec{x}, y, \langle h(\vec{x}, 0), \ldots, h(\vec{x}, y)\rangle). \end{aligned}$$

The following schema captures this idea more succinctly:

$$h(\vec{x}, y) = g(\vec{x}, y, \langle h(\vec{x}, 0), \ldots, h(\vec{x}, y-1)\rangle)$$

with the understanding that the last argument to g is just the empty sequence when y is 0. In either formulation, the idea is that in computing the "successor step," the function h can make use of the entire sequence of values computed so far. This is known as a *course-of-values* recursion. For a particular example, it can be used to justify the following type of definition:

$$h(\vec{x}, y) = \begin{cases} g(\vec{x}, y, h(\vec{x}, k(\vec{x}, y))) & \text{if } k(\vec{x}, y) < y \\ f(\vec{x}) & \text{otherwise} \end{cases}$$

In other words, the value of h at y can be computed in terms of the value of h at *any* previous value, given by k.

Problem 29.13. Define the remainder function $r(x, y)$ by course-of-values recursion. (If x, y are natural numbers and $y > 0$, $r(x, y)$ is the number less than y such that $x = z \times y + r(x, y)$ for some z. For definiteness, let's say that if $y = 0$, $r(x, 0) = 0$.)

You should think about how to obtain these functions using ordinary primitive recursion. One final version of primitive recursion is more flexible in that one is allowed to change the *parameters* (side values) along the way:

$$\begin{aligned} h(\vec{x}, 0) &= f(\vec{x}) \\ h(\vec{x}, y+1) &= g(\vec{x}, y, h(k(\vec{x}), y)) \end{aligned}$$

This, too, can be simulated with ordinary primitive recursion. (Doing so is tricky. For a hint, try unwinding the computation by hand.)

---

content/computability/recursive-functions/non-pr-functions.tex

29.14 Non-Primitive Recursive Functions

cmp:rec:npr:sec

The primitive recursive functions do not exhaust the intuitively computable functions. It should be intuitively clear that we can make a list of all the unary primitive recursive functions, f_0, f_1, f_2, ... such that we can effectively compute the value of f_x on input y; in other words, the function $g(x, y)$, defined by

$$g(x, y) = f_x(y)$$

is computable. But then so is the function

$$\begin{aligned} h(x) &= g(x, x) + 1 \\ &= f_x(x) + 1. \end{aligned}$$

For each primitive recursive function f_i, the value of h and f_i differ at i. So h is computable, but not primitive recursive; and one can say the same about g. This is an "effective" version of Cantor's diagonalization argument.

One can provide more explicit examples of computable functions that are not primitive recursive. For example, let the notation $g^n(x)$ denote $g(g(\ldots g(x)))$, with n g's in all; and define a sequence $g_0, g_1, \ldots$ of functions by

$$\begin{aligned} g_0(x) &= x + 1 \\ g_{n+1}(x) &= g_n^x(x) \end{aligned}$$

You can confirm that each function g_n is primitive recursive. Each successive function grows much faster than the one before; $g_1(x)$ is equal to $2x$, $g_2(x)$ is equal to $2^x \cdot x$, and $g_3(x)$ grows roughly like an exponential stack of x 2's. The Ackermann–Péter function is essentially the function $G(x) = g_x(x)$, and one can show that this grows faster than any primitive recursive function.

Let us return to the issue of enumerating the primitive recursive functions. Remember that we have assigned symbolic notations to each primitive recursive function; so it suffices to enumerate notations. We can assign a natural number $\#(F)$ to each notation F, recursively, as follows:

$$\begin{aligned}
\#(0) &= \langle 0 \rangle \\
\#(S) &= \langle 1 \rangle \\
\#(P_i^n) &= \langle 2, n, i \rangle \\
\#(\mathrm{Comp}_{k,l}[H, G_0, \ldots, G_{k-1}]) &= \langle 3, k, l, \#(H), \#(G_0), \ldots, \#(G_{k-1}) \rangle \\
\#(\mathrm{Rec}_l[G, H]) &= \langle 4, l, \#(G), \#(H) \rangle
\end{aligned}$$

Here we are using the fact that every sequence of numbers can be viewed as a natural number, using the codes from the last section. The upshot is that every code is assigned a natural number. Of course, some sequences (and hence some numbers) do not correspond to notations; but we can let f_i be the unary primitive recursive function with notation coded as i, if i codes such a notation; and the constant 0 function otherwise. The net result is that we have an explicit way of enumerating the unary primitive recursive functions.

(In fact, some functions, like the constant zero function, will appear more than once on the list. This is not just an artifact of our coding, but also a result of the fact that the constant zero function has more than one notation. We will later see that one can not computably avoid these repetitions; for example, there is no computable function that decides whether or not a given notation represents the constant zero function.)

We can now take the function $g(x, y)$ to be given by $f_x(y)$, where f_x refers to the enumeration we have just described. How do we know that $g(x, y)$ is computable? Intuitively, this is clear: to compute $g(x, y)$, first "unpack" x, and see if it is a notation for a unary function. If it is, compute the value of that function on input y.

digression

You may already be convinced that (with some work!) one can write a program (say, in Java or C++) that does this; and now we can appeal to the Church–Turing thesis, which says that anything that, intuitively, is computable can be computed by a Turing machine.

Of course, a more direct way to show that $g(x, y)$ is computable is to describe a Turing machine that computes it, explicitly. This would, in particular, avoid the Church–Turing thesis and appeals to intuition. Soon we will have built up enough machinery to show that $g(x, y)$ is computable, appealing to a model of computation that can be *simulated* on a Turing machine: namely, the recursive functions.

---

content/computability/recursive-functions/partial-functions.tex

29.15 Partial Recursive Functions

cmp:rec:par:sec

To motivate the definition of the recursive functions, note that our proof that there are computable functions that are not primitive recursive actually establishes much more. The argument was simple: all we used was the fact that it is possible to enumerate functions $f_0, f_1, \ldots$ such that, as a function of x and y, $f_x(y)$ is computable. So the argument applies to *any class of functions that can be enumerated in such a way.* This puts us in a bind: we would like to describe the computable functions explicitly; but any explicit description of a collection of computable functions cannot be exhaustive!

The way out is to allow *partial* functions to come into play. We will see that it *is* possible to enumerate the partial computable functions. In fact, we already pretty much know that this is the case, since it is possible to enumerate Turing machines in a systematic way. We will come back to our diagonal argument later, and explore why it does not go through when partial functions are included.

The question is now this: what do we need to add to the primitive recursive functions to obtain all the partial recursive functions? We need to do two things:

1. Modify our definition of the primitive recursive functions to allow for partial functions as well.

2. *Add* something to the definition, so that some new partial functions are included.

The first is easy. As before, we will start with zero, successor, and projections, and close under composition and primitive recursion. The only difference is that we have to modify the definitions of composition and primitive recursion to allow for the possibility that some of the terms in the definition are not defined. If f and g are partial functions, we will write $f(x) \downarrow$ to mean that f is defined at x, i.e., x is in the domain of f; and $f(x) \uparrow$ to mean the opposite, i.e., that f is not defined at x. We will use $f(x) \simeq g(x)$ to mean that either $f(x)$ and $g(x)$ are both undefined, or they are both defined and equal. We will use these notations for more complicated terms as well. We will adopt the convention that if h and $g_0, \ldots, g_k$ all are partial functions, then

$$h(g_0(\vec{x}), \ldots, g_k(\vec{x}))$$

is defined if and only if each g_i is defined at $\vec{x}$, and h is defined at $g_0(\vec{x})$, $\ldots$, $g_k(\vec{x})$. With this understanding, the definitions of composition and primitive recursion for partial functions is just as above, except that we have to replace "$=$" by "$\simeq$".

What we will add to the definition of the primitive recursive functions to obtain partial functions is the *unbounded search operator.* If $f(x, \vec{z})$ is any partial function on the natural numbers, define $\mu x\, f(x, \vec{z})$ to be

> the least x such that $f(0, \vec{z}), f(1, \vec{z}), \ldots, f(x, \vec{z})$ are all defined, and $f(x, \vec{z}) = 0$, if such an x exists

with the understanding that $\mu x\ f(x, \vec{z})$ is undefined otherwise. This defines $\mu x\ f(x, \vec{z})$ uniquely.

explanation

Note that our definition makes no reference to Turing machines, or algorithms, or any specific computational model. But like composition and primitive recursion, there is an operational, computational intuition behind unbounded search. When it comes to the computability of a partial function, arguments where the function is undefined correspond to inputs for which the computation does not halt. The procedure for computing $\mu x\ f(x, \vec{z})$ will amount to this: compute $f(0, \vec{z}), f(1, \vec{z}), f(2, \vec{z})$ until a value of 0 is returned. If any of the intermediate computations do not halt, however, neither does the computation of $\mu x\ f(x, \vec{z})$.

If $R(x, \vec{z})$ is any relation, $\mu x\ R(x, \vec{z})$ is defined to be $\mu x\ (1 \dot{-} \chi_R(x, \vec{z}))$. In other words, $\mu x\ R(x, \vec{z})$ returns the least value of x such that $R(x, \vec{z})$ holds. So, if $f(x, \vec{z})$ is a total function, $\mu x\ f(x, \vec{z})$ is the same as $\mu x\ (f(x, \vec{z}) = 0)$. But note that our original definition is more general, since it allows for the possibility that $f(x, \vec{z})$ is not everywhere defined (whereas, in contrast, the characteristic function of a relation is always total).

Definition 29.26. The set of *partial recursive functions* is the smallest set of partial functions from the natural numbers to the natural numbers (of various arities) containing zero, successor, and projections, and closed under composition, primitive recursion, and unbounded search.

Of course, some of the partial recursive functions will happen to be total, i.e., defined for every argument.

Definition 29.27. The set of *recursive functions* is the set of partial recursive functions that are total.

cmp:rec:par: defn:recursive-fn

A recursive function is sometimes called "total recursive" to emphasize that it is defined everywhere.

---

content/computability/recursive-functions/normal-form.tex

29.16 The Normal Form Theorem

cmp:rec:nft: sec
cmp:rec:nft: thm:kleene-nf

Theorem 29.28 (Kleene's Normal Form Theorem). *There is a primitive recursive relation $T(e, x, s)$ and a primitive recursive function $U(s)$, with the following property: if f is any partial recursive function, then for some e,*

$$f(x) \simeq U(\mu s\ T(e, x, s))$$

for every x.

explanation

The proof of the normal form theorem is involved, but the basic idea is simple. Every partial recursive function has an *index* e, intuitively, a number

coding its program or definition. If $f(x)\downarrow$, the computation can be recorded systematically and coded by some number s, and the fact that s codes the computation of f on input x can be checked primitive recursively using only x and the definition e. Consequently, the relation T, "the function with index e has a computation for input x, and s codes this computation," is primitive recursive. Given the full record of the computation s, the "upshot" of s is the value of $f(x)$, and it can be obtained from s primitive recursively as well.

The normal form theorem shows that only a single unbounded search is required for the definition of any partial recursive function. Basically, we can search through all numbers until we find one that codes a computation of the function with index e for input x. We can use the numbers e as "names" of partial recursive functions, and write φ_e for the function f defined by the equation in the theorem. Note that any partial recursive function can have more than one index—in fact, every partial recursive function has infinitely many indices.

---

content/computability/recursive-functions/halting-problem.tex

29.17 The Halting Problem

cmp:rec:hlt:sec

The *halting problem* in general is the problem of deciding, given the specification e (e.g., program) of a computable function and a number n, whether the computation of the function on input n halts, i.e., produces a result. Famously, Alan Turing proved that this problem itself cannot be solved by a computable function, i.e., the function

$$h(e,n) = \begin{cases} 1 & \text{if computation } e \text{ halts on input } n \\ 0 & \text{otherwise,} \end{cases}$$

is not computable.

In the context of partial recursive functions, the role of the specification of a program may be played by the index e given in Kleene's normal form theorem. If f is a partial recursive function, any e for which the equation in the normal form theorem holds, is an index of f. Given a number e, the normal form theorem states that

$$\varphi_e(x) \simeq U(\mu s\, T(e,x,s))$$

is partial recursive, and for every partial recursive $f\colon \mathbb{N} \to \mathbb{N}$, there is an $e \in \mathbb{N}$ such that $\varphi_e(x) \simeq f(x)$ for all $x \in \mathbb{N}$. In fact, for each such f there is not just one, but infinitely many such e. The *halting function* h is defined by

$$h(e,x) = \begin{cases} 1 & \text{if } \varphi_e(x)\downarrow \\ 0 & \text{otherwise.} \end{cases}$$

Note that $h(e,x) = 0$ if $\varphi_e(x)\uparrow$, but also when e is not the index of a partial recursive function at all.

Theorem 29.29. *The halting function h is not partial recursive.* cmp:rec:hlt: thm:halting-problem

Proof. If h were partial recursive, we could define

$$d(y) = \begin{cases} 1 & \text{if } h(y,y) = 0 \\ \mu x\ x \neq x & \text{otherwise.} \end{cases}$$

Since no number x satisfies $x \neq x$, there is no $\mu x\ x \neq x$, and so $d(y) \uparrow$ iff $h(y,y) \neq 0$. From this definition it follows that

1. $d(y) \downarrow$ iff $\varphi_y(y) \uparrow$ or y is not the index of a partial recursive function.
2. $d(y) \uparrow$ iff $\varphi_y(y) \downarrow$.

If h were partial recursive, then d would be partial recursive as well. Thus, by the Kleene normal form theorem, it has an index e_d. Consider the value of $h(e_d, e_d)$. There are two possible cases, 0 and 1.

1. If $h(e_d, e_d) = 1$ then $\varphi_{e_d}(e_d) \downarrow$. But $\varphi_{e_d} \simeq d$, and $d(e_d)$ is defined iff $h(e_d, e_d) = 0$. So $h(e_d, e_d) \neq 1$.
2. If $h(e_d, e_d) = 0$ then either e_d is not the index of a partial recursive function, or it is and $\varphi_{e_d}(e_d) \uparrow$. But again, $\varphi_{e_d} \simeq d$, and $d(e_d)$ is undefined iff $\varphi_{e_d}(e_d) \downarrow$.

The upshot is that e_d cannot, after all, be the index of a partial recursive function. But if h were partial recursive, d would be too, and so our definition of e_d as an index of it would be admissible. We must conclude that h cannot be partial recursive. □

---

content/computability/recursive-functions/general-recursive-functions.tex

29.18 General Recursive Functions

There is another way to obtain a set of total functions. Say a total function $f(x, \vec{z})$ is *regular* if for every sequence of natural numbers $\vec{z}$, there is an x such that $f(x, \vec{z}) = 0$. In other words, the regular functions are exactly those functions to which one can apply unbounded search, and end up with a total function. One can, conservatively, restrict unbounded search to regular functions: cmp:rec:gen: sec

Definition 29.30. The set of *general recursive functions* is the smallest set of functions from the natural numbers to the natural numbers (of various arities) containing zero, successor, and projections, and closed under composition, primitive recursion, and unbounded search applied to *regular* functions. cmp:rec:gen: defn:general-recursive

Clearly every general recursive function is total. The difference between Definition 29.30 and Definition 29.27 is that in the latter one is allowed to use partial recursive functions along the way; the only requirement is that the function you end up with at the end is total. So the word "general," a historic relic, is a misnomer; on the surface, Definition 29.30 is *less* general than Definition 29.27. But, fortunately, the difference is illusory; though the definitions are different, the set of general recursive functions and the set of recursive functions are one and the same.

Chapter 30

Computability Theory

Material in this chapter should be reviewed and expanded. In particular, there are no exercises yet.

---

content/computability/computability-theory/introduction.tex

30.1 Introduction

cmp:thy:int:sec

The branch of logic known as *computability theory* deals with issues having to do with the computability, or relative computability, of functions and sets. It is evidence of Kleene's influence that the subject used to be known as *recursion theory*, and today, both names are commonly used.

Let us call a function $f\colon \mathbb{N} \nrightarrow \mathbb{N}$ *partial computable* if it can be computed in some model of computation. If f is total we will simply say that f is *computable*. A relation R with computable characteristic function χ_R is also called computable. If f and g are partial functions, we will write $f(x)\downarrow$ to mean that f is defined at x, i.e., x is in the domain of f; and $f(x)\uparrow$ to mean the opposite, i.e., that f is not defined at x. We will use $f(x) \simeq g(x)$ to mean that either $f(x)$ and $g(x)$ are both undefined, or they are both defined and equal.

One can explore the theory of computability without having to refer to a specific model of computation. To do this, one shows that there is a universal

partial computable function $\mathrm{Un}(k, x)$. This allows us to enumerate the partial computable functions. We will adopt the notation φ_k to denote the k-th unary partial computable function, defined by $\varphi_k(x) \simeq \mathrm{Un}(k, x)$. (Kleene used $\{k\}$ for this purpose, but this notation has not been used as much recently.) Slightly more generally, we can uniformly enumerate the partial computable functions of arbitrary arities, and we will use φ_k^n to denote the k-th n-ary partial recursive function.

If $f(\vec{x}, y)$ is a total or partial function, then $\mu y\ f(\vec{x}, y)$ is the function of $\vec{x}$ that returns the least y such that $f(\vec{x}, y) = 0$, assuming that all of $f(\vec{x}, 0), \ldots, f(\vec{x}, y-1)$ are defined; if there is no such y, $\mu y\ f(\vec{x}, y)$ is undefined. If $R(\vec{x}, y)$ is a relation, $\mu y\ R(\vec{x}, y)$ is defined to be the least y such that $R(\vec{x}, y)$ is true; in other words, the least y such that $1 \dot{-} \chi_R(\vec{x}, y) = 0$.

To show that a function is computable, there are two ways one can proceed:

1. Rigorously: describe a Turing machine or partial recursive function explicitly, and show that it computes the function you have in mind;

2. Informally: describe an algorithm that computes it, and appeal to Church's thesis.

There is no fine line between the two; a detailed description of an algorithm should provide enough information so that it is relatively clear how one could, in principle, design the right Turing machine or sequence of partial recursive definitions. Fully rigorous definitions are unlikely to be informative, and we will try to find a happy medium between these two approaches; in short, we will try to find intuitive yet rigorous proofs that the precise definitions could be obtained.

---

content/computability/computability-theory/coding-computations.tex

30.2 Coding Computations

cmp:thy:cod:sec

In every model of computation, it is possible to do the following:

1. Describe the *definitions* of computable functions in a systematic way. For instance, you can think of Turing machine specifications, recursive definitions, or programs in a programming language as providing these definitions.

2. Describe the complete record of the computation of a function given by some definition for a given input. For instance, a Turing machine computation can be described by the sequence of configurations (state of the machine, contents of the tape) for each step of computation.

3. Test whether a putative record of a computation is in fact the record of how a computable function with a given definition would be computed for a given input (on which the function is defined, i.e., the computation halts).

4. Extract from such a description of the complete record of a computation the value of the function for a given input. For instance, the contents of the tape in the very last step of a halting Turing machine computation is the value.

Using coding, it is possible to assign to each description of a computable function a numerical *index* in such a way that the instructions can be recovered from the index in a computable way. Similarly, the complete record of a computation can be coded by a single number as well. The resulting arithmetical relation "s codes the record of computation of the function with index e for input x" and the function "output of computation sequence with code s" are then computable; in fact, they are primitive recursive.

This fundamental fact is very powerful, and allows us to prove a number of striking and important results about computability, independently of the model of computation chosen.

---

content/computability/computability-theory/normal-form.tex

30.3 The Normal Form Theorem

cmp:thy:nfm:sec

Suppose we can describe definitions of computable functions, and test if some putative description of the complete record of the computation of that function on some input is correct. Then it stands to reason that independently of the model of computation, we can determine the value of any computable function f on any input x as follows:

1. Search through all possible descriptions of records of computation.
2. Test if a given record is the record of a computation of $f(x)$.
3. Extract the value of $f(x)$ from the correct record if we have found it.

That this is in fact true is the content of Kleene's normal form theorem.

cmp:thy:nfm:thm:normal-form

Theorem 30.1 (Kleene's Normal Form Theorem). *There is a primitive recursive relation $T(e, x, s)$ and a primitive recursive function $U(s)$, with the following property: if f is any partial computable function, then for some e,*

$$f(x) \simeq U(\mu s\, T(e, x, s))$$

for every x.

Proof Sketch. For any model of computation one can rigorously define a description of the computable function f and code such description using a natural number e. One can also rigorously define a notion of "computation sequence" which records the process of computing the function with index e for input x. Such a computation sequence can likewise be coded as a number s. This can be done in such a way that

1. the relation $T(e, x, s)$, which holds iff a number s codes the computation sequence of the function with index e on input x, and
2. the function $U(s)$ which maps a computation sequence coded by s to the end result of that computation

are both computable. In fact, the relation T and the function U are primitive recursive. □

explanation

In order to give a rigorous proof of the Normal Form Theorem, we would have to fix a model of computation and carry out the coding of descriptions of computable functions and of computation sequences in detail, and verify that the relation T and function U are primitive recursive. For most applications, it suffices that T and U are computable and that U is total.

It is probably best to remember the proof of the normal form theorem in slogan form: $\mu s\, T(e, x, s)$ searches for a computation sequence of the function with index e on input x, and U returns the output of the computation sequence if one can be found.

If the model of computation is the partial recursive functions (which is what Kleene originally used), it shows that only a single use of unbounded search, i.e., a single $\mu y\, f(\vec{x}, y)$ operator is necessary for the definition of any function. In this sense it shows that any partial recursive function has a normal form.

T and U can be used to define the enumeration φ_0, φ_1, φ_2, ... From now on, we will assume that we have fixed a suitable choice of T and U, and take the equation

$$\varphi_e(x) \simeq U(\mu s\, T(e, x, s))$$

to be the *definition* of φ_e.

Here is another useful fact:

Theorem 30.2. *Every partial computable function has infinitely many indices.*

Again, this is intuitively clear. Given any (description of) a computable function, one can come up with a different description which computes the same function (input-output pair) but does so, e.g., by first doing something that has no effect on the computation (say, test if $0 = 0$, or count to 5, etc.). The index of the altered description will always be different from the original index. Both are indices of the same function, just computed slightly differently.

---

content/computability/computability-theory/s-m-n.tex

30.4 The s-m-n Theorem

explanation

The next theorem is known as the "s-m-n theorem," for a reason that will be clear in a moment. The hard part is understanding just what the theorem says; once you understand the statement, it will seem fairly obvious.

cmp:thy:smn:sec

cmp:thy:smn: thm:s-m-n

Theorem 30.3. *For each pair of natural numbers n and m, there is a primitive recursive function s^m_n such that for every sequence e, a_0, …, a_{m-1}, y_0 ,…, y_{n-1}, we have*

$$\varphi^n_{s^m_n(e,a_0,\ldots,a_{m-1})}(y_0,\ldots,y_{n-1}) \simeq \varphi^{m+n}_e(a_0,\ldots,a_{m-1},y_0,\ldots,y_{n-1}).$$

It is helpful to think of s^m_n as acting on *programs*. That is, s^m_n takes a program e for an $(m+n)$-ary function, as well as fixed inputs a_0, …, a_{m-1}; and it returns a program $s^m_n(x, a_0, \ldots, a_{m-1})$ for the n-ary function of the remaining arguments. It you think of x as the description of a Turing machine, then $s^m_n(e, a_0, \ldots, a_{m-1})$ is the Turing machine that, on input y_0, …, y_{n-1}, prepends a_0, …, a_{m-1} to the input string, and runs e. Each s^m_n is then just a primitive recursive function that finds a code for the appropriate Turing machine. explanation

content/computability/computability-theory/universal-part-function.tex

30.5 The Universal Partial Computable Function

cmp:thy:uni: sec
cmp:thy:uni: thm:univ-comp

Theorem 30.4. *There is a universal partial computable function* $\mathrm{Un}(e, x)$. *In other words, there is a function* $\mathrm{Un}(e, x)$ *such that:*

1. $\mathrm{Un}(e, x)$ *is partial computable.*
2. *If $f(x)$ is any partial computable function, then there is a natural number e such that $f(x) \simeq \mathrm{Un}(e, x)$ for every x.*

Proof. Let $\mathrm{Un}(e, x) \simeq U(\mu s\ T(e, x, s))$, where U and T are as in Kleene's normal form theorem (Theorem 30.1). □

This is just a precise way of saying that we have an effective enumeration of the partial computable functions; the idea is that if we write f_e for the function defined by $f_e(x) = \mathrm{Un}(e, x)$, then the sequence f_0, f_1, f_2, … includes all the partial computable functions, with the property that $f_e(x)$ can be computed "uniformly" in e and x. For simplicity, we are using a binary function that is universal for unary functions, but by coding sequences of numbers we can easily generalize this to more arguments. For example, note that if $f(x, y, z)$ is a 3-place partial recursive function, then the function $g(x) \simeq f((x)_0, (x)_1, (x)_2)$ is a unary recursive function. explanation

content/computability/computability-theory/no-universal-function.tex

30.6 No Universal Computable Function

cmp:thy:nou: sec

Although there is a partial computable function that is total for the partial computable functions, there is no total computable function that is universal for the total computable functions.

cmp:thy:nou: thm:no-univ

Theorem 30.5. *There is no universal computable function. In other words, any function* $\mathrm{Un}'(k, x)$ *which is such that if* $f(x)$ *is a total computable function, then there is a natural number* k *such that* $f(x) = \mathrm{Un}'(k, x)$ *for every* x*, is not computable.*

Proof. The proof is a simple diagonalization: if $\mathrm{Un}'(k, x)$ were total and computable, then

$$d(x) = \mathrm{Un}'(x, x) + 1$$

would also be total and computable. However, by definition, $d(k)$ is not equal to $\mathrm{Un}'(k, k)$. Hence, for every k, the values of $d(x)$ and $\mathrm{Un}'(k, x)$ differ for at least one x, namely $x = k$. □

explanation

Theorem 30.4 above shows that we can get around this diagonalization argument, but only at the expense of allowing the universal function to be partial. That is, Un is universal for the total computable functions, it just isn't total. The diagonalization argument doesn't work in the partial case.

Problem 30.1. To understand why the diagonalization argument in the proof of Theorem 30.5 does not work in the partial case, consider the function $f(x) \simeq \mathrm{Un}(x, x) + 1$. Is it partial computable? If so, it has an index e, i.e., $f(x) \simeq \mathrm{Un}(e, x)$. What can you say about $f(e)$?

---

content/computability/computability-theory/halting-problem.tex

30.7 The Halting Problem

cmp:thy:hlt: sec

By construction, the universal partial computable function $\mathrm{Un}(e, x)$ is defined if and only if the computation of the function coded by e produces a value for input x. It is natural to ask if we can decide whether this is the case. In fact, it is not. For the Turing machine model of computation, this means that whether a given Turing machine halts on a given input is computationally undecidable. The following theorem is therefore known as the "undecidability of the halting problem." We will provide two proofs below. The first continues the thread of our previous discussion, while the second is more direct.

cmp:thy:hlt: thm:halting-problem

Theorem 30.6. *Let*

$$h(e, x) = \begin{cases} 1 & \text{if } \mathrm{Un}(e, x) \text{ is defined} \\ 0 & \text{otherwise.} \end{cases}$$

Then h *is not computable.*

Proof. Suppose h is computable. We show that this would let us define a universal computable function. Define

$$\mathrm{Un}'(e,x) = \begin{cases} \mathrm{Un}(e,x) & \text{if } h(e,x) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

But now $\mathrm{Un}'(e,x)$ is a total function, and is computable if h is. For instance, we could define g using primitive recursion, by

$$\begin{aligned} g(0,e,x) &\simeq 0 \\ g(y+1,e,x) &\simeq \mathrm{Un}(e,x); \end{aligned}$$

then

$$\mathrm{Un}'(e,x) \simeq g(h(e,x),e,x).$$

Since $\mathrm{Un}'(e,x)$ agrees with $\mathrm{Un}(e,x)$ wherever the latter is defined, Un' is universal for those partial computable functions that happen to be total. But this contradicts Theorem 30.5. □

Proof. Suppose $h(e,x)$ were computable. Define the function g by

$$g(x) = \begin{cases} 0 & \text{if } h(x,x) = 0 \\ \uparrow & \text{otherwise.} \end{cases}$$

The function g is partial computable. For example, one can define it as $\mu y\ h(x,x) = 0$. So, for some e, $g(x) \simeq \mathrm{Un}(e,x)$ for every x. Is g defined at e? If it is, then, by the definition of g, $h(e,e) = 0$ (h can only take the value 0 if it is defined). By the definition of h, this means that $\mathrm{Un}(e,e)$ is undefined. By our assumption that $g(x) \simeq \mathrm{Un}(e,x)$ for every x, we have that $g(e)$ is undefined, a contradiction. On the other hand, if $g(e)$ is undefined, then $h(e,e) \neq 0$, and so $h(e,e) = 1$. It follows that $\mathrm{Un}(e,e)$ is defined. But since $g(x) \simeq \mathrm{Un}(e,x)$, then $g(e)$ would also be defined. Again, a contradiction. □

explanation

We can describe this argument in terms of Turing machines. Suppose there were a Turing machine H that takes as input a description of a Turing machine E and an input x, and decides whether or not E halts on input x. Then we could build another Turing machine G which takes a single input x, runs H to decide if the machine M_x with index x halts on input x, and does the opposite. In other words, if H reports that M_x halts on input x, G goes into an infinite loop, and if H reports that M_x doesn't halt on input x, then G just halts. Does G halt on its own index as input? The argument above shows that it does if and only if it doesn't—a contradiction. So our supposition that there is a such Turing machine H must be false.

content/computability/computability-theory/russells-paradox.tex

30.8 Comparison with Russell's Paradox

cmp:thy:rus:sec

It is instructive to compare and contrast the arguments in this section with Russell's paradox:

1. Russell's paradox: let $S = \{x : x \notin x\}$. Then $S \in S$ if and only if $X \notin S$, a contradiction.

   Conclusion: There is no such set S. Assuming the existence of a "set of all sets" is inconsistent with the other axioms of set theory.

2. A modification of Russell's paradox: let F be the "function" from the set of all functions to $\{0, 1\}$, defined by

$$F(f) = \begin{cases} 1 & \text{if } f \text{ is in the domain of } f, \text{ and } f(f) = 0 \\ 0 & \text{otherwise} \end{cases}$$

   A similar argument shows that $F(F) = 0$ if and only if $F(F) = 1$, a contradiction.

   Conclusion: F is not a function. The "set of all functions" is too big to be the domain of a function.

3. The diagonalization argument: let f_0, f_1, ... be the enumeration of the partial computable functions, and let $G\colon \mathbb{N} \to \{0, 1\}$ be defined by

$$G(x) = \begin{cases} 1 & \text{if } f_x(x) \downarrow = 0 \\ 0 & \text{otherwise} \end{cases}$$

   If G is computable, then it is the function f_k for some k. But then $G(k) = 1$ if and only if $G(k) = 0$, a contradiction.

   Conclusion: G is not computable. Note that according to the axioms of set theory, G is still a function; there is no paradox here, just a clarification.

That talk of partial functions, computable functions, partial computable functions, and so on can be confusing. The set of all partial functions from $\mathbb{N}$ to $\mathbb{N}$ is a big collection of objects. Some of them are total, some of them are computable, some are both total and computable, and some are neither. Keep in mind that when we say "function," by default, we mean a total function. Thus we have:

1. computable functions
2. partial computable functions that are not total
3. functions that are not computable
4. partial functions that are neither total nor computable

To sort this out, it might help to draw a big square representing all the partial functions from $\mathbb{N}$ to $\mathbb{N}$, and then mark off two overlapping regions, corresponding to the total functions and the computable partial functions, respectively. It is a good exercise to see if you can describe an object in each of the resulting regions in the diagram.

---

content/computability/computability-theory/computable-sets.tex

30.9 Computable Sets

cmp:thy:cps:sec

We can extend the notion of computability from computable functions to computable sets:

Definition 30.7. Let S be a set of natural numbers. Then S is *computable* iff its characteristic function χ_S is. In other words, S is computable iff the function

$$\chi_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases}$$

is computable. Similarly, a relation $R(x_0, \ldots, x_{k-1})$ is computable if and only if its characteristic function is.

Computable sets and relations are also called *decidable*.

Notice that we now have a number of notions of computability: for partial functions, for functions, and for sets. Do not get them confused! The Turing machine computing a partial function returns the output of the function, for input values at which the function is defined; the Turing machine computing a set returns either 1 or 0, after deciding whether or not the input value is in the set or not. explanation

---

content/computability/computability-theory/ce-sets.tex

30.10 Computably Enumerable Sets

cmp:thy:ces:sec

Definition 30.8. A set is *computably enumerable* if it is empty or the range of a computable function.

Historical Remarks Computably enumerable sets are also called *recursively enumerable* instead. This is the original terminology, and today both are commonly used, as well as the abbreviations "c.e." and "r.e."

You should think about what the definition means, and why the terminology is appropriate. The idea is that if S is the range of the computable function f, then explanation

$$S = \{f(0), f(1), f(2), \ldots\},$$

and so f can be seen as "enumerating" the elements of S. Note that according to the definition, f need not be an increasing function, i.e., the enumeration need not be in increasing order. In fact, f need not even be injective, i.e., repetitions in the enumeration $f(0)$, $f(1)$, $f(2)$, ... of S are allowed. For instance, the constant function $f(x) = 0$ enumerates the set $\{0\}$.

Any computable set is computably enumerable. To see this, suppose S is computable. If S is empty, then by definition it is computably enumerable. Otherwise, let a be any element of S. Define f by

$$f(x) = \begin{cases} x & \text{if } \chi_S(x) = 1 \\ a & \text{otherwise.} \end{cases}$$

Then f is a computable function, and S is the range of f.

---

content/computability/computability-theory/equiv-ce-defs.tex

30.11 Equivalent Defininitions of Computably Enumerable Sets

The following gives a number of important equivalent statements of what it means to be computably enumerable. cmp:thy:eqc:sec

Theorem 30.9. *Let S be a set of natural numbers. Then the following are equivalent:* cmp:thy:eqc:thm:ce-equiv

1. *S is computably enumerable.* cmp:thy:eqc:case:ce
2. *S is the range of a* partial *computable function.* cmp:thy:eqc:case:ran-pc
3. *S is empty or the range of a primitive recursive function.* cmp:thy:eqc:case:ran-prim
4. *S is the* domain *of a partial computable function.* cmp:thy:eqc:case:ce-domain

explanation The first three clauses say that we can equivalently take any non-empty computably enumerable set to be enumerated by either a computable function, a partial computable function, or a primitive recursive function. The fourth clause tells us that if S is computably enumerable, then for some index e,

$$S = \{x : \varphi_e(x)\downarrow\}.$$

In other words, S is the set of inputs on for which the computation of φ_e halts. For that reason, computably enumerable sets are sometimes called *semi-decidable*: if a number is in the set, you eventually get a "yes," but if it isn't, you never get a "no"!

Proof. Since every primitive recursive function is computable and every computable function is partial computable, (3) implies (1) and (1) implies (2). (Note that if S is empty, S is the range of the partial computable function that

is nowhere defined.) If we show that (2) implies (3), we will have shown the first three clauses equivalent.

So, suppose S is the range of the partial computable function φ_e. If S is empty, we are done. Otherwise, let a be any element of S. By Kleene's normal form theorem, we can write

$$\varphi_e(x) = U(\mu s\ T(e, x, s)).$$

In particular, $\varphi_e(x) \downarrow$ and $= y$ if and only if there is an s such that $T(e, x, s)$ and $U(s) = y$. Define $f(z)$ by

$$f(z) = \begin{cases} U((z)_1) & \text{if } T(e, (z)_0, (z)_1) \\ a & \text{otherwise.} \end{cases}$$

Then f is primitive recursive, because T and U are. Expressed in terms of Turing machines, if z codes a pair $\langle (z)_0, (z)_1 \rangle$ such that $(z)_1$ is a halting computation of machine M_e on input $(z)_0$, then f returns the output of the computation; otherwise, it returns a.We need to show that S is the range of f, i.e., for any natural number y, $y \in S$ if and only if it is in the range of f. In the forwards direction, suppose $y \in S$. Then y is in the range of φ_e, so for some x and s, $T(e, x, s)$ holds and $U(s) = y$. But then $y = f(\langle x, s \rangle)$. Conversely, suppose y is in the range of f. Then either $y = a$, or for some z, $T(e, (z)_0, (z)_1)$ and $U((z)_1) = y$. Since, in the latter case, $\varphi_e(x) \downarrow = y$, either way, y is in S.

(The notation $\varphi_e(x) \downarrow = y$ means "$\varphi_e(x)$ is defined and equal to y." We could just as well use $\varphi_e(x) = y$, but the extra arrow is sometimes helpful in reminding us that we are dealing with a partial function.)

To finish up the proof of Theorem 30.9, it suffices to show that (1) and (4) are equivalent. First, let us show that (1) implies (4). Suppose S is the range of a computable function f, i.e.,

$$S = \{y : \text{for some } x,\ f(x) = y\}.$$

Let

$$g(y) = \mu x\ (f(x) = y).$$

Then g is a partial computable function, and $g(y)$ is defined if and only if for some x, $f(x) = y$. In other words, the domain of g is the range of f. Expressed in terms of Turing machines: given a Turing machine F that enumerates the elements of S, let G be the Turing machine that semi-decides S by searching through the outputs of F to see if a given element is in the set, halts if it is and keeps searching forever if it isn't.

Finally, to show (4) implies (1), suppose that S is the domain of the partial computable function φ_e, i.e.,

$$S = \{x : \varphi_e(x) \downarrow\}.$$

If S is empty, we are done; otherwise, let a be any element of S. Define f by

$$f(z) = \begin{cases} (z)_0 & \text{if } T(e, (z)_0, (z)_1) \\ a & \text{otherwise.} \end{cases}$$

Then, as above, a number x is in the range of f if and only if $\varphi_e(x)\downarrow$, i.e., if and only if $x \in S$. Expressed in terms of Turing machines: given a machine M_e that semi-decides S, enumerate the elements of S by running through all possible Turing machine computations, and returning the inputs that correspond to halting computations. □

Clause (4) of Theorem 30.9 provides us with a convenient way of enumerating the computably enumerable sets: for each e, let W_e denote the domain of φ_e, i.e.,

$$W_e = \{x : \varphi_e(x)\downarrow\}.$$

Then if A is any computably enumerable set, $A = W_e$, for some e.

The following provides yet another characterization of the computably enumerable sets.

cmp:thy:eqc:thm:exists-char

Theorem 30.10. *A set S is computably enumerable if and only if there is a computable relation $R(x, y)$ such that*

$$S = \{x : \exists y\, R(x, y)\}.$$

Proof. In the forward direction, suppose S is computably enumerable. Then for some e, $S = W_e$. For this value of e we can write S as

$$S = \{x : \exists y\, T(e, x, y)\}.$$

In the reverse direction, suppose $S = \{x : \exists y\, R(x, y)\}$. Define f by

$$f(x) \simeq \mu y\ R(x, y).$$

Then f is partial computable, and S is the domain of f. □

---

content/computability/computability-theory/non-comp-set.tex

30.12 There Are Non-Computable Sets

cmp:thy:ncp:sec

We saw above that every computable set is computably enumerable. Is the converse true? The following shows that, in general, it is not.

cmp:thy:ncp:thm:K-0

Theorem 30.11. *Let K_0 be the set $\{\langle e, x\rangle : \varphi_e(x)\downarrow\}$. Then K_0 is computably enumerable but not computable.*

Proof. To see that K_0 is computably enumerable, note that it is the domain of the function f defined by

$$f(z) = \mu y\ (\mathrm{len}(z) = 2 \wedge T((z)_0, (z)_1, y)).$$

For, if $\varphi_e(x)$ is defined, $f(\langle e, x\rangle)$ finds a halting computation sequence; if $\varphi_e(x)$ is undefined, so is $f(\langle e, x\rangle)$; and if z doesn't even code a pair, then $f(z)$ is also undefined.

The fact that K_0 is not computable is just the undecidability of the halting problem, Theorem 30.6. □

The set K_0 is the set of pairs $\langle e, x\rangle$ such that $\varphi_e(x)\downarrow$, i.e., $\langle e, x\rangle \in K_0$ iff φ_e is defined (halts) on input x, so it is also called the "halting set." The set $K = \{e : \varphi_e(e)\downarrow\}$ is the "self-halting set." It is often used as a canonical undecidable set.

cmp:thy:ncp:thm:K **Theorem 30.12.** *The self-halting set $K = \{e : \varphi_e(e)\downarrow\}$ is c.e. but not decidable.*

Proof. Suppose K is decidable, i.e., its characteristic function χ_K is computable. Let

$$d(e) = \begin{cases} 1 & \text{if } \chi_K(e) = 0 \\ \uparrow & \text{otherwise.} \end{cases}$$

Let k be the index of d, i.e., $d \simeq \varphi_k$. Then $d(k) \simeq \varphi_k(k)$. This contradicts the fact that $d(k)\downarrow$ iff $\varphi_k(k)\uparrow$, which follows from the definition of d.

K is the domain of $f(x) = \mu y\, T(x, x, y)$ and so is c.e. □

---

content/computability/computability-theory/ce-closed-cup-cap.tex

30.13 Computably Enumerable Sets are Closed under Union and Intersection

cmp:thy:clo:sec The following theorem gives some closure properties on the set of computably enumerable sets.

Theorem 30.13. *Suppose A and B are computably enumerable. Then so are $A \cap B$ and $A \cup B$.*

Proof. Theorem 30.9 allows us to use various characterizations of the computably enumerable sets. By way of illustration, we will provide a few different proofs.

For the first proof, suppose A is enumerated by a computable function f, and B is enumerated by a computable function g. Let

$$h(x) = \mu y\ (f(y) = x \vee g(y) = x) \text{ and}$$
$$j(x) = \mu y\ (f((y)_0) = x \wedge g((y)_1) = x).$$

Then $A \cup B$ is the domain of h, and $A \cap B$ is the domain of j.

Here is what is going on, in computational terms: given procedures that enumerate A and B, we can semi-decide if an element x is in $A \cup B$ by looking for x in either enumeration; and we can semi-decide if an element x is in $A \cap B$ for looking for x in both enumerations at the same time. explanation

For the second proof, suppose again that A is enumerated by f and B is enumerated by g. Let

$$k(x) = \begin{cases} f(x/2) & \text{if } x \text{ is even} \\ g((x-1)/2) & \text{if } x \text{ is odd.} \end{cases}$$

Then k enumerates $A \cup B$; the idea is that k just alternates between the enumerations offered by f and g. Enumerating $A \cap B$ is tricker. If $A \cap B$ is empty, it is trivially computably enumerable. Otherwise, let c be any element of $A \cap B$, and define l by

$$l(x) = \begin{cases} f((x)_0) & \text{if } f((x)_0) = g((x)_1) \\ c & \text{otherwise.} \end{cases}$$

In computational terms, l runs through pairs of elements in the enumerations of f and g, and outputs every match it finds; otherwise, it just stalls by outputting c.

For the last proof, suppose A is the *domain* of the partial function $m(x)$ and B is the domain of the partial function $n(x)$. Then $A \cap B$ is the domain of the partial function $m(x) + n(x)$.

explanation In computational terms, if A is the set of values for which m halts and B is the set of values for which n halts, $A \cap B$ is the set of values for which both procedures halt.

Expressing $A \cup B$ as a set of halting values is more difficult, because one has to simulate m and n in parallel. Let d be an index for m and let e be an index for n; in other words, $m = \varphi_d$ and $n = \varphi_e$. Then $A \cup B$ is the domain of the function

$$p(x) = \mu y \, (T(d, x, y) \vee T(e, x, y)).$$

explanation In computational terms, on input x, p searches for either a halting computation for m or a halting computation for n, and halts if it finds either one. □

---

content/computability/computability-theory/complement-ce.tex

30.14 Computably Enumerable Sets not Closed under Complement

Suppose A is computably enumerable. Is the complement of A, $\overline{A} = \mathbb{N} \setminus A$, always computably enumerable as well? The following theorem and corollary show that the answer is "no." cmp:thy:cmp:sec

Theorem 30.14. *Let A be any set of natural numbers. Then A is computable if and only if both A and $\overline{A}$ are computably enumerable.* cmp:thy:cmp:thm:ce-comp

Proof. The forwards direction is easy: if A is computable, then $\overline{A}$ is computable as well ($\chi_A = 1 \dot{-} \chi_{\overline{A}}$), and so both are computably enumerable.

In the other direction, suppose A and $\overline{A}$ are both computably enumerable. Let A be the domain of φ_d, and let $\overline{A}$ be the domain of φ_e. Define h by

$$h(x) = \mu s \, (T(d, x, s) \vee T(e, x, s)).$$

In other words, on input x, h searches for either a halting computation of φ_d or a halting computation of φ_e. Now, if $x \in A$, it will succeed in the first case,

and if $x \in \overline{A}$, it will succeed in the second case. So, h is a total computable function. But now we have that for every x, $x \in A$ if and only if $T(e, x, h(x))$, i.e., if φ_e is the one that is defined. Since $T(e, x, h(x))$ is a computable relation, A is computable. □

It is easier to understand what is going on in informal computational terms: explanation
to decide A, on input x search for halting computations of φ_e and φ_f. One of them is bound to halt; if it is φ_e, then x is in A, and otherwise, x is in $\overline{A}$.

cmp:thy:cmp:cor:comp-k **Corollary 30.15.** *$\overline{K_0}$ is not computably enumerable.*

Proof. We know that K_0 is computably enumerable, but not computable. If $\overline{K_0}$ were computably enumerable, then K_0 would be computable by Theorem 30.14, contradicting Theorem 30.11. □

---

content/computability/computability-theory/reducibility.tex

30.15 Reducibility

cmp:thy:red:sec We now know that there is at least one set, K_0, that is computably enumerable but not computable. It should be clear that there are others. The method of reducibility provides a powerful method of showing that other sets have these properties, without constantly having to return to first principles. explanation

Generally speaking, a "reduction" of a set A to a set B is a method of transforming answers to whether or not elements are in B into answers as to whether or not elements are in A. We will focus on a notion called "many-one reducibility," but there are many other notions of reducibility available, with varying properties. Notions of reducibility are also central to the study of computational complexity, where efficiency issues have to be considered as well. For example, a set is said to be "NP-complete" if it is in NP and every NP problem can be reduced to it, using a notion of reduction that is similar to the one described below, only with the added requirement that the reduction can be computed in polynomial time.

We have already used the notion of reduction notion implicitly. Define the set K by

$$K = \{x : \varphi_x(x) \downarrow\},$$

i.e., $K = \{x : x \in W_x\}$. Our proof that the halting problem in unsolvable (Theorem 30.6) shows most directly that K is not computable. Recall that K_0 is the set

$$K_0 = \{\langle e, x\rangle : \varphi_e(x) \downarrow\},$$

i.e., $K_0 = \{\langle x, e\rangle : x \in W_e\}$. It is easy to extend any proof of the uncomputability of K to the uncomputability of K_0: if K_0 were computable, we could decide whether or not an element x is in K simply by asking whether or not the pair $\langle x, x\rangle$ is in K_0. The function f which maps x to $\langle x, x\rangle$ is an example of a *reduction* of K to K_0.

Definition 30.16. Let A and B be sets of natural numbers. A computable function $f\colon \mathbb{N} \to \mathbb{N}$ is a *many-one reduction* of A to B iff, for every natural number x,

$$x \in A \quad \text{if and only if} \quad f(x) \in B.$$

If such a reduction f exists, we say that A is *many-one reducible* to B, written $A \leq_m B$. If A is many-one reducible to B and vice-versa, then A and B are said to be *many-one equivalent*, written $A \equiv_m B$.

If the function f in the definition above happens to be injective, A is said to be *one-one reducible* to B. Most of the reductions described below meet this stronger requirement, but we will not use this fact.

digression

It is true, but by no means obvious, that one-one reducibility really is a stronger requirement than many-one reducibility. In other words, there are infinite sets A and B such that A is many-one reducible to B but not one-one reducible to B.

---

content/computability/computability-theory/prop-reduce.tex

30.16 Properties of Reducibility

We write $A \leq_m B$ if A reduces to B, and this notation suggests that if $A \leq_m B$, then A is "no harder than" B and that B is "as hard or harder than" A, and that $\leq_m$, like the usual $\leq$ on numbers, orders sets (or decision problems) by their complexity. The following two propositions support this intuition. The first one say taht $\leq_m$ is transitive.

cmp:thy:ppr:sec

Proposition 30.17. *If $A \leq_m B$ and $B \leq_m C$, then $A \leq_m C$.*

cmp:thy:ppr:prop:trans-red

Proof. Composing a reduction of A to B with a reduction of B to C yields a reduction of A to C. □

Problem 30.2. Prove Proposition 30.17 by showing that if f and g are many-one reductions of A to B and B to C, respectively, then $g \circ f$ is a many-one reduction of A to C.

Proposition 30.18. *Let A and B be any sets, and suppose $A \leq_m B$.*

cmp:thy:ppr:prop:reduce

1. *If B is computably enumerable, so is A.*
2. *If B is computable, so is A.*

Proof. Let f be a many-one reduction from A to B. For the first claim, just check that if B is the domain of a partial function g, then A is the domain of $g \circ f$:

$$\begin{aligned} x \in A &\text{ iff } f(x) \in B \\ &\text{iff } g(f(x)) \downarrow . \end{aligned}$$

For the second claim, remember that if B is computable then B and $\overline{B}$ are computably enumerable (Theorem 30.14). It is not hard to check that f is also a many-one reduction of $\overline{A}$ to $\overline{B}$, so, by the first part of this proof, A and $\overline{A}$ are computably enumerable. So A is computable as well by Theorem 30.14. (Alternatively, you can check that $\chi_A = \chi_B \circ f$; so if χ_B is computable, then so is χ_A.) □

Problem 30.3. Suppose f is a many-one reduction of A to B. Show that f is also a many-one reduction of $\overline{A}$ to $\overline{B}$.

Problem 30.4. Show that if $f\colon A \to B$ is a many-one reduction, then $\chi_A = \chi_B \circ f$.

digression

A more general notion of reducibility called *Turing reducibility* is useful in other contexts, especially for proving undecidability results. Note that by Corollary 30.15, the complement of K_0 is not reducible to K_0, since it is not computably enumerable. But, intuitively, if you knew the answers to questions about K_0, you would know the answer to questions about its complement as well. A set A is said to be Turing reducible to B if one can determine answers to questions in A using a computable procedure that can ask questions about B. This is more liberal than many-one reducibility, in which (1) you are only allowed to ask one question about B, and (2) a "yes" answer has to translate to a "yes" answer to the question about A, and similarly for "no." It is still the case that if A is Turing reducible to B and B is computable then A is computable as well (though, as we have seen, the analogous statement does not hold for computable enumerability).

You should think about the various notions of reducibility we have discussed, and understand the distinctions between them. We will, however, only deal with many-one reducibility in this chapter. Incidentally, both types of reducibility discussed in the last paragraph have analogues in computational complexity, with the added requirement that the Turing machines run in polynomial time: the complexity version of many-one reducibility is known as *Karp reducibility*, while the complexity version of Turing reducibility is known as *Cook reducibility*.

---

content/computability/computability-theory/complete-ce-sets.tex

30.17 Complete Computably Enumerable Sets

cmp:thy:cce:sec

Definition 30.19. A set A is a *complete computably enumerable set* (under many-one reducibility) if

1. A is computably enumerable, and
2. for any other computably enumerable set B, $B \leq_m A$.

In other words, complete computably enumerable sets are the "hardest" computably enumerable sets possible. They allow one to answer questions about *any* computably enumerable set.

Theorem 30.20. *K, K_0, and K_1 are all complete computably enumerable sets.*

Proof. To see that K_0 is complete, let B be any computably enumerable set. Then for some index e,

$$B = W_e = \{x : \varphi_e(x)\downarrow\}.$$

Let f be the function $f(x) = \langle e, x\rangle$. Then for every natural number x, $x \in B$ if and only if $f(x) \in K_0$. In other words, f reduces B to K_0.

To see that K_1 is complete, note that in the proof of Proposition 30.21 we reduced K_0 to it. So, by Proposition 30.17, any computably enumerable set can be reduced to K_1 as well.

K can be reduced to K_0 in much the same way. □

Problem 30.5. Give a reduction of K to K_0.

digression

So, it turns out that all the examples of computably enumerable sets that we have considered so far are either computable, or complete. This should seem strange! Are there any examples of computably enumerable sets that are neither computable nor complete? The answer is yes, but it wasn't until the middle of the 1950s that this was established by Friedberg and Muchnik, independently.

---

content/computability/computability-theory/k-1.tex

30.18 An Example of Reducibility

Let us consider an application of Proposition 30.18.

cmp:thy:k1:sec

Proposition 30.21. *Let*

cmp:thy:k1:prop:k1

$$K_1 = \{e : \varphi_e(0)\downarrow\}.$$

Then K_1 is computably enumerable but not computable.

Proof. Since $K_1 = \{e : \exists s\, T(e, 0, s)\}$, K_1 is computably enumerable by Theorem 30.10.

To show that K_1 is not computable, let us show that K_0 is reducible to it.

explanation

This is a little bit tricky, since using K_1 we can only ask questions about computations that start with a particular input, 0. Suppose you have a smart friend who can answer questions of this type (friends like this are known as "oracles"). Then suppose someone comes up to you and asks you whether or not $\langle e, x\rangle$ is in K_0, that is, whether or not machine e halts on input x. One

thing you can do is build another machine, e_x, that, for *any* input, ignores that input and instead runs e on input x. Then clearly the question as to whether machine e halts on input x is equivalent to the question as to whether machine e_x halts on input 0 (or any other input). So, then you ask your friend whether this new machine, e_x, halts on input 0; your friend's answer to the modified question provides the answer to the original one. This provides the desired reduction of K_0 to K_1.

Using the universal partial computable function, let f be the 3-ary function defined by

$$f(x, y, z) \simeq \varphi_x(y).$$

Note that f ignores its third input entirely. Pick an index e such that $f = \varphi^3_e$; so we have

$$\varphi^3_e(x, y, z) \simeq \varphi_x(y).$$

By the *s-m-n* theorem, there is a function $s(e, x, y)$ such that, for every z,

$$\begin{aligned}\varphi_{s(e,x,y)}(z) &\simeq \varphi^3_e(x, y, z)\\ &\simeq \varphi_x(y).\end{aligned}$$

In terms of the informal argument above, $s(e, x, y)$ is an index for the machine that, for any input z, ignores that input and computes $\varphi_x(y)$. explanation

In particular, we have

$$\varphi_{s(e,x,y)}(0)\downarrow \quad \text{if and only if} \quad \varphi_x(y)\downarrow.$$

In other words, $\langle x, y\rangle \in K_0$ if and only if $s(e, x, y) \in K_1$. So the function g defined by

$$g(w) = s(e, (w)_0, (w)_1)$$

is a reduction of K_0 to K_1. □

---

content/computability/computability-theory/total.tex

30.19 Totality is Undecidable

cmp:thy:tot:sec

Let us consider one more example of using the *s-m-n* theorem to show that something is noncomputable. Let Tot be the set of indices of total computable functions, i.e.

$$\text{Tot} = \{x : \text{for every } y,\ \varphi_x(y)\downarrow\}.$$

cmp:thy:tot:prop:total

Proposition 30.22. Tot *is not computable.*

Proof. To see that Tot is not computable, it suffices to show that K is reducible to it. Let $h(x, y)$ be defined by

$$h(x, y) \simeq \begin{cases} 0 & \text{if } x \in K\\ \uparrow & \text{otherwise}\end{cases}$$

Note that $h(x, y)$ does not depend on y at all. It should not be hard to see that h is partial computable: on input x, y, the we compute h by first simulating the function φ_x on input x; if this computation halts, $h(x, y)$ outputs 0 and halts. So $h(x, y)$ is just zero($\mu s\, T(x, x, s)$), where zero is the constant zero function.

Using the *s-m-n* theorem, there is a primitive recursive function $k(x)$ such that for every x and y,

$$\varphi_{k(x)}(y) = \begin{cases} 0 & \text{if } x \in K \\ \uparrow & \text{otherwise} \end{cases}$$

So $\varphi_{k(x)}$ is total if $x \in K$, and undefined otherwise. Thus, k is a reduction of K to Tot. □

digression

It turns out that Tot is not even computably enumerable—its complexity lies further up on the "arithmetical hierarchy." But we will not worry about this strengthening here.

---

content/computability/computability-theory/rice-theorem.tex

30.20 Rice's Theorem

cmp:thy:rce:sec

If you think about it, you will see that the specifics of Tot do not play into the proof of Proposition 30.22. We designed $h(x, y)$ to act like the constant function $j(y) = 0$ exactly when x is in K; but we could just as well have made it act like any other partial computable function under those circumstances. This observation lets us state a more general theorem, which says, roughly, that no nontrivial property of computable functions is decidable.

Keep in mind that $\varphi_0, \varphi_1, \varphi_2, \ldots$ is our standard enumeration of the partial computable functions.

Theorem 30.23 (Rice's Theorem). *Let C be any set of partial computable functions, and let $A = \{n : \varphi_n \in C\}$. If A is computable, then either C is empty or C is the set of all the partial computable functions.*

An *index set* is a set A with the property that if n and m are indices which "compute" the same function, then either both n and m are in A, or neither is. It is not hard to see that the set A in the theorem has this property. Conversely, if A is an index set and C is the set of functions computed by these indices, then $A = \{n : \varphi_n \in C\}$.

explanation

With this terminology, Rice's theorem is equivalent to saying that no nontrivial index set is decidable. To understand what the theorem says, it is helpful to emphasize the distinction between *programs* (say, in your favorite programming language) and the functions they compute. There are certainly questions about programs (indices), which are syntactic objects, that are computable: does this program have more than 150 symbols? Does it have more than 22 lines? Does it have a "while" statement? Does the string "hello world" ever

appear as the argument to a "print" statement? Rice's theorem says that no nontrivial question about the program's *behavior* is computable. This includes questions like these: does the program halt on input 0? Does it ever halt? Does it ever output an even number?

Proof of Rice's theorem. Suppose C is neither empty nor the set of all the partial computable functions, and let A be the set of indices of functions in C. We will show that if A were computable, we could solve the halting problem; so A is not computable.

Without loss of generality, we can assume that the function f which is nowhere defined is not in C (otherwise, switch C and its complement in the argument below). Let g be any function in C. The idea is that if we could decide A, we could tell the difference between indices computing f, and indices computing g; and then we could use that capability to solve the halting problem.

Here's how. Using the universal partial computable functions, we can define a function

$$h(x,y) \simeq \begin{cases} \text{undefined} & \text{if } \varphi_x(x) \uparrow \\ g(y) & \text{otherwise.} \end{cases}$$

To compute h, first we try to compute $\varphi_x(x)$; if that computation halts, we go on to compute $g(y)$; and if *that* computation halts, we return the output. More formally, we can write

$$h(x,y) \simeq P^2_0(g(y), \mathrm{Un}(x,x)).$$

where $P^2_0(z_0, z_1) = z_0$ is the 2-place projection function returning the 0-th argument, which is computable.

Then h is a composition of partial computable functions, and the right side is defined and equal to $g(y)$ just when $\mathrm{Un}(x,x)$ and $g(y)$ are both defined.

Notice that for a fixed x, if $\varphi_x(x)$ is undefined, then $h(x,y)$ is undefined for every y; and if $\varphi_x(x)$ is defined, then $h(x,y) \simeq g(y)$. So, for any fixed value of x, either $h(x,y)$ acts just like f or it acts just like g, and deciding whether or not $\varphi_x(x)$ is defined amounts to deciding which of these two cases holds. But this amounts to deciding whether or not $h_x(y) \simeq h(x,y)$ is in C or not, and if A were computable, we could do just that.

More formally, since h is partial computable, it is equal to the function φ_e for some index e. By the *s-m-n* theorem there is a primitive recursive function s such that for each x, $\varphi_{s(e,x)}(y) = h_x(y)$. Now we have that for each x, if $\varphi_x(x) \downarrow$, then $\varphi_{s(e,x)}$ is the same function as g, and so $s(e,x)$ is in A. On the other hand, if $\varphi_x(x) \uparrow$, then $\varphi_{s(e,x)}$ is the same function as f, and so $s(e,x)$ is not in A. In other words, we have that for every x, $x \in K$ if and only if $s(e,x) \in A$. If A were computable, K would be also, which is a contradiction. So A is not computable. □

Rice's theorem is very powerful. The following immediate corollary shows some sample applications.

Corollary 30.24. *The following sets are undecidable.*

1. $\{x : 17 \text{ is in the range of } \varphi_x\}$
2. $\{x : \varphi_x \text{ is constant}\}$
3. $\{x : \varphi_x \text{ is total}\}$
4. $\{x : \text{whenever } y < y', \varphi_x(y)\downarrow, \text{ and if } \varphi_x(y')\downarrow, \text{ then } \varphi_x(y) < \varphi_x(y')\}$

Proof. These are all nontrivial index sets. □

---

content/computability/computability-theory/fixed-point-thm.tex

30.21 The Fixed-Point Theorem

cmp:thy:fix:sec

Let's consider the halting problem again. As temporary notation, let us write $\ulcorner\varphi_x(y)\urcorner$ for $\langle x, y\rangle$; think of this as representing a "name" for the value $\varphi_x(y)$. With this notation, we can reword one of our proofs that the halting problem is undecidable.

Question: is there a computable function h, with the following property? For every x and y,

$$h(\ulcorner\varphi_x(y)\urcorner) = \begin{cases} 1 & \text{if } \varphi_x(y)\downarrow \\ 0 & \text{otherwise.} \end{cases}$$

Answer: No; otherwise, the partial function

$$g(x) \simeq \begin{cases} 0 & \text{if } h(\ulcorner\varphi_x(x)\urcorner) = 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

would be computable, and so have some index e. But then we have

$$\varphi_e(e) \simeq \begin{cases} 0 & \text{if } h(\ulcorner\varphi_e(e)\urcorner) = 0 \\ \text{undefined} & \text{otherwise,} \end{cases}$$

in which case $\varphi_e(e)$ is defined if and only if it isn't, a contradiction.

Now, take a look at the equation with φ_e. There is an instance of self-reference there, in a sense: we have arranged for the value of $\varphi_e(e)$ to depend on $\ulcorner\varphi_e(e)\urcorner$, in a certain way. The fixed-point theorem says that we *can* do this, in general—not just for the sake of proving contradictions.

Lemma 30.25 gives two equivalent ways of stating the fixed-point theorem. Logically speaking, the fact that the statements are equivalent follows from the fact that they are both true; but what we really mean is that each one follows straightforwardly from the other, so that they can be taken as alternative statements of the same theorem.

Lemma 30.25. *The following statements are equivalent:*

cmp:thy:fix:lem:fixed-equiv

1. *For every partial computable function $g(x, y)$, there is an index e such that for every y,*
$$\varphi_e(y) \simeq g(e, y).$$
2. *For every computable function $f(x)$, there is an index e such that for every y,*
$$\varphi_e(y) \simeq \varphi_{f(e)}(y).$$

Proof. (1) $\Rightarrow$ (2): Given f, define g by $g(x, y) \simeq \mathrm{Un}(f(x), y)$. Use (1) to get an index e such that for every y,

$$\begin{aligned}\varphi_e(y) &= \mathrm{Un}(f(e), y)\\ &= \varphi_{f(e)}(y).\end{aligned}$$

(2) $\Rightarrow$ (1): Given g, use the *s-m-n* theorem to get f such that for every x and y, $\varphi_{f(x)}(y) \simeq g(x, y)$. Use (2) to get an index e such that

$$\begin{aligned}\varphi_e(y) &= \varphi_{f(e)}(y)\\ &= g(e, y).\end{aligned}$$

This concludes the proof. □

explanation

Before showing that statement (1) is true (and hence (2) as well), consider how bizarre it is. Think of e as being a computer program; statement (1) says that given any partial computable $g(x, y)$, you can find a computer program e that computes $g_e(y) \simeq g(e, y)$. In other words, you can find a computer program that computes a function that references the program itself.

Theorem 30.26. *The two statements in Lemma 30.25 are true. Specifically, for every partial computable function $g(x, y)$, there is an index e such that for every y,*
$$\varphi_e(y) \simeq g(e, y).$$

Proof. The ingredients are already implicit in the discussion of the halting problem above. Let $\mathrm{diag}(x)$ be a computable function which for each x returns an index for the function $f_x(y) \simeq \varphi_x(x, y)$, i.e.

$$\varphi_{\mathrm{diag}(x)}(y) \simeq \varphi_x(x, y).$$

Think of diag as a function that transforms a program for a 2-ary function into a program for a 1-ary function, obtained by fixing the original program as its first argument. The function diag can be defined formally as follows: first define s by

$$s(x, y) \simeq \mathrm{Un}^2(x, x, y),$$

where Un^2 is a 3-ary function that is universal for partial computable 2-ary functions. Then, by the *s-m-n* theorem, we can find a primitive recursive function diag satisfying

$$\varphi_{\mathrm{diag}(x)}(y) \simeq s(x, y).$$

Now, define the function l by

$$l(x, y) \simeq g(\text{diag}(x), y).$$

and let $\ulcorner l \urcorner$ be an index for l. Finally, let $e = \text{diag}(\ulcorner l \urcorner)$. Then for every y, we have

$$\begin{aligned}
\varphi_e(y) &\simeq \varphi_{\text{diag}(\ulcorner l \urcorner)}(y) \\
&\simeq \varphi_{\ulcorner l \urcorner}(\ulcorner l \urcorner, y) \\
&\simeq l(\ulcorner l \urcorner, y) \\
&\simeq g(\text{diag}(\ulcorner l \urcorner), y) \\
&\simeq g(e, y),
\end{aligned}$$

as required. □

explanation

What's going on? Suppose you are given the task of writing a computer program that prints itself out. Suppose further, however, that you are working with a programming language with a rich and bizarre library of string functions. In particular, suppose your programming language has a function diag which works as follows: given an input string s, diag locates each instance of the symbol 'x' occurring in s, and replaces it by a quoted version of the original string. For example, given the string

```
hello x world
```

as input, the function returns

```
hello 'hello x world' world
```

as output. In that case, it is easy to write the desired program; you can check that

```
print(diag('print(diag(x))'))
```

does the trick. For more common programming languages like C++ and Java, the same idea (with a more involved implementation) still works.

We are only a couple of steps away from the proof of the fixed-point theorem. Suppose a variant of the print function $\text{print}(x, y)$ accepts a string x and another numeric argument y, and prints the string x repeatedly, y times. Then the "program"

```
getinput(y);
print(diag('getinput(y); print(diag(x), y)'), y)
```

prints itself out y times, on input y. Replacing the getinput—print—diag skeleton by an arbitrary function $g(x, y)$ yields

```
g(diag('g(diag(x), y)'), y)
```

which is a program that, on input y, runs g on the program itself and y. Thinking of "quoting" with "using an index for," we have the proof above.

For now, it is o.k. if you want to think of the proof as formal trickery, or black magic. But you should be able to reconstruct the details of the argument given above. When we prove the incompleteness theorems (and the related "fixed-point theorem") we will discuss other ways of understanding why it works.

digression

The same idea can be used to get a "fixed point" combinator. Suppose you have a lambda term g, and you want another term k with the property that k is β-equivalent to gk. Define terms

$$\mathrm{diag}(x) = xx$$

and

$$l(x) = g(\mathrm{diag}(x))$$

using our notational conventions; in other words, l is the term $\lambda x.\, g(xx)$. Let k be the term ll. Then we have

$$\begin{aligned} k &= (\lambda x.\, g(xx))(\lambda x.\, g(xx)) \\ &\twoheadrightarrow g((\lambda x.\, g(xx))(\lambda x.\, g(xx))) \\ &= gk. \end{aligned}$$

If one takes

$$Y = \lambda g.\, ((\lambda x.\, g(xx))(\lambda x.\, g(xx)))$$

then Yg and $g(Yg)$ reduce to a common term; so $Yg \equiv_\beta g(Yg)$. This is known as "Curry's combinator." If instead one takes

$$Y = (\lambda xg.\, g(xxg))(\lambda xg.\, g(xxg))$$

then in fact Yg reduces to $g(Yg)$, which is a stronger statement. This latter version of Y is known as "Turing's combinator."

---

content/computability/computability-theory/application-fixed-point.tex

30.22 Applying the Fixed-Point Theorem

cmp:thy:apf:sec

The fixed-point theorem essentially lets us define partial computable functions in terms of their indices. For example, we can find an index e such that for every y,

$$\varphi_e(y) = e + y.$$

As another example, one can use the proof of the fixed-point theorem to design a program in Java or C++ that prints itself out.

Remember that if for each e, we let W_e be the domain of φ_e, then the sequence W_0, W_1, W_2, ... enumerates the computably enumerable sets. Some of these sets are computable. One can ask if there is an algorithm which takes as input a value x, and, if W_x happens to be computable, returns an index for its characteristic function. The answer is "no," there is no such algorithm:

Theorem 30.27. *There is no partial computable function f with the following property: whenever W_e is computable, then $f(e)$ is defined and $\varphi_{f(e)}$ is its characteristic function.*

Proof. Let f be any computable function; we will construct an e such that W_e is computable, but $\varphi_{f(e)}$ is not its characteristic function. Using the fixed point theorem, we can find an index e such that

$$\varphi_e(y) \simeq \begin{cases} 0 & \text{if } y = 0 \text{ and } \varphi_{f(e)}(0) \downarrow = 0 \\ \text{undefined} & \text{otherwise.} \end{cases}$$

That is, e is obtained by applying the fixed-point theorem to the function defined by

$$g(x, y) \simeq \begin{cases} 0 & \text{if } y = 0 \text{ and } \varphi_{f(x)}(0) \downarrow = 0 \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Informally, we can see that g is partial computable, as follows: on input x and y, the algorithm first checks to see if y is equal to 0. If it is, the algorithm computes $f(x)$, and then uses the universal machine to compute $\varphi_{f(x)}(0)$. If this last computation halts and returns 0, the algorithm returns 0; otherwise, the algorithm doesn't halt.

But now notice that if $\varphi_{f(e)}(0)$ is defined and equal to 0, then $\varphi_e(y)$ is defined exactly when y is equal to 0, so $W_e = \{0\}$. If $\varphi_{f(e)}(0)$ is not defined, or is defined but not equal to 0, then $W_e = \emptyset$. Either way, $\varphi_{f(e)}$ is not the characteristic function of W_e, since it gives the wrong answer on input 0. □

---

content/computability/computability-theory/def-functions-self-reference.tex

30.23 Defining Functions using Self-Reference

cmp:thy:slf:sec

It is generally useful to be able to define functions in terms of themselves. For example, given computable functions k, l, and m, the fixed-point lemma tells us that there is a partial computable function f satisfying the following equation for every y:

$$f(y) \simeq \begin{cases} k(y) & \text{if } l(y) = 0 \\ f(m(y)) & \text{otherwise.} \end{cases}$$

Again, more specifically, f is obtained by letting

$$g(x, y) \simeq \begin{cases} k(y) & \text{if } l(y) = 0 \\ \varphi_x(m(y)) & \text{otherwise} \end{cases}$$

and then using the fixed-point lemma to find an index e such that $\varphi_e(y) = g(e, y)$.

For a concrete example, the "greatest common divisor" function $\gcd(u, v)$ can be defined by

$$\gcd(u, v) \simeq \begin{cases} v & \text{if } u = 0 \\ \gcd(\mathrm{mod}(v, u), u) & \text{otherwise} \end{cases}$$

where $\mathrm{mod}(v, u)$ denotes the remainder of dividing v by u. An appeal to the fixed-point lemma shows that gcd is partial computable. (In fact, this can be put in the format above, letting y code the pair $\langle u, v \rangle$.) A subsequent induction on u then shows that, in fact, gcd is total.

Of course, one can cook up self-referential definitions that are much fancier than the examples just discussed. Most programming languages support definitions of functions in terms of themselves, one way or another. Note that this is a little bit less dramatic than being able to define a function in terms of an *index* for an algorithm computing the functions, which is what, in full generality, the fixed-point theorem lets you do.

Part VI

Turing Machines

Chapter 31

Turing Machine Computations

content/turing-machines/machines-computations/introduction.tex

31.1 Introduction

tur:mac:int:sec

What does it mean for a function, say, from $\mathbb{N}$ to $\mathbb{N}$ to be *computable*? Among the first answers, and the most well known one, is that a function is computable if it can be computed by a Turing machine. This notion was set out by Alan Turing in 1936. Turing machines are an example of *a model of computation*—they are a mathematically precise way of defining the idea of a "computational procedure." What exactly that means is debated, but it is widely agreed that Turing machines are one way of specifying computational procedures. Even though the term "Turing machine" evokes the image of a physical machine with moving parts, strictly speaking a Turing machine is a purely mathematical construct, and as such it idealizes the idea of a computational procedure. For instance, we place no restriction on either the time or memory requirements of a Turing machine: Turing machines can compute something even if the computation would require more storage space or more steps than there are atoms in the universe.

explanation

It is perhaps best to think of a Turing machine as a program for a special kind of imaginary mechanism. This mechanism consists of a *tape* and a *read-write head*. In our version of Turing machines, the tape is infinite in one direction (to the right), and it is divided into *squares*, each of which may contain a symbol from a finite *alphabet*. Such alphabets can contain any number of different symbols, but we will mainly make do with three: $\triangleright$, 0, and 1. When the mechanism is started, the tape is empty (i.e., each square contains the symbol 0) except for the leftmost square, which contains $\triangleright$, and a finite number

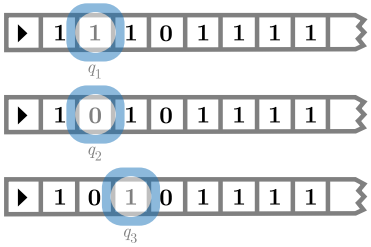


Figure 31.1: A Turing machine executing its program.

tur:mac:int:
fig:tm

of squares which contain the *input*. At any time, the mechanism is in one of a finite number of *states*. At the outset, the head scans the leftmost square and in a specified *initial state*. At each step of the mechanism's run, the content of the square currently scanned together with the state the mechanism is in and the Turing machine program determine what happens next. The Turing machine program is given by a partial function which takes as input a state q and a symbol σ and outputs a triple $\langle q', \sigma', D\rangle$. Whenever the mechanism is in state q and reads symbol σ, it replaces the symbol on the current square with σ', the head moves left, right, or stays put according to whether D is L, R, or N, and the mechanism goes into state q'.

For instance, consider the situation in Figure 31.1. The visible part of the tape of the Turing machine contains the end-of-tape symbol $\triangleright$ on the leftmost square, followed by three 1's, a 0, and four more 1's. The head is reading the third square from the left, which contains a 1, and is in state q_1—we say "the machine is reading a 1 in state q_1." If the program of the Turing machine returns, for input $\langle q_1, 1\rangle$, the triple $\langle q_2, 0, N\rangle$, the machine would now replace the 1 on the third square with a 0, leave the read/write head where it is, and switch to state q_2. If then the program returns $\langle q_3, 0, R\rangle$ for input $\langle q_2, 0\rangle$, the machine would now overwrite the 0 with another 0 (effectively, leaving the content of the tape under the read/write head unchanged), move one square to the right, and enter state q_3. And so on.

We say that the machine *halts* when it encounters some state, q_n, and symbol, σ such that there is no instruction for $\langle q_n, \sigma\rangle$, i.e., the transition function for input $\langle q_n, \sigma\rangle$ is undefined. In other words, the machine has no instruction to carry out, and at that point, it ceases operation. Halting is sometimes represented by a specific halt state h. This will be demonstrated in more detail later on.

The beauty of Turing's paper, "On computable numbers," is that he presents not only a formal definition, but also an argument that the definition captures the intuitive notion of computability. From the definition, it should be clear that any function computable by a Turing machine is computable in the intuitive sense. Turing offers three types of argument that the converse is true, i.e., that any function that we would naturally regard as computable is computable by such a machine. They are (in Turing's words):

digression

1. A direct appeal to intuition.
2. A proof of the equivalence of two definitions (in case the new definition has a greater intuitive appeal).
3. Giving examples of large classes of numbers which are computable.

Our goal is to try to define the notion of computability "in principle," i.e., without taking into account practical limitations of time and space. Of course, with the broadest definition of computability in place, one can then go on to consider computation with bounded resources; this forms the heart of the subject known as "computational complexity."

Historical Remarks Alan Turing invented Turing machines in 1936. While his interest at the time was the decidability of first-order logic, the paper has been described as a definitive paper on the foundations of computer design. In the paper, Turing focuses on computable real numbers, i.e., real numbers whose decimal expansions are computable; but he notes that it is not hard to adapt his notions to computable functions on the natural numbers, and so on. Notice that this was a full five years before the first working general purpose computer was built in 1941 (by the German Konrad Zuse in his parent's living room), seven years before Turing and his colleagues at Bletchley Park built the code-breaking Colossus (1943), nine years before the American ENIAC (1945), twelve years before the first British general purpose computer—the Manchester Small-Scale Experimental Machine—was built in Manchester (1948), and thirteen years before the Americans first tested the BINAC (1949). The Manchester SSEM has the distinction of being the first stored-program computer—previous machines had to be rewired by hand for each new task.

---

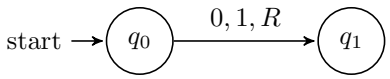
content/turing-machines/machines-computations/representing-tms.tex

31.2 Representing Turing Machines

explanation

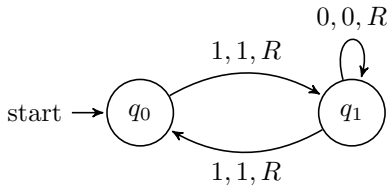
Turing machines can be represented visually by *state diagrams*. The diagrams are composed of state cells connected by arrows. Unsurprisingly, each state cell represents a state of the machine. Each arrow represents an instruction that can be carried out from that state, with the specifics of the instruction written above or below the appropriate arrow. Consider the following machine, which has only two internal states, q_0 and q_1, and one instruction:

tur:mac:rep:sec



Recall that the Turing machine has a read/write head and a tape with the input written on it. The instruction can be read as *if reading a* 0 *in state* q_0*, write a* 1*, move right, and move to state* q_1. This is equivalent to the transition function mapping $\langle q_0, 0\rangle$ to $\langle q_1, 1, R\rangle$.

Example 31.1. *Even Machine*: The following Turing machine halts if, and only if, there are an even number of 1's on the tape (under the assumption that all 1's come before the first 0 on the tape).



The state diagram corresponds to the following transition function:

$$\delta(q_0, 1) = \langle q_1, 1, R\rangle,$$
$$\delta(q_1, 1) = \langle q_0, 1, R\rangle,$$
$$\delta(q_1, 0) = \langle q_1, 0, R\rangle$$

The above machine halts only when the input is an even number of strokes. explanation Otherwise, the machine (theoretically) continues to operate indefinitely. For any machine and input, it is possible to trace through the *configurations* of the machine in order to determine the output. We will give a formal definition of configurations later. For now, we can intuitively think of configurations as a series of diagrams showing the state of the machine at any point in time during operation. Configurations show the content of the tape, the state of the machine and the location of the read/write head.

Let us trace through the configurations of the even machine if it is started with an input of four 1's. In this case, we expect that the machine will halt. We will then run the machine on an input of three 1's, where the machine will run forever.

The machine starts in state q_0, scanning the leftmost 1. We can represent the initial state of the machine as follows:

$$\triangleright 1_0 1110\ldots$$

The above configuration is straightforward. As can be seen, the machine starts in state one, scanning the leftmost 1. This is represented by a subscript of the state name on the first 1. The applicable instruction at this point is $\delta(q_0, 1) = \langle q_1, 1, R\rangle$, and so the machine moves right on the tape and changes to state q_1.

$$\triangleright 11_1 110\ldots$$

Since the machine is now in state q_1 scanning a 1, we have to "follow" the instruction $\delta(q_1, 1) = \langle q_0, 1, R\rangle$. This results in the configuration

$$\triangleright 111_0 10\ldots$$

As the machine continues, the rules are applied again in the same order, resulting in the following two configurations:

$$\triangleright 1111_1 0\ldots$$

$$\triangleright 11110_0 \ldots$$

The machine is now in state q_0 scanning a 0. Based on the transition diagram, we can easily see that there is no instruction to be carried out, and thus the machine has halted. This means that the input has been accepted.

Suppose next we start the machine with an input of three 1's. The first few configurations are similar, as the same instructions are carried out, with only a small difference of the tape input:

$$\triangleright 1_0 110 \ldots$$

$$\triangleright 11_1 10 \ldots$$

$$\triangleright 111_0 0 \ldots$$

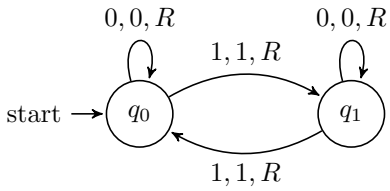
$$\triangleright 1110_1 \ldots$$

The machine has now traversed past all the 1's, and is reading a 0 in state q_1. As shown in the diagram, there is an instruction of the form $\delta(q_1, 0) = \langle q_1, 0, R \rangle$. Since the tape is filled with 0 indefinitely to the right, the machine will continue to execute this instruction *forever*, staying in state q_1 and moving ever further to the right. The machine will never halt, and does not accept the input.

explanation

It is important to note that not all machines will halt. If halting means that the machine runs out of instructions to execute, then we can create a machine that never halts simply by ensuring that there is an outgoing arrow for each symbol at each state. The even machine can be modified to run indefinitely by adding an instruction for scanning a 0 at q_0.

Example 31.2.



explanation

Machine tables are another way of representing Turing machines. Machine tables have the tape alphabet displayed on the x-axis, and the set of machine states across the y-axis. Inside the table, at the intersection of each state and symbol, is written the rest of the instruction—the new state, new symbol, and direction of movement. Machine tables make it easy to determine in what state, and for what symbol, the machine halts. Whenever there is a gap in the table is a possible point for the machine to halt. Unlike state diagrams and instruction sets, where the points at which the machine halts are not always immediately obvious, any halting points are quickly identified by finding the gaps in the machine table.

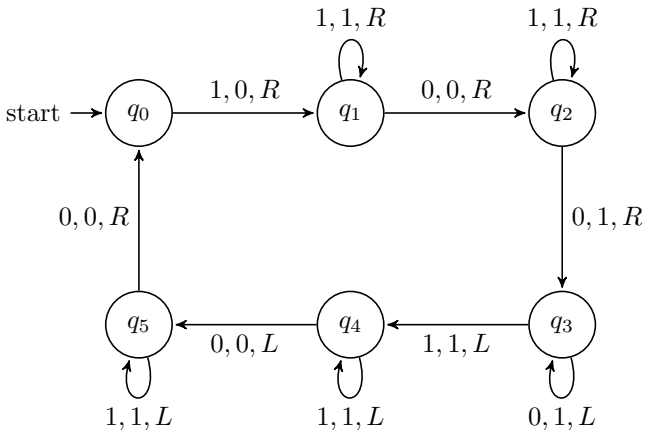


Figure 31.2: A doubler machine

tur:mac:rep:
fig:doubler

Example 31.3. The machine table for the even machine is:

	0	1	▷
q_0		$1, q_1, R$	
q_1	$0, q_1, R$	$1, q_0, R$	

As we can see, the machine halts when scanning a 0 in state q_0.

So far we have only considered machines that read and accept input. However, Turing machines have the capacity to both read and write. An example of such a machine (although there are many, many examples) is a *doubler*. A doubler, when started with a block of n 1's on the tape, outputs a block of $2n$ 1's. explanation

tur:mac:rep:
ex:doubler

Example 31.4. Before building a doubler machine, it is important to come up with a *strategy* for solving the problem. Since the machine (as we have formulated it) cannot remember how many 1's it has read, we need to come up with a way to keep track of all the 1's on the tape. One such way is to separate the output from the input with a 0. The machine can then erase the first 1 from the input, traverse over the rest of the input, leave a 0, and write two new 1's. The machine will then go back and find the second 1 in the input, and double that one as well. For each one 1 of input, it will write two 1's of output. By erasing the input as the machine goes, we can guarantee that no 1 is missed or doubled twice. When the entire input is erased, there will be $2n$ 1's left on the tape. The state diagram of the resulting Turing machine is depicted in Figure 31.2.

Problem 31.1. Choose an arbitrary input and trace through the configurations of the doubler machine in Example 31.4.

Problem 31.2. Design a Turing-machine with alphabet $\{\triangleright, 0, A, B\}$ that accepts, i.e., halts on, any string of A's and B's where the number of A's is the same as the number of B's *and* all the A's precede all the B's, and rejects, i.e., does not halt on, any string where the number of A's is not equal to the number of B's or the A's do not precede all the B's. (E.g., the machine should accept $AABB$, and $AAABBB$, but reject both AAB and $AABBAABB$.)

Problem 31.3. Design a Turing-machine with alphabet $\{\triangleright, 0, A, B\}$ that takes as input any string α of A's and B's and duplicates them to produce an output of the form $\alpha\alpha$. (E.g. input $ABBA$ should result in output $ABBAABBA$).

Problem 31.4. *Alphabetical?:* Design a Turing-machine with alphabet $\{\triangleright, 0, A, B\}$ that when given as input a finite sequence of A's and B's checks to see if all the A's appear to the left of all the B's or not. The machine should leave the input string on the tape, and either halt if the string is "alphabetical", or loop forever if the string is not.

Problem 31.5. *Alphabetizer:* Design a Turing-machine with alphabet $\{\triangleright, 0, A, B\}$ that takes as input a finite sequence of A's and B's rearranges them so that all the A's are to the left of all the B's. (e.g., the sequence $BABAA$ should become the sequence $AAABB$, and the sequence $ABBABB$ should become the sequence $AABBBB$).

---

content/turing-machines/machines-computations/turing-machines.tex

31.3 Turing Machines

explanation

The formal definition of what constitutes a Turing machine looks abstract, but is actually simple: it merely packs into one mathematical structure all the information needed to specify the workings of a Turing machine. This includes (1) which states the machine can be in, (2) which symbols are allowed to be on the tape, (3) which state the machine should start in, and (4) what the instruction set of the machine is.

tur:mac:tur:sec

Definition 31.5 (Turing machine). A *Turing machine* M is a tuple $\langle Q, \Sigma, q_0, \delta\rangle$ consisting of

1. a finite set of *states* Q,
2. a finite *alphabet* Σ which includes $\triangleright$ and 0,
3. an *initial state* $q_0 \in Q$,
4. a finite *instruction set* $\delta\colon Q \times \Sigma \nrightarrow Q \times \Sigma \times \{L, R, N\}$.

The partial function δ is also called the *transition function* of M.

We assume that the tape is infinite in one direction only. For this reason it is useful to designate a special symbol $\triangleright$ as a marker for the left end of the tape. This makes it easier for Turing machine programs to tell when they're "in danger" of running off the tape. We could assume that this symbol is never overwritten, i.e., that $\delta(q, \triangleright) = \langle q', \triangleright, x\rangle$ if $\delta(q, \triangleright)$ is defined. Some textbooks do this, we do not. You can simply be careful when constructing your Turing machine that it never overwrites $\triangleright$. Moreover, there are cases where allowing such overwriting provides some convenient flexibility. explanation

Example 31.6. *Even Machine:* The even machine is formally the quadruple $\langle Q, \Sigma, q_0, \delta\rangle$ where

$$\begin{aligned} Q &= \{q_0, q_1\} \\ \Sigma &= \{\triangleright, 0, 1\}, \\ \delta(q_0, 1) &= \langle q_1, 1, R\rangle, \\ \delta(q_1, 1) &= \langle q_0, 1, R\rangle, \\ \delta(q_1, 0) &= \langle q_1, 0, R\rangle. \end{aligned}$$

---

content/turing-machines/machines-computations/configuration.tex

31.4 Configurations and Computations

cmp:tur:con:sec

Recall tracing through the configurations of the even machine earlier. The imaginary mechanism consisting of tape, read/write head, and Turing machine program is really just an intuitive way of visualizing what a Turing machine computation is. Formally, we can define the computation of a Turing machine on a given input as a sequence of *configurations*—and a configuration in turn is a sequence of symbols (corresponding to the contents of the tape at a given point in the computation), a number indicating the position of the read/write head, and a state. Using these, we can define what the Turing machine M computes on a given input. explanation

Definition 31.7 (Configuration). A *configuration* of Turing machine $M = \langle Q, \Sigma, q_0, \delta\rangle$ is a triple $\langle C, m, q\rangle$ where

1. $C \in \Sigma^*$ is a finite sequence of symbols from Σ,
2. $m \in \mathbb{N}$ is a number $< \text{len}(C)$, and
3. $q \in Q$

Intuitively, the sequence C is the content of the tape (symbols of all squares from the leftmost square to the last non-blank or previously visited square), m is the number of the square the read/write head is scanning (beginning with 0 being the number of the leftmost square), and q is the current state of the machine.

explanation

The potential input for a Turing machine is a sequence of symbols, usually a sequence that encodes a number in some form. The initial configuration of the Turing machine is that configuration in which we start the Turing machine to work on that input: the tape contains the tape end marker immediately followed by the input written on the squares to the right, the read/write head is scanning the leftmost square of the input (i.e., the square to the right of the left end marker), and the mechanism is in the designated start state q_0.

Definition 31.8 (Initial configuration). The *initial configuration* of M for input $I \in \Sigma^*$ is

$$\langle \triangleright \frown I, 1, q_0 \rangle.$$

explanation

The $\frown$ symbol is for *concatenation*—the input string begins immediately to the left end marker.

Definition 31.9. We say that a configuration $\langle C, m, q \rangle$ *yields the configuration* $\langle C', m', q' \rangle$ *in one step* (according to M), iff

1. the m-th symbol of C is σ,
2. the instruction set of M specifies $\delta(q, \sigma) = \langle q', \sigma', D \rangle$,
3. the m-th symbol of C' is σ', and
4. a) $D = L$ and $m' = m - 1$ if $m > 0$, otherwise $m' = 0$, or
   b) $D = R$ and $m' = m + 1$, or
   c) $D = N$ and $m' = m$,
5. if $m' = \text{len}(C)$, then $\text{len}(C') = \text{len}(C) + 1$ and the m'-th symbol of C' is 0. Otherwise $\text{len}(C') = \text{len}(C)$.
6. for all i such that $i < \text{len}(C)$ and $i \neq m$, $C'(i) = C(i)$,

Definition 31.10. A *run of M on input I* is a sequence C_i of configurations of M, where C_0 is the initial configuration of M for input I, and each C_i yields C_{i+1} in one step. cmp:tur:con: defn:run-output

We say that M *halts on input I after k steps* if $C_k = \langle C, m, q \rangle$, the mth symbol of C is σ, and $\delta(q, \sigma)$ is undefined. In that case, the *output* of M for input I is O, where O is a string of symbols not ending in 0 such that $C = \triangleright \frown O \frown 0^j$ for some $j \in \mathbb{N}$. (0^j is a sequence of j 0's.)

explanation

According to this definition, the output O of M always ends in a symbol other than 0, or, if at time k the entire tape is filled with 0 (except for the leftmost $\triangleright$), O is the empty string.

---

content/turing-machines/machines-computations/unary-numbers.tex

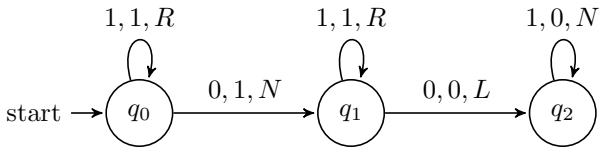


Figure 31.3: A machine computing $f(x, y) = x + y$

tur:mac:una:fig:adder

31.5 Unary Representation of Numbers

tur:mac:una:sec

explanation

Turing machines work on sequences of symbols written on their tape. Depending on the alphabet a Turing machine uses, these sequences of symbols can represent various inputs and outputs. Of particular interest, of course, are Turing machines which compute *arithmetical* functions, i.e., functions of natural numbers. A simple way to represent positive integers is by coding them as sequences of a single symbol 1. If $n \in \mathbb{N}$, let 1^n be the empty sequence if $n = 0$, and otherwise the sequence consisting of exactly n 1's.

Definition 31.11 (Computation). A Turing machine M *computes* the function $f\colon \mathbb{N}^k \to \mathbb{N}$ iff M halts on input

$$1^{n_1}01^{n_2}0\ldots01^{n_k}$$

with output $1^{f(n_1,\ldots,n_k)}$.

Problem 31.6. Give a definition for when a Turing machine M computes the function $f\colon \mathbb{N}^k \to \mathbb{N}^m$.

tur:mac:una:ex:adder

Example 31.12. *Addition:* Let's build a machine that computes the function $f(n, m) = n + m$. This requires a machine that starts with two blocks of 1's of length n and m on the tape, and halts with one block consisting of $n + m$ 1's. The two input blocks of 1's are separated by a 0, so one method would be to write a stroke on the square containing the 0, and erase the last 1.

Problem 31.7. Trace through the configurations of the machine from Example 31.12 for input $\langle 3, 2\rangle$. What happens if the machine computes $0 + 0$?

explanation

In Example 31.4, we gave an example of a Turing machine that takes as input a sequence of 1's and halts with a sequence of twice as many 1's on the tape—the doubler machine. However, because the output contains 0's to the left of the doubled block of 1's, it does not actually compute the function $f(x) = 2x$, as you might have assumed. We'll describe two ways of fixing that.

Example 31.13. The machine in Figure 31.4 computes the function $f(x) = 2x$. Instead of erasing the input and writing two 1's at the far right for every 1 in the input as the machine from Example 31.4 does, this machine adds a

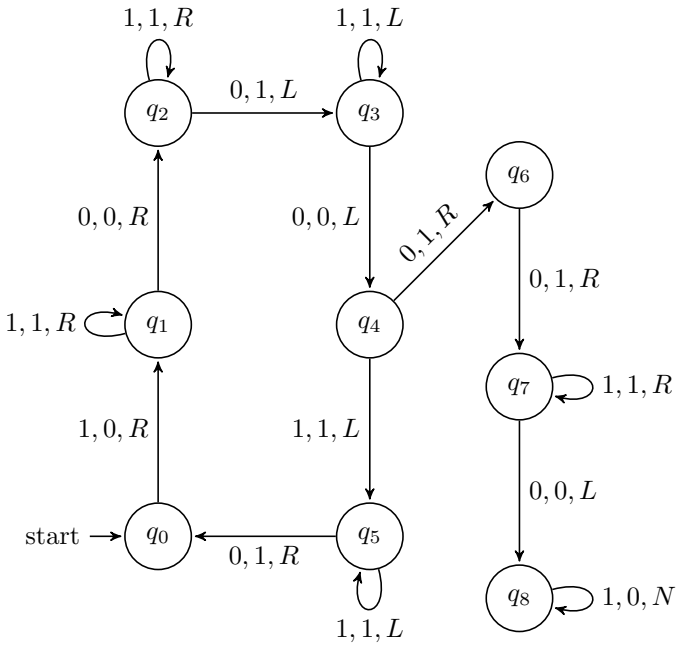


Figure 31.4: A machine computing $f(x) = 2x$

tur:mac:una: fig:doubler-disc

single 1 to the right for every 1 in the input. It has to keep track of where the input ends, so it leaves a 0 between the input and the added strokes, which it fills with a 1 at the very end. And we have to "remember" where we are in the input, so we temporarily replace a 1 in the input block by a 0.

Example 31.14. A second possibility for computing $f(x) = 2x$ is to keep the original doubler machine, but add states and instructions at the end which move the doubled block of strokes to the far left of the tape. The machine in Figure 31.5 does just this last part: started on a tape consisting of a block of 0's followed by a block of 1's (and the head positioned anywhere in the block of 0's), it erases the 1's one at a time and writes them at the beginning of the tape. In order to be able to tell when it is done, it first marks the end of the block of 1's with a ▷ symbol, which gets deleted at the end. We've started numbering the states at q_6, so they can be added to the doubler machine. All you'll need is an additional instruction $\delta(q_0, 0) = \langle q_6, 0, N\rangle$, i.e., an arrow from q_0 to q_6 labelled $0, 0, N$. (There is one subtle problem: the resulting machine does not work for input $x = 0$. We'll leave this as an exercise.)

tur:mac:una: ex:mover

Problem 31.8. In Example 31.14 we described a machine consisting of a combination of the doubler machine from Figure 31.4 and the mover machine from

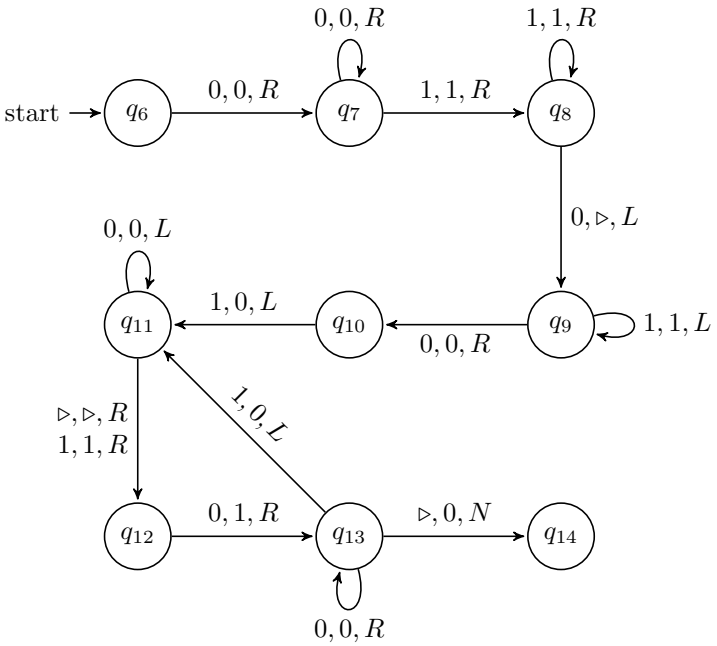


Figure 31.5: Moving a block of 1's to the left

tur:mac:una:
fig:mover

Figure 31.5. What happens if you start this combined machine on input $x = 0$, i.e., on an empty tape? How would you fix the machine so that in this case the machine halts with output $2x = 0$? (You should be able to do this by adding one state and one transition.)

Problem 31.9. *Subtraction:* Design a Turing machine that when given an input of two non-empty strings of strokes of length n and m, where $n > m$, computes the function $f(n, m) = n - m$.

Problem 31.10. *Equality:* Design a Turing machine to compute the following function:

$$\text{equality}(n, m) = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases}$$

where n and $m \in \mathbb{Z}^+$.

Problem 31.11. Design a Turing machine to compute the function $\min(x, y)$ where x and y are positive integers represented on the tape by strings of 1's separated by a 0. You may use additional symbols in the alphabet of the machine.

The function min selects the smallest value from its arguments, so $\min(3, 5) = 3$, $\min(20, 16) = 16$, and $\min(4, 4) = 4$, and so on.

Definition 31.15. A Turing machine M computes the partial function $f : \mathbb{N}^k \nrightarrow \mathbb{N}$ iff,

1. M halts on input $1^{n_1} \frown 0 \frown \ldots \frown 0 \frown 1^{n_k}$ with output 1^m if $f(n_1, \ldots, n_k) = m$.

2. M does not halt at all, or with an output that is not a single block of 1's if $f(n_1, \ldots, n_k)$ is undefined.

---

content/turing-machines/machines-computations/halting-states.tex

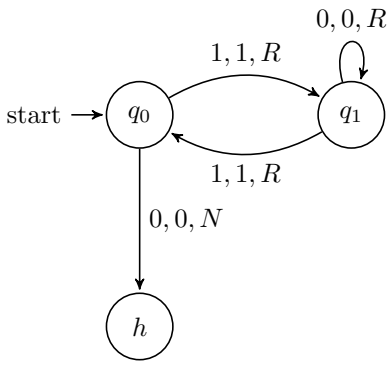
31.6 Halting States

explanation

Although we have defined our machines to halt only when there is no instruction to carry out, common representations of Turing machines have a dedicated *halting state* h, such that $h \in Q$. tur:mac:halt-sec

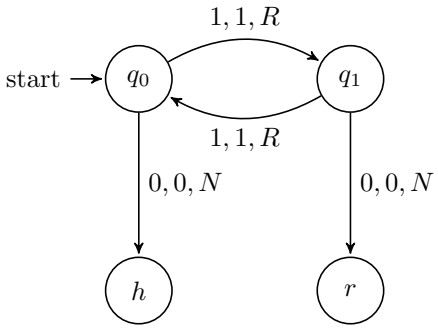
The idea behind a halting state is simple: when the machine has finished operation (it is ready to accept input, or has finished writing the output), it goes into a state h where it halts. Some machines have two halting states, one that accepts input and one that rejects input.

Example 31.16. *Halting States.* To elucidate this concept, let us begin with an alteration of the even machine. Instead of having the machine halt in state q_0 if the input is even, we can add an instruction to send the machine into a halting state.



Let us further expand the example. When the machine determines that the input is odd, it never halts. We can alter the machine to include a *reject*

state by replacing the looping instruction with an instruction to go to a reject state r.



Adding a dedicated halting state can be advantageous in cases like this, where it makes explicit when the machine accepts/rejects certain inputs. However, it is important to note that no computing power is gained by adding a dedicated halting state. Similarly, a less formal notion of halting has its own advantages. The definition of halting used so far in this chapter makes the proof of the *Halting Problem* intuitive and easy to demonstrate. For this reason, we continue with our original definition. explanation

---

content/turing-machines/machines-computations/disciplined-machines.tex

31.7 Disciplined Machines

tur:mac:dis:sec In section section 31.6, we considered Turing machines that have a single, designated halting state h—such machines are guaranteed to halt, if they halt at all, in state h. In this way, machines with a single halting state are more "disciplined" than we allow Turing machines in general to be. There are other restrictions we might impose on the behavior of Turing machines. For instance, we also have not prohibited Turing machines from ever erasing the tape-end marker on square 0, or to attempt to move left from square 0. (Our definition states that the head simply stays on square 0 in this case; other definitions have the machine halt.) It is likewise sometimes desirable to be able to assume that a Turing machine, if it halts at all, halts on square 1. explanation

tur:mac:dis:defn:disciplined **Definition 31.17.** A Turing machine M is *disciplined* iff

1. it has a designated single halting state h,
2. it halts, if it halts at all, while scanning square 1,
3. it never erases the ▷ symbol on square 0, and
4. it never attempts to move left from square 0.

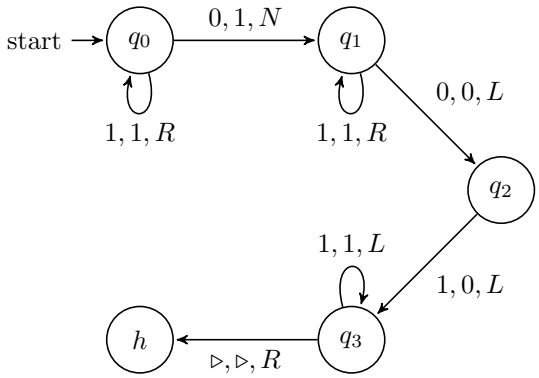


Figure 31.6: A disciplined addition machine

tur:mac:dis: fig:adder-disc

explanation

We have already discussed that any Turing machine can be changed into one with the same behavior but with a designated halting state. This is done simply by adding a new state h, and adding an instruction $\delta(q, \sigma) = \langle h, \sigma, N\rangle$ for any pair $\langle q, \sigma\rangle$ where the original δ is undefined. It is true, although tedious to prove, that any Turing machine M can be turned into a disciplined Turing machine M' which halts on the same inputs and produces the same output. For instance, if the Turing machine halts and is not on square 1, we can add some instructions to make the head move left until it finds the tape-end marker, then move one square to the right, then halt. We'll leave you to think about how the other conditions can be dealt with.

Example 31.18. In Figure 31.6, we turn the addition machine from Example 31.12 into a disciplined machine.

Proposition 31.19. *For every Turing machine M, there is a disciplined Turing machine M' which halts with output O if M halts with output O, and does not halt if M does not halt. In particular, any function $f\colon \mathbb{N}^n \to \mathbb{N}$ computable by a Turing machine is also computable by a disciplined Turing machine.*

tur:mac:dis: prop:disciplined

Problem 31.12. Give a disciplined machine that computes $f(x) = x + 1$.

Problem 31.13. Find a disciplined machine which, when started on input 1^n produces output $1^n \frown 0 \frown 1^n$.

content/turing-machines/machines-computations/combining-machines.tex

31.8 Combining Turing Machines

tur:mac:cmb:sec

explanation

The examples of Turing machines we have seen so far have been fairly simple in nature. But in fact, any problem that can be solved with any modern programming language can also be solved with Turing machines. To build more complex Turing machines, it is important to convince ourselves that we can combine them, so we can build machines to solve more complex problems by breaking the procedure into simpler parts. If we can find a natural way to break a complex problem down into constituent parts, we can tackle the problem in several stages, creating several simple Turing machines and combining them into one machine that can solve the problem. This point is especially important when tackling the Halting Problem in the next section.

How do we combine Turing machines $M = \langle Q, \Sigma, q_0, \delta\rangle$ and $M' = \langle Q', \Sigma', q_0', \delta'\rangle$? We now use the configuration of the tape after M has halted as the input configuration of a run of machine M'. To get a single Turing machine $M \frown M'$ that does this, do the following:

1. Renumber (or relabel) all the states Q' of M' so that M and M' have no states in common ($Q \cap Q' = \emptyset$).
2. The states of $M \frown M'$ are $Q \cup Q'$.
3. The tape alphabet is $\Sigma \cup \Sigma'$.
4. The start state is q_0.
5. The transition function is the function δ'' given by:

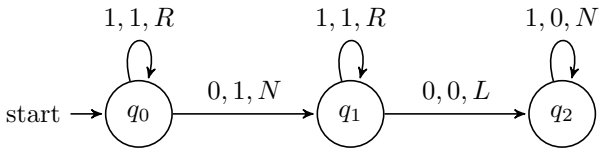
$$\delta''(q,\sigma) = \begin{cases} \delta(q,\sigma) & \text{if } q \in Q \\ \delta'(q,\sigma) & \text{if } q \in Q' \\ \langle q_0', \sigma, N\rangle & \text{if } q \in Q \text{ and } \delta(q,\sigma) \text{ is undefined} \end{cases}$$

The resulting machine uses the instructions of M when it is in a state $q \in Q$, the instructions of M' when it is in a state $q \in Q'$. When it is in a state $q \in Q$ and is scanning a symbol σ for which M has no transition (i.e., M would have halted), it enters the start state of M' (and leaves the tape contents and head position as it is).

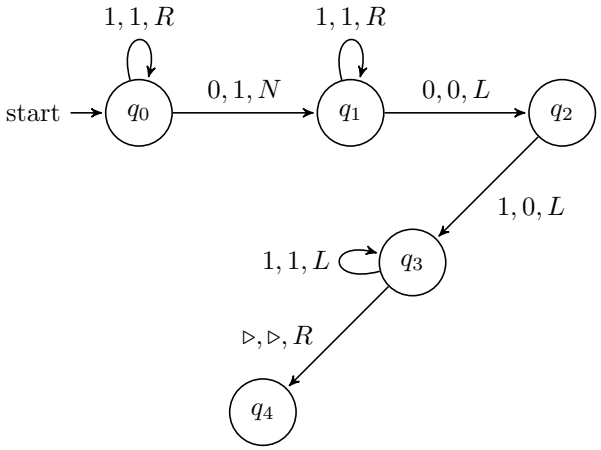
Note that unless the machine M is disciplined, we don't know where the tape head is when M halts, so the halting configuration of M need not have the head scanning square 1. When combining machines, it's important to keep this in mind.

Example 31.20. *Combining Machines:* We'll design a machine which, when started on input consisting of two blocks of 1's of length n and m, halts with a single block of $2(m+n)$ 1's on the tape. In order to build this machine, we can combine two machines we are already familiar with: the addition machine, and

the doubler. We begin by drawing a state diagram for the addition machine.



Instead of halting in state q_2, we want to continue operation in order to double the output. Recall that the doubler machine erases the first stroke in the input and writes two strokes in a separate output. Let's add an instruction to make sure the tape head is reading the first stroke of the output of the addition machine.



It is now easy to double the input—all we have to do is connect the doubler machine onto state q_4. This requires renaming the states of the doubler machine so that they start at q_4 instead of q_0—this way we don't end up with two starting states. The final diagram should look as in Figure 31.7.

Proposition 31.21. *If M and M' are disciplined and compute the functions $f\colon \mathbb{N}^k \to \mathbb{N}$ and $f'\colon \mathbb{N} \to \mathbb{N}$, respectively, then $M \frown M'$ is disciplined and computes $f' \circ f$.*

Proof. Since M is disciplined, when it halts with output $f(n_1, \ldots, n_k) = m$, the head is scanning square 1. If we now enter the start state of M', the machine will halt with output $f'(m)$, again scanning square 1. The other conditions of Definition 31.17 are also satisfied. □

Problem 31.14. Give a disciplined Turing machine computing $f(x) = x + 2$ by taking the machine M from Problem 31.12 and construct $M \frown M$.

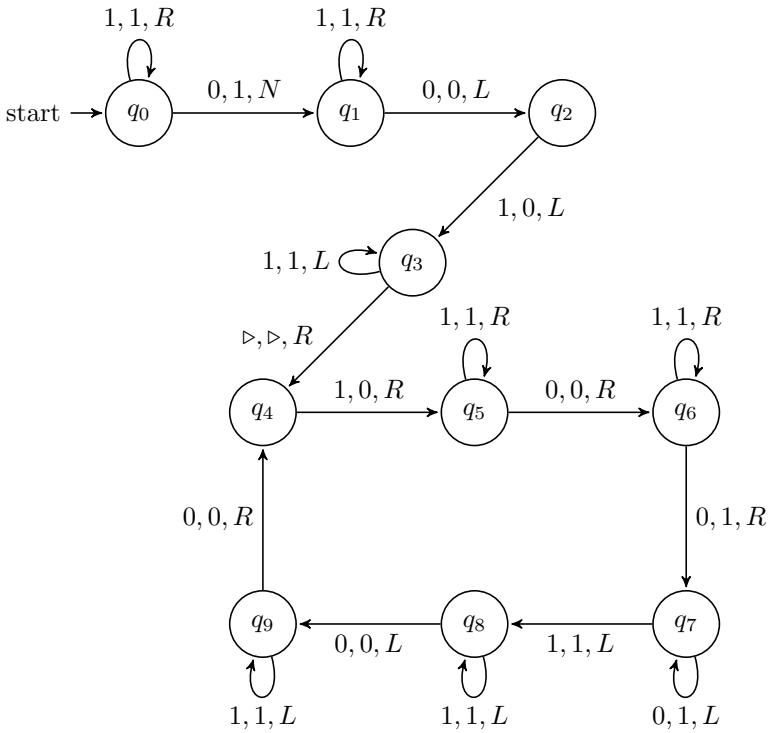


Figure 31.7: Combining adder and doubler machines

tur:mac:cmb:fig:combined

31.9 Variants of Turing Machines

tur:mac:var:sec

There are in fact many possible ways to define Turing machines, of which ours is only one. In some ways, our definition is more liberal than others. We allow arbitrary finite alphabets, a more restricted definition might allow only two tape symbols, 1 and 0. We allow the machine to write a symbol to the tape and move at the same time, other definitions allow either writing or moving. We allow the possibility of writing without moving the tape head, other definitions leave out the N "instruction." In other ways, our definition is more restrictive. We assumed that the tape is infinite in one direction only, other definitions allow the tape to be infinite both to the left and the right. In fact, one can even allow any number of separate tapes, or even an infinite grid of squares. We represent the instruction set of the Turing machine by a transition function; other definitions use a transition relation where the machine has more than one possible instruction in any given situation.

This last relaxation of the definition is particularly interesting. In our definition, when the machine is in state q reading symbol σ, $\delta(q, \sigma)$ determines

what the new symbol, state, and tape head position is. But if we allow the instruction set to be a relation between current state-symbol pairs $\langle q, \sigma \rangle$ and new state-symbol-direction triples $\langle q', \sigma', D \rangle$, the action of the Turing machine may not be uniquely determined—the instruction relation may contain both $\langle q, \sigma, q', \sigma', D \rangle$ and $\langle q, \sigma, q'', \sigma'', D' \rangle$. In this case we have a *non-deterministic* Turing machine. These play an important role in computational complexity theory.

There are also different conventions for when a Turing machine halts: we say it halts when the transition function is undefined, other definitions require the machine to be in a special designated halting state. We have explained in section 31.6 why requiring a designated halting state is not a restriction which impacts what Turing machines can compute. Since the tapes of our Turing machines are infinite in one direction only, there are cases where a Turing machine can't properly carry out an instruction: if it reads the leftmost square and is supposed to move left. According to our definition, it just stays put instead of "falling off", but we could have defined it so that it halts when that happens. This definition is also equivalent: we could simulate the behavior of a Turing machine that halts when it attempts to move left from square 0 by deleting every transition $\delta(q, \triangleright) = \langle q', \sigma, L \rangle$—then instead of attempting to move left on $\triangleright$ the machine halts.[1]

There are also different ways of representing numbers (and hence the input-output function computed by a Turing machine): we use unary representation, but you can also use binary representation. This requires two symbols in addition to 0 and $\triangleright$.

Now here is an interesting fact: none of these variations matters as to which functions are Turing computable. *If a function is Turing computable according to one definition, it is Turing computable according to all of them.*

We won't go into the details of verifying this. Here's just one example: we gain no additional computing power by allowing a tape that is infinite in both directions, or multiple tapes. The reason is, roughly, that a Turing machine with a single one-way infinite tape can simulate multiple or two-way infinite tapes. E.g., using additional states and instructions, we can "translate" a program for a machine with multiple tapes or two-way infinite tape into one with a single one-way infinite tape. The translated machine can use the even squares for the squares of tape 1 (or the "positive" squares of a two-way infinite tape) and the odd squares for the squares of tape 2 (or the "negative" squares).

content/turing-machines/machines-computations/church-turing-thesis.tex

31.10 The Church–Turing Thesis

tur:mac:ctt:sec

[1] This doesn't *quite* work, since nothing prevents us from writing and reading $\triangleright$ on squares other than square 0 (see Example 31.14). We can get around that by adding a second $\triangleright'$ symbol to use instead for such a purpose.

Turing machines are supposed to be a precise replacement for the concept of an effective procedure. Turing thought that anyone who grasped both the concept of an effective procedure and the concept of a Turing machine would have the intuition that anything that could be done via an effective procedure could be done by Turing machine. This claim is given support by the fact that all the other proposed precise replacements for the concept of an effective procedure turn out to be extensionally equivalent to the concept of a Turing machine —that is, they can compute exactly the same set of functions. This claim is called the *Church–Turing thesis*.

Definition 31.22 (Church–Turing thesis). The *Church–Turing Thesis* states that anything computable via an effective procedure is Turing computable.

The Church–Turing thesis is appealed to in two ways. The first kind of use of the Church–Turing thesis is an excuse for laziness. Suppose we have a description of an effective procedure to compute something, say, in "pseudo-code." Then we can invoke the Church–Turing thesis to justify the claim that the same function is computed by some Turing machine, even if we have not in fact constructed it.

The other use of the Church–Turing thesis is more philosophically interesting. It can be shown that there are functions which cannot be computed by Turing machines. From this, using the Church–Turing thesis, one can conclude that it cannot be effectively computed, using any procedure whatsoever. For if there were such a procedure, by the Church–Turing thesis, it would follow that there would be a Turing machine for it. So if we can prove that there is no Turing machine that computes it, there also can't be an effective procedure. In particular, the Church–Turing thesis is invoked to claim that the so-called halting problem not only cannot be solved by Turing machines, it cannot be effectively solved at all.

Chapter 32

Undecidability

---

content/turing-machines/undecidability/introduction.tex

32.1 Introduction

tur:und:int:sec

It might seem obvious that not every function, even every arithmetical function, can be computable. There are just too many, whose behavior is too complicated. Functions defined from the decay of radioactive particles, for instance, or other chaotic or random behavior. Suppose we start counting 1-second intervals from a given time, and define the function $f(n)$ as the number of particles in the universe that decay in the n-th 1-second interval after that initial moment. This seems like a candidate for a function we cannot ever hope to compute.

But it is one thing to not be able to imagine how one would compute such functions, and quite another to actually prove that they are uncomputable. In fact, even functions that seem hopelessly complicated may, in an abstract sense, be computable. For instance, suppose the universe is finite in time—some day, in the very distant future the universe will contract into a single point, as some cosmological theories predict. Then there is only a finite (but incredibly large) number of seconds from that initial moment for which $f(n)$ is defined. And any function which is defined for only finitely many inputs is computable: we could list the outputs in one big table, or code it in one very big Turing machine state transition diagram.

We are often interested in special cases of functions whose values give the answers to yes/no questions. For instance, the question "is n a prime number?" is associated with the function

$$\text{isprime}(n) = \begin{cases} 1 & \text{if } n \text{ is prime} \\ 0 & \text{otherwise.} \end{cases}$$

We say that a yes/no question can be *effectively decided*, if the associated 1/0-valued function is effectively computable.

To prove mathematically that there are functions which cannot be effectively computed, or problems that cannot effectively decided, it is essential to fix a specific model of computation, and show that there are functions it cannot compute or problems it cannot decide. We can show, for instance, that not every function can be computed by Turing machines, and not every problem can be decided by Turing machines. We can then appeal to the Church–Turing thesis to conclude that not only are Turing machines not powerful enough to compute every function, but no effective procedure can.

The key to proving such negative results is the fact that we can assign numbers to Turing machines themselves. The easiest way to do this is to enumerate them, perhaps by fixing a specific way to write down Turing machines and their programs, and then listing them in a systematic fashion. Once we see that this can be done, then the existence of Turing-uncomputable functions follows by simple cardinality considerations: the set of functions from $\mathbb{N}$ to $\mathbb{N}$ (in fact, even just from $\mathbb{N}$ to $\{0, 1\}$) are non-enumerable, but since we can enumerate all the Turing machines, the set of Turing-computable functions is only denumerable.

We can also define *specific* functions and problems which we can prove to be uncomputable and undecidable, respectively. One such problem is the so-called *Halting Problem.* Turing machines can be finitely described by listing their instructions. Such a description of a Turing machine, i.e., a Turing machine program, can of course be used as input to another Turing machine. So we can consider Turing machines that decide questions about other Turing machines. One particularly interesting question is this: "Does the given Turing machine eventually halt when started on input n?" It would be nice if there were a Turing machine that could decide this question: think of it as a quality-control Turing machine which ensures that Turing machines don't get caught in infinite loops and such. The interesting fact, which Turing proved, is that there cannot be such a Turing machine. There cannot be a single Turing machine which, when started on input consisting of a description of a Turing machine M and some number n, will always halt with either output 1 or 0 according to whether M machine would have halted when started on input n or not.

Once we have examples of specific undecidable problems we can use them to show that other problems are undecidable, too. For instance, one celebrated undecidable problem is the question, "Is the first-order formula φ valid?". There is no Turing machine which, given as input a first-order formula φ, is guaranteed to halt with output 1 or 0 according to whether φ is valid or not. Historically, the question of finding a procedure to effectively solve this problem was called simply "the" decision problem; and so we say that the decision problem is unsolvable. Turing and Church proved this result independently at around the same time, so it is also called the Church–Turing Theorem.

---

content/turing-machines/undecidability/enumerating-tms.tex

32.2 Enumerating Turing Machines

tur:und:enu:sec

We can show that the set of all Turing machines is enumerable. This follows from the fact that each Turing machine can be finitely described. The set of states and the tape vocabulary are finite sets. The transition function is a partial function from $Q \times \Sigma$ to $Q \times \Sigma \times \{L, R, N\}$, and so likewise can be specified by listing its values for the finitely many argument pairs for which it is defined.

explanation

This is true as far as it goes, but there is a subtle difference. The definition of Turing machines made no restriction on what elements the set of states and tape alphabet can have. So, e.g., for every real number, there technically is a Turing machine that uses that number as a state. However, the *behavior* of the Turing machine is independent of which objects serve as states and vocabulary. Consider the two Turing machines in Figure 32.1. These two diagrams correspond to two machines, M with the tape alphabet $\Sigma = \{\triangleright, 0, 1\}$ and set of states $\{q_0, q_1\}$, and M' with alphabet $\Sigma' = \{\triangleright, 0, A\}$ and states $\{s, h\}$. But their instructions are otherwise the same: M will halt on a sequence of n 1's iff n is even, and M' will halt on a sequence of n A's iff n is even. All

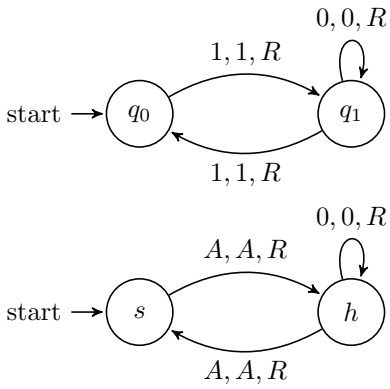


Figure 32.1: Variants of the *Even* machine

tur:und:enu:
fig:variants

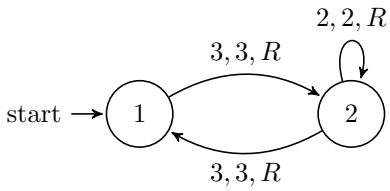


Figure 32.2: A standard *Even* machine

tur:und:enu:
fig:standard-even

we've done is rename 1 to A, q_0 to s, and q_1 to h. This example generalizes: we can think of Turing machines as the same as long as one results from the other by such a renaming of symbols and states. In fact, we can simply think of the symbols and states of a Turing machine as positive integers: instead of σ_0 think 1, instead of σ_1 think 2, etc.; $\triangleright$ is 1, 0 is 2, etc. In this way, the *Even* machine becomes the machine depicted in Figure 32.2. We might call a Turing machine with states and symbols that are positive integers a *standard* machine, and only consider standard machines from now on.[1]

We wanted to show that the set of Turing machines is enumerable, and with the above considerations in mind, it is enough to show that the set of standard Turing machines is enumerable. Suppose we are given a standard Turing machine $M = \langle Q, \Sigma, q_0, \delta \rangle$. How could we describe it using a finite string of positive integers? We'll first list the number of states, the states themselves, the number of symbols, the symbols themselves, and the starting state. (Remember, all of these are positive integers, since M is a standard machine.) What about δ? The set of possible arguments, i.e., pairs $\langle q, \sigma \rangle$, is

[1] The terminology "standard machine" is not standard.

finite, since Q and Σ are finite. So the information in δ is simply the finite list of all 5-tuples $\langle q, \sigma, q', \sigma', d\rangle$ where $\delta(q,\sigma) = \langle q', \sigma', D\rangle$, and d is a number that codes the direction D (say, 1 for L, 2 for R, and 3 for N).

In this way, every standard Turing machine can be described by a finite list of positive integers, i.e., as a sequence $s_M \in (\mathbb{Z}^+)^*$. For instance, the standard *Even* machine is coded by the sequence

$$2, \underbrace{1,2}_{Q}, 3, \overbrace{1,2,3}^{\Sigma}, 1, \underbrace{1,3,2,3,2}_{\delta(1,3)=\langle 2,3,R\rangle}, \overbrace{2,2,2,2,2}^{\delta(2,2)=\langle 2,2,R\rangle}, \underbrace{2,3,1,3,2}_{\delta(2,3)=\langle 1,3,R\rangle}.$$

Theorem 32.1. *There are functions from $\mathbb{N}$ to $\mathbb{N}$ which are not Turing computable.*

Proof. We know that the set of finite sequences of positive integers $(\mathbb{Z}^+)^*$ is enumerable (Problem 4.7). This gives us that the set of descriptions of standard Turing machines, as a subset of $(\mathbb{Z}^+)^*$, is itself enumerable. Every Turing computable function $\mathbb{N}$ to $\mathbb{N}$ is computed by some (in fact, many) Turing machines. By renaming its states and symbols to positive integers (in particular, $\triangleright$ as 1, 0 as 2, and 1 as 3) we can see that every Turing computable function is computed by a standard Turing machine. This means that the set of all Turing computable functions from $\mathbb{N}$ to $\mathbb{N}$ is also enumerable.

On the other hand, the set of all functions from $\mathbb{N}$ to $\mathbb{N}$ is not enumerable (Problem 4.35). If all functions were computable by some Turing machine, we could enumerate the set of all functions by listing all the descriptions of Turing machines that compute them. So there are some functions that are not Turing computable. □

Problem 32.1. Can you think of a way to describe Turing machines that does not require that the states and alphabet symbols are explicitly listed? You may define your own notion of "standard" machine, but say something about why every Turing machine can be computed by a "standard" machine in your new sense.

---

content/turing-machines/undecidability/universal-tm.tex

32.3 Universal Turing Machines

tur:und:uni:sec

In section 32.2 we discussed how every Turing machine can be described by a finite sequence of integers. This sequence encodes the states, alphabet, start state, and instructions of the Turing machine. We also pointed out that the set of all of these descriptions is enumerable. Since the set of such descriptions is denumerable, this means that there is a surjective function from $\mathbb{N}$ to these descriptions. Such a surjective function can be obtained, for instance, using Cantor's zig-zag method. It gives us a way of enumerating all (descriptions) of

Turing machines. If we fix one such enumeration, it now makes sense to talk of the 1st, 2nd, ..., eth Turing machine. These numbers are called *indices*.

Definition 32.2. If M is the eth Turing machine (in our fixed enumeration), we say that e is an *index* of M. We write M_e for the eth Turing machine.

A machine may have more than one index, e.g., two descriptions of M may differ in the order in which we list its instructions, and these different descriptions will have different indices.

Importantly, it is possible to give the enumeration of Turing machine descriptions in such a way that we can effectively compute the description of M from its index, and to effectively compute an index of a machine M from its description. By the Church–Turing thesis, it is then possible to find a Turing machine which recovers the description of the Turing machine with index e and writes the corresponding description on its tape as output. The description would be a sequence of blocks of 1's (representing the positive integers in the sequence describing M_e).

Given this, it now becomes natural to ask: what functions of Turing machine indices are themselves computable by Turing machines? What properties of Turing machine indices can be decided by Turing machines? An example: the function that maps an index e to the number of states the Turing machine with index e has, is computable by a Turing machine. Here's what such a Turing machine would do: started on a tape containing a single block of e 1's, it would first decode e into its description. The description is now represented by a sequence of blocks of 1's on the tape. Since the first element in this sequence is the number of states. So all that has to be done now is to erase everything but the first block of 1's and then halt.

A remarkable result is the following:

tur:und:uni: thm:universal-tm

Theorem 32.3. *There is a* universal Turing machine U *which, when started on input* $\langle e, n\rangle$

1. *halts iff* M_e *halts on input* n*, and*
2. *if* M_e *halts with output* m*, so does* U*.*

U thus computes the function $f : \mathbb{N} \times \mathbb{N} \nrightarrow \mathbb{N}$ *given by* $f(e, n) = m$ *if* M_e *started on input* n *halts with output* m*, and undefined otherwise.*

Proof. To actually produce U is basically impossible, since it is an extremely complicated machine. But we can describe in outline how it works, and then invoke the Church–Turing thesis. When it starts, U's tape contains a block of e 1's followed by a block of n 1's. It first "decodes" the index e to the right of the input n. This produces a list of numbers (i.e., blocks of 1's separated by 0's) that describes the instructions of machine M_e. U then writes the number of the start state of M_e and the number 1 on the tape to the right of the description of M_e. (Again, these are represented in unary, as blocks of 1's.) Next, it copies the input (block of n 1's) to the right—but it replaces each 1 by a block of three

1's (remember, the number of the 1 symbol is 3, 1 being the number of ▷ and 2 being the number of 0). At the left end of this sequence of blocks (separated by 0 symbols on the tape of U), it writes a single 1, the code for ▷.

U now has on its tape: the index e, the number n, the code number of the start state (the "current state"), the number of the initial head position 1 (the "current head position"), and the initial contents of the "tape" (a sequence of blocks of 1's representing the code numbers of the symbols of M_e—the "symbols"—separated by 0's).

It now simulates what M_e would do if started on input n, by doing the following:

1. Find the number k of the "current head position" (at the beginning, that's 1),
2. Move to the kth block in the "tape" to see what the "symbol" there is,
3. Find the instruction matching the current "state" and "symbol,"
4. Move back to the kth block on the "tape" and replace the "symbol" there with the code number of the symbol M_e would write,
5. Move the head to where it records the current "state" and replace the number there with the number of the new state,
6. Move to the place where it records the "tape position" and erase a 1 or add a 1 (if the instruction says to move left or right, respectively).
7. Repeat.[2]

tur:und:uni:find-inst

If M_e started on input n never halts, then U also never halts, so its output is undefined.

If in step (3) it turns out that the description of M_e contains no instruction for the current "state"/"symbol" pair, then M_e would halt. If this happens, U erases the part of its tape to the left of the "tape." For each block of three 1's (representing a 1 on M_e's tape), it writes a 1 on the left end of its own tape, and successively erases the "tape." When this is done, U's tape contains a single block of 1's of length m.

If U encounters something other than a block of three 1's on the "tape," it immediately halts. Since U's tape in this case does not contain a single block of 1's, its output is not a natural number, i.e., $f(e, n)$ is undefined in this case.□

content/turing-machines/undecidability/halting-problem.tex

[2] We're glossing over some subtle difficulties here. E.g., U may need some extra space when it increases the counter where it keeps track of the "current head position"—in that case it will have to move the entire "tape" to the right.

32.4 The Halting Problem

explanation Assume we have fixed some enumeration of Turing machine descriptions. Each Turing machine thus receives an *index*: its place in the enumeration M_1, M_2, M_3, ... of Turing machine descriptions. tur:und:halt:sec

We know that there must be non-Turing-computable functions: the set of Turing machine descriptions—and hence the set of Turing machines—is enumerable, but the set of all functions from $\mathbb{N}$ to $\mathbb{N}$ is not. But we can find specific examples of non-computable functions as well. One such function is the halting function.

Definition 32.4 (Halting function). The *halting function* h is defined as

$$h(e,n) = \begin{cases} 0 & \text{if machine } M_e \text{ does not halt for input } n \\ 1 & \text{if machine } M_e \text{ halts for input } n \end{cases}$$

Definition 32.5 (Halting problem). The *Halting Problem* is the problem of determining (for any e, n) whether the Turing machine M_e halts for an input of n strokes.

explanation We show that h is not Turing-computable by showing that a related function, s, is not Turing-computable. This proof relies on the fact that anything that can be computed by a Turing machine can be computed by a disciplined Turing machine (section 31.7), and the fact that two Turing machines can be hooked together to create a single machine (section 31.8).

Definition 32.6. The function s is defined as

$$s(e) = \begin{cases} 0 & \text{if machine } M_e \text{ does not halt for input } e \\ 1 & \text{if machine } M_e \text{ halts for input } e \end{cases}$$

Lemma 32.7. *The function s is not Turing computable.*

Proof. We suppose, for contradiction, that the function s is Turing computable. Then there would be a Turing machine S that computes s. We may assume, without loss of generality, that when S halts, it does so while scanning the first square (i.e., that it is disciplined). This machine can be "hooked up" to another machine J, which halts if it is started on input 0 (i.e., if it reads 0 in the initial state while scanning the square to the right of the end-of-tape symbol), and otherwise wanders off to the right, never halting. $S \frown J$, the machine created by hooking S to J, is a Turing machine, so it is M_e for some e (i.e., it appears somewhere in the enumeration). Start M_e on an input of e 1s. There are two possibilities: either M_e halts or it does not halt.

1. Suppose M_e halts for an input of e 1s. Then $s(e) = 1$. So S, when started on e, halts with a single 1 as output on the tape. Then J starts with a 1 on the tape. In that case J does not halt. But M_e is the machine $S \frown J$,

so it should do exactly what S followed by J would do (i.e., in this case, wander off to the right and never halt). So M_e cannot halt for an input of e 1's.

2. Now suppose M_e does not halt for an input of e 1s. Then $s(e) = 0$, and S, when started on input e, halts with a blank tape. J, when started on a blank tape, immediately halts. Again, M_e does what S followed by J would do, so M_e must halt for an input of e 1's.

In each case we arrive at a contradiction with our assumption. This shows there cannot be a Turing machine S: s is not Turing computable. □

tur:und:hal: thm:halting-problem

Theorem 32.8 (Unsolvability of the Halting Problem). *The halting problem is unsolvable, i.e., the function h is not Turing computable.*

Proof. Suppose h were Turing computable, say, by a Turing machine H. We could use H to build a Turing machine that computes s: First, make a copy of the input (separated by a 0 symbol). Then move back to the beginning, and run H. We can clearly make a machine that does the former (see Problem 31.13), and if H existed, we would be able to "hook it up" to such a copier machine to get a new machine which would determine if M_e halts on input e, i.e., computes s. But we've already shown that no such machine can exist. Hence, h is also not Turing computable. □

Problem 32.2. The Three Halting (3-Halt) problem is the problem of giving a decision procedure to determine whether or not an arbitrarily chosen Turing Machine halts for an input of three 1's on an otherwise blank tape. Prove that the 3-Halt problem is unsolvable.

Problem 32.3. Show that if the halting problem is solvable for Turing machine and input pairs M_e and n where $e \neq n$, then it is also solvable for the cases where $e = n$.

Problem 32.4. We proved that the halting problem is unsolvable if the input is a number e, which identifies a Turing machine M_e via an enumeration of all Turing machines. What if we allow the description of Turing machines from section 32.2 directly as input? Can there be a Turing machine which decides the halting problem but takes as input descriptions of Turing machines rather than indices? Explain why or why not.

Problem 32.5. Show that the *partial* function s' is defined as

$$s'(e) = \begin{cases} 1 & \text{if machine } M_e \text{ halts for input } e \\ \text{undefined} & \text{if machine } M_e \text{ does not halt for input } e \end{cases}$$

is Turing computable.

content/turing-machines/undecidability/decision-problem.tex

32.5 The Decision Problem

tur:und:dec:sec

We say that first-order logic is *decidable* iff there is an effective method for determining whether or not a given sentence is valid. As it turns out, there is no such method: the problem of deciding validity of first-order sentences is unsolvable.

In order to establish this important negative result, we prove that the decision problem cannot be solved by a Turing machine. That is, we show that there is no Turing machine which, whenever it is started on a tape that contains a first-order sentence, eventually halts and outputs either 1 or 0 depending on whether the sentence is valid or not. By the Church–Turing thesis, every function which is computable is Turing computable. So if this "validity function" were effectively computable at all, it would be Turing computable. If it isn't Turing computable, then, it also cannot be effectively computable.

Our strategy for proving that the decision problem is unsolvable is to reduce the halting problem to it. This means the following: We have proved that the function $h(e, w)$ that halts with output 1 if the Turing machine described by e halts on input w and outputs 0 otherwise, is not Turing computable. We will show that if there were a Turing machine that decides validity of first-order sentences, then there is also Turing machine that computes h. Since h cannot be computed by a Turing machine, there cannot be a Turing machine that decides validity either.

The first step in this strategy is to show that for every input w and a Turing machine M, we can effectively describe a sentence $\tau(M, w)$ representing the instruction set of M and the input w and a sentence $\alpha(M, w)$ expressing "M eventually halts" such that:

$$\vDash \tau(M, w) \to \alpha(M, w) \text{ iff } M \text{ halts for input } w.$$

The bulk of our proof will consist in describing these sentences $\tau(M, w)$ and $\alpha(M, w)$ and in verifying that $\tau(M, w) \to \alpha(M, w)$ is valid iff M halts on input w.

---

content/turing-machines/undecidability/representing-tms.tex

32.6 Representing Turing Machines

explanation

tur:und:rep:sec

In order to represent Turing machines and their behavior by a sentence of first-order logic, we have to define a suitable language. The language consists of two parts: predicate symbols for describing configurations of the machine, and expressions for numbering execution steps ("moments") and positions on the tape.

We introduce two kinds of predicate symbols, both of them 2-place: For each state q, a predicate symbol Q_q, and for each tape symbol σ, a predicate symbol S_σ. The former allow us to describe the state of M and the position of its tape head, the latter allow us to describe the contents of the tape.

In order to express the positions of the tape head and the number of steps executed, we need a way to express numbers. This is done using a constant symbol o, and a 1-place function $\prime$, the successor function. By convention it is written *after* its argument (and we leave out the parentheses).

For each number n there is a canonical term $\overline{n}$, the *numeral* for n, which represents it in $\mathcal{L}_M$. $\overline{0}$ is o, $\overline{1}$ is o', $\overline{2}$ is o'', and so on. More formally:

$$\overline{0} = \mathsf{o}$$
$$\overline{n+1} = \overline{n}'$$

The term $\overline{0}$, i.e., o names the leftmost position on the tape as well as the time before the first execution step (the initial configuration). The term $\overline{1}$, i.e., o' names the square to the right of the leftmost square, and the time after the first execution step, and so on.

We also introduce a predicate symbol $<$ to express both the ordering of tape positions (when it means "to the left of") and execution steps (then it means "before").

Once we have the language in place, we list the "axioms" of $\tau(M, w)$, i.e., the sentences which, taken together, describe the behavior of M when run on input w. There will be sentences which lay down conditions on o, $\prime$, and $<$, sentences that describes the input configuration, and sentences that describe what the configuration of M is after it executes a particular instruction.

tur:und:rep: defn:tm-descr

Definition 32.9. Given a Turing machine $M = \langle Q, \Sigma, q_0, \delta\rangle$, the language $\mathcal{L}_M$ consists of:

1. A two-place predicate symbol $Q_q(x, y)$ for every state $q \in Q$. Intuitively, $Q_q(\overline{m}, \overline{n})$ expresses "after n steps, M is in state q scanning the mth square."
2. A two-place predicate symbol $S_\sigma(x, y)$ for every symbol $\sigma \in \Sigma$. Intuitively, $S_\sigma(\overline{m}, \overline{n})$ expresses "after n steps, the mth square contains symbol σ."
3. A constant symbol o
4. A one-place function symbol $\prime$
5. A two-place predicate symbol $<$

The sentences describing the operation of the Turing machine M on input $w = \sigma_{i_1} \dots \sigma_{i_k}$ are the following:

1. Axioms describing numbers and $<$:
   a) A sentence that says that every number is less than its successor:

$$\forall x\, x < x'$$

b) A sentence that ensures that $<$ is transitive:

$$\forall x \forall y \forall z \left((x < y \wedge y < z) \to x < z\right)$$

2. Axioms describing the input configuration:

a) After 0 steps—before the machine starts—M is in the initial state q_0, scanning square 1:

$$Q_{q_0}(\overline{1}, \overline{0})$$

b) The first $k+1$ squares contain the symbols $\triangleright$, σ_{i_1}, …, σ_{i_k}:

$$S_{\triangleright}(\overline{0}, \overline{0}) \wedge S_{\sigma_{i_1}}(\overline{1}, \overline{0}) \wedge \cdots \wedge S_{\sigma_{i_k}}(\overline{k}, \overline{0})$$

c) Otherwise, the tape is empty:

$$\forall x \left(\overline{k} < x \to S_0(x, \overline{0})\right)$$

3. Axioms describing the transition from one configuration to the next:

For the following, let $\varphi(x, y)$ be the conjunction of all sentences of the form

$$\forall z \left(((z < x \vee x < z) \wedge S_\sigma(z, y)) \to S_\sigma(z, y')\right)$$

where $\sigma \in \Sigma$. We use $\varphi(\overline{m}, \overline{n})$ to express "other than at square m, the tape after $n+1$ steps is the same as after n steps."

a) For every instruction $\delta(q_i, \sigma) = \langle q_j, \sigma', R\rangle$, the sentence: tur:und:rep: rep-right

$$\begin{aligned}\forall x \forall y \big((&Q_{q_i}(x, y) \wedge S_\sigma(x, y)) \to \\ &(Q_{q_j}(x', y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y))\big)\end{aligned}$$

This says that if, after y steps, the machine is in state q_i scanning square x which contains symbol σ, then after $y+1$ steps it is scanning square $x+1$, is in state q_j, square x now contains σ', and every square other than x contains the same symbol as it did after y steps.

b) For every instruction $\delta(q_i, \sigma) = \langle q_j, \sigma', L\rangle$, the sentence: tur:und:rep: rep-left

$$\begin{aligned}\forall x \forall y \big((&Q_{q_i}(x', y) \wedge S_\sigma(x', y)) \to \\ &(Q_{q_j}(x, y') \wedge S_{\sigma'}(x', y') \wedge \varphi(x, y))\big) \wedge \\ \forall y \big((&Q_{q_i}(\overline{0}, y) \wedge S_\sigma(\overline{0}, y)) \to \\ &(Q_{q_j}(\overline{0}, y') \wedge S_{\sigma'}(\overline{0}, y') \wedge \varphi(\overline{0}, y))\big)\end{aligned}$$

Take a moment to think about how this works: now we don't start with "if scanning square x …" but: "if scanning square $x+1$ …" A move to the left means that in the next step the machine is scanning square x. But the square that is written on is $x+1$. We do it this way since we don't have subtraction or a predecessor function.

Note that numbers of the form $x+1$ are 1, 2, ..., i.e., this doesn't cover the case where the machine is scanning square 0 and is supposed to move left (which of course it can't—it just stays put). That special case is covered by the second conjunction: it says that if, after y steps, the machine is scanning square 0 in state q_i and square 0 contains symbol σ, then after $y+1$ steps it's still scanning square 0, is now in state q_j, the symbol on square 0 is σ', and the squares other than square 0 contain the same symbols they contained ofter y steps.

tur:und:rep: rep-stay

c) For every instruction $\delta(q_i, \sigma) = \langle q_j, \sigma', N\rangle$, the sentence:

$$\forall x\, \forall y\, ((Q_{q_i}(x,y) \wedge S_\sigma(x,y)) \rightarrow (Q_{q_j}(x,y') \wedge S_{\sigma'}(x,y') \wedge \varphi(x,y)))$$

Let $\tau(M,w)$ be the conjunction of all the above sentences for Turing machine M and input w.

In order to express that M eventually halts, we have to find a sentence that says "after some number of steps, the transition function will be undefined." Let X be the set of all pairs $\langle q, \sigma\rangle$ such that $\delta(q,\sigma)$ is undefined. Let $\alpha(M,w)$ then be the sentence

$$\exists x\, \exists y\, (\bigvee_{\langle q,\sigma\rangle \in X} (Q_q(x,y) \wedge S_\sigma(x,y)))$$

If we use a Turing machine with a designated halting state h, it is even easier: then the sentence $\alpha(M,w)$

$$\exists x\, \exists y\, Q_h(x,y)$$

expresses that the machine eventually halts.

tur:und:rep: prop:mlessk

Proposition 32.10. *If $m < k$, then $\tau(M,w) \vDash \overline{m} < \overline{k}$*

Proof. Exercise. □

Problem 32.6. Prove Proposition 32.10. (Hint: use induction on $k - m$).

---

content/turing-machines/undecidability/verification.tex

32.7 Verifying the Representation

tur:und:ver: sec

In order to verify that our representation works, we have to prove two things. First, we have to show that if M halts on input w, then $\tau(M,w) \rightarrow \alpha(M,w)$ is valid. Then, we have to show the converse, i.e., that if $\tau(M,w) \rightarrow \alpha(M,w)$ is valid, then M does in fact eventually halt when run on input w.

explanation

The strategy for proving these is very different. For the first result, we have to show that a sentence of first-order logic (namely, $\tau(M,w) \to \alpha(M,w)$) is valid. The easiest way to do this is to give a derivation. Our proof is supposed to work for all M and w, though, so there isn't really a single sentence for which we have to give a derivation, but infinitely many. So the best we can do is to prove by induction that, whatever M and w look like, and however many steps it takes M to halt on input w, there will be a derivation of $\tau(M,w) \to \alpha(M,w)$.

Naturally, our induction will proceed on the number of steps M takes before it reaches a halting configuration. In our inductive proof, we'll establish that for each step n of the run of M on input w, $\tau(M,w) \vDash \chi(M,w,n)$, where $\chi(M,w,n)$ correctly describes the configuration of M run on w after n steps. Now if M halts on input w after, say, n steps, $\chi(M,w,n)$ will describe a halting configuration. We'll also show that $\chi(M,w,n) \vDash \alpha(M,w)$, whenever $\chi(M,w,n)$ describes a halting configuration. So, if M halts on input w, then for some n, M will be in a halting configuration after n steps. Hence, $\tau(M,w) \vDash \chi(M,w,n)$ where $\chi(M,w,n)$ describes a halting configuration, and since in that case $\chi(M,w,n) \vDash \alpha(M,w)$, we get that $T(M,w) \vDash \alpha(M,w)$, i.e., that $\vDash \tau(M,w) \to \alpha(M,w)$.

The strategy for the converse is very different. Here we assume that $\vDash \tau(M,w) \to \alpha(M,w)$ and have to prove that M halts on input w. From the hypothesis we get that $\tau(M,w) \vDash \alpha(M,w)$, i.e., $\alpha(M,w)$ is true in every structure in which $\tau(M,w)$ is true. So we'll describe a structure $\mathfrak{M}$ in which $\tau(M,w)$ is true: its domain will be $\mathbb{N}$, and the interpretation of all the Q_q and S_σ will be given by the configurations of M during a run on input w. So, e.g., $\mathfrak{M} \vDash Q_q(\overline{m},\overline{n})$ iff T, when run on input w for n steps, is in state q and scanning square m. Now since $\tau(M,w) \vDash \alpha(M,w)$ by hypothesis, and since $\mathfrak{M} \vDash \tau(M,w)$ by construction, $\mathfrak{M} \vDash \alpha(M,w)$. But $\mathfrak{M} \vDash \alpha(M,w)$ iff there is some $n \in |\mathfrak{M}| = \mathbb{N}$ so that M, run on input w, is in a halting configuration after n steps.

Definition 32.11. Let $\chi(M,w,n)$ be the sentence

$$Q_q(\overline{m},\overline{n}) \wedge S_{\sigma_0}(\overline{0},\overline{n}) \wedge \cdots \wedge S_{\sigma_k}(\overline{k},\overline{n}) \wedge \forall x\,(\overline{k} < x \to S_0(x,\overline{n}))$$

where q is the state of M at time n, M is scanning square m at time n, square i contains symbol σ_i at time n for $0 \leq i \leq k$ and k is the right-most non-blank square of the tape at time 0, or the right-most square the tape head has visited after n steps, whichever is greater.

Lemma 32.12. *If M run on input w is in a halting configuration after n steps, then $\chi(M,w,n) \vDash \alpha(M,w)$.* *tur:und:ver: lem:halt-config-implies-halt*

Proof. Suppose that M halts for input w after n steps. There is some state q, square m, and symbol σ such that:

1. After n steps, M is in state q scanning square m on which σ appears.
2. The transition function $\delta(q,\sigma)$ is undefined.

$\chi(M, w, n)$ is the description of this configuration and will include the clauses $Q_q(\overline{m}, \overline{n})$ and $S_\sigma(\overline{m}, \overline{n})$. These clauses together imply $\alpha(M, w)$:

$$\exists x \, \exists y \, (\bigvee_{\langle q,\sigma\rangle \in X} (Q_q(x, y) \wedge S_\sigma(x, y)))$$

since $Q_{q'}(\overline{m}, \overline{n}) \wedge S_{\sigma'}(\overline{m}, \overline{n}) \vDash \bigvee_{\langle q,\sigma\rangle \in X} (Q_q(\overline{m}, \overline{n}) \wedge S_\sigma(\overline{m}, \overline{n}))$, as $\langle q', \sigma'\rangle \in X$.□

So if M halts for input w, then there is some n such that $\chi(M, w, n) \vDash \alpha(M, w)$. We will now show that for any time n, $\tau(M, w) \vDash \chi(M, w, n)$. explanation

tur:und:ver: lem:config **Lemma 32.13.** *For each n, if M has not halted after n steps, $\tau(M, w) \vDash \chi(M, w, n)$.*

Proof. Induction basis: If $n = 0$, then the conjuncts of $\chi(M, w, 0)$ are also conjuncts of $\tau(M, w)$, so entailed by it.

Inductive hypothesis: If M has not halted before the nth step, then $\tau(M, w) \vDash \chi(M, w, n)$. We have to show that (unless $\chi(M, w, n)$ describes a halting configuration), $\tau(M, w) \vDash \chi(M, w, n + 1)$.

Suppose $n > 0$ and after n steps, M started on w is in state q scanning square m. Since M does not halt after n steps, there must be an instruction of one of the following three forms in the program of M:

tur:und:ver: right 1. $\delta(q, \sigma) = \langle q', \sigma', R\rangle$

tur:und:ver: left 2. $\delta(q, \sigma) = \langle q', \sigma', L\rangle$

tur:und:ver: stay 3. $\delta(q, \sigma) = \langle q', \sigma', N\rangle$

We will consider each of these three cases in turn.

1. Suppose there is an instruction of the form (1). By Definition 32.9(3a), this means that

$$\forall x \, \forall y \, ((Q_q(x, y) \wedge S_\sigma(x, y)) \rightarrow (Q_{q'}(x', y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y)))$$

is a conjunct of $\tau(M, w)$. This entails the following sentence (universal instantiation, $\overline{m}$ for x and $\overline{n}$ for y):

$$(Q_q(\overline{m}, \overline{n}) \wedge S_\sigma(\overline{m}, \overline{n})) \rightarrow (Q_{q'}(\overline{m}', \overline{n}') \wedge S_{\sigma'}(\overline{m}, \overline{n}') \wedge \varphi(\overline{m}, \overline{n})).$$

By induction hypothesis, $\tau(M, w) \vDash \chi(M, w, n)$, i.e.,

$$Q_q(\overline{m}, \overline{n}) \wedge S_{\sigma_0}(\overline{0}, \overline{n}) \wedge \cdots \wedge S_{\sigma_k}(\overline{k}, \overline{n}) \wedge \forall x \, (\overline{k} < x \rightarrow S_0(x, \overline{n}))$$

Since after n steps, tape square m contains σ, the corresponding conjunct is $S_\sigma(\overline{m}, \overline{n})$, so this entails:

$$Q_q(\overline{m}, \overline{n}) \wedge S_\sigma(\overline{m}, \overline{n})$$

We now get

$$\begin{aligned} &Q_{q'}(\overline{m}', \overline{n}') \wedge S_{\sigma'}(\overline{m}, \overline{n}') \wedge \\ &\quad S_{\sigma_0}(\overline{0}, \overline{n}') \wedge \cdots \wedge S_{\sigma_k}(\overline{k}, \overline{n}') \wedge \\ &\quad \forall x\, (\overline{k} < x \to S_0(x, \overline{n}')) \end{aligned}$$

as follows: The first line comes directly from the consequent of the preceding conditional, by modus ponens. Each conjunct in the middle line—which excludes $S_{\sigma_m}(\overline{m}, \overline{n}')$—follows from the corresponding conjunct in $\chi(M, w, n)$ together with $\varphi(\overline{m}, \overline{n})$.

If $m < k$, $\tau(M, w) \vdash \overline{m} < \overline{k}$ (Proposition 32.10) and by transitivity of $<$, we have $\forall x\, (\overline{k} < x \to \overline{m} < x)$. If $m = k$, then $\forall x\, (\overline{k} < x \to \overline{m} < x)$ by logic alone. The last line then follows from the corresponding conjunct in $\chi(M, w, n)$, $\forall x\, (\overline{k} < x \to \overline{m} < x)$, and $\varphi(\overline{m}, \overline{n})$. If $m < k$, this already is $\chi(M, w, n+1)$.

Now suppose $m = k$. In that case, after $n+1$ steps, the tape head has also visited square $k+1$, which now is the right-most square visited. So $\chi(M, w, n+1)$ has a new conjunct, $S_0(\overline{k}', \overline{n}')$, and the last conjunct is $\forall x\, (\overline{k}' < x \to S_0(x, \overline{n}'))$. We have to verify that these two sentences are also implied.

We already have $\forall x\, (\overline{k} < x \to S_0(x, \overline{n}'))$. In particular, this gives us $\overline{k} < \overline{k}' \to S_0(\overline{k}', \overline{n}')$. From the axiom $\forall x\, x < x'$ we get $\overline{k} < \overline{k}'$. By modus ponens, $S_0(\overline{k}', \overline{n}')$ follows.

Also, since $\tau(M, w) \vdash \overline{k} < \overline{k}'$, the axiom for transitivity of $<$ gives us $\forall x\, (\overline{k}' < x \to S_0(x, \overline{n}'))$. (We leave the verification of this as an exercise.)

2. Suppose there is an instruction of the form (2). Then, by Definition 32.9(3b),

$$\begin{aligned} &\forall x\, \forall y\, ((Q_q(x', y) \wedge S_\sigma(x', y)) \to \\ &\quad (Q_{q'}(x, y') \wedge S_{\sigma'}(x', y') \wedge \varphi(x, y))) \wedge \\ &\forall y\, ((Q_{q_i}(\overline{0}, y) \wedge S_\sigma(\overline{0}, y)) \to \\ &\quad (Q_{q_j}(\overline{0}, y') \wedge S_{\sigma'}(\overline{0}, y') \wedge \varphi(\overline{0}, y))) \end{aligned}$$

is a conjunct of $\tau(M, w)$. If $m > 0$, then let $l = m - 1$ (i.e., $m = l + 1$). The first conjunct of the above sentence entails the following:

$$\begin{aligned} &(Q_q(\overline{l}', \overline{n}) \wedge S_\sigma(\overline{l}', \overline{n})) \to \\ &\quad (Q_{q'}(\overline{l}, \overline{n}') \wedge S_{\sigma'}(\overline{l}', \overline{n}') \wedge \varphi(\overline{l}, \overline{n})) \end{aligned}$$

Otherwise, let $l = m = 0$ and consider the following sentence entailed by the second conjunct:

$$((Q_{q_i}(\overline{0}, \overline{n}) \wedge S_\sigma(\overline{0}, \overline{n})) \rightarrow (Q_{q_j}(\overline{0}, \overline{n}') \wedge S_{\sigma'}(\overline{0}, \overline{n}') \wedge \varphi(\overline{0}, \overline{n})))$$

Either sentence implies

$$Q_{q'}(\overline{l}, \overline{n}') \wedge S_{\sigma'}(\overline{m}, \overline{n}') \wedge S_{\sigma_0}(\overline{0}, \overline{n}') \wedge \cdots \wedge S_{\sigma_k}(\overline{k}, \overline{n}') \wedge \forall x\, (\overline{k} < x \rightarrow S_0(x, \overline{n}'))$$

as before. (Note that in the first case, $\overline{l}' \equiv \overline{l+1} \equiv \overline{m}$ and in the second case $\overline{l} \equiv \overline{0}$.) But this just is $\chi(M, w, n+1)$.

3. Case (3) is left as an exercise.

We have shown that for any n, $\tau(M, w) \vDash \chi(M, w, n)$. □

Problem 32.7. Complete case (3) of the proof of Lemma 32.13.

Problem 32.8. Give a derivation of $S_{\sigma_i}(\overline{i}, \overline{n}')$ from $S_{\sigma_i}(\overline{i}, \overline{n})$ and $\varphi(m, n)$ (assuming $i \neq m$, i.e., either $i < m$ or $m < i$).

Problem 32.9. Give a derivation of $\forall x\, (\overline{k}' < x \rightarrow S_0(x, \overline{n}'))$ from $\forall x\, (\overline{k} < x \rightarrow S_0(x, \overline{n}'))$, $\forall x\, x < x'$, and $\forall x \forall y \forall z\, ((x < y \wedge y < z) \rightarrow x < z)$.)

tur:und:ver: lem:valid-if-halt

Lemma 32.14. *If M halts on input w, then $\tau(M, w) \rightarrow \alpha(M, w)$ is valid.*

Proof. By Lemma 32.13, we know that, for any time n, the description $\chi(M, w, n)$ of the configuration of M at time n is entailed by $\tau(M, w)$. Suppose M halts after k steps. At that point, it will be scanning square m, for some $m \in \mathbb{N}$. Then $\chi(M, w, k)$ describes a halting configuration of M, i.e., it contains as conjuncts both $Q_q(\overline{m}, \overline{k})$ and $S_\sigma(\overline{m}, \overline{k})$ with $\delta(q, \sigma)$ undefined. Thus, by Lemma 32.12, $\chi(M, w, k) \vDash \alpha(M, w)$. But since $\tau(M, w) \vDash \chi(M, w, k)$, we have $\tau(M, w) \vDash \alpha(M, w)$ and therefore $\tau(M, w) \rightarrow \alpha(M, w)$ is valid. □

explanation

To complete the verification of our claim, we also have to establish the reverse direction: if $\tau(M, w) \rightarrow \alpha(M, w)$ is valid, then M does in fact halt when started on input w.

tur:und:ver: lem:halt-if-valid

Lemma 32.15. *If $\vDash \tau(M, w) \rightarrow \alpha(M, w)$, then M halts on input w.*

Proof. Consider the $\mathcal{L}_M$-structure $\mathfrak{M}$ with domain $\mathbb{N}$ which interprets o as 0, $\mathsf{'}$ as the successor function, and $<$ as the less-than relation, and the predicates Q_q and S_σ as follows:

$$Q_q^{\mathfrak{M}} = \{\langle m, n\rangle : \begin{array}{l}\text{started on } w\text{, after } n \text{ steps,}\\ M \text{ is in state } q \text{ scanning square } m\end{array}\}$$

$$S_\sigma^{\mathfrak{M}} = \{\langle m, n\rangle : \begin{array}{l}\text{started on } w\text{, after } n \text{ steps,}\\ \text{square } m \text{ of } M \text{ contains symbol } \sigma\end{array}\}$$

In other words, we construct the structure $\mathfrak{M}$ so that it describes what M started on input w actually does, step by step. Clearly, $\mathfrak{M} \vDash \tau(M, w)$. If $\vDash \tau(M, w) \to \alpha(M, w)$, then also $\mathfrak{M} \vDash \alpha(M, w)$, i.e.,

$$\mathfrak{M} \vDash \exists x\, \exists y\, (\bigvee_{\langle q,\sigma\rangle \in X} (Q_q(x, y) \wedge S_\sigma(x, y))).$$

As $|\mathfrak{M}| = \mathbb{N}$, there must be m, $n \in \mathbb{N}$ so that $\mathfrak{M} \vDash Q_q(\overline{m}, \overline{n}) \wedge S_\sigma(\overline{m}, \overline{n})$ for some q and σ such that $\delta(q, \sigma)$ is undefined. By the definition of $\mathfrak{M}$, this means that M started on input w after n steps is in state q and reading symbol σ, and the transition function is undefined, i.e., M has halted. □

---

content/turing-machines/undecidability/unsolvability-decision-problem.tex

32.8 The Decision Problem is Unsolvable

tur:und:uns:sec
tur:und:uns:thm:decision-prob

Theorem 32.16. *The decision problem is unsolvable: There is no Turing machine D, which when started on a tape that contains a sentence ψ of first-order logic as input, D eventually halts, and outputs 1 iff ψ is valid and 0 otherwise.*

Proof. Suppose the decision problem were solvable, i.e., suppose there were a Turing machine D. Then we could solve the halting problem as follows. We construct a Turing machine E that, given as input the number e of Turing machine M_e and input w, computes the corresponding sentence $\tau(M_e, w) \to \alpha(M_e, w)$ and halts, scanning the leftmost square on the tape. The machine $E \frown D$ would then, given input e and w, first compute $\tau(M_e, w) \to \alpha(M_e, w)$ and then run the decision problem machine D on that input. D halts with output 1 iff $\tau(M_e, w) \to \alpha(M_e, w)$ is valid and outputs 0 otherwise. By Lemma 32.15 and Lemma 32.14, $\tau(M_e, w) \to \alpha(M_e, w)$ is valid iff M_e halts on input w. Thus, $E \frown D$, given input e and w halts with output 1 iff M_e halts on input w and halts with output 0 otherwise. In other words, $E \frown D$ would solve the halting problem. But we know, by Theorem 32.8, that no such Turing machine can exist. □

tur:und:uns: cor:undecidable-sat **Corollary 32.17.** *It is undecidable if an arbitrary sentence of first-order logic is satisfiable.*

Proof. Suppose satisfiability were decidable by a Turing machine S. Then we could solve the decision problem as follows: Given a sentence B as input, move ψ to the right one square. Return to square 1 and write the symbol $\neg$.

Now run the Turing machine S. It eventually halts with output either 1 (if $\neg\psi$ is satisfiable) or 0 (if $\neg\psi$ is unsatisfiable) on the tape. If there is a 1 on square 1, erase it; if square 1 is empty, write a 1, then halt.

This Turing machine always halts, and its output is 1 iff $\neg\psi$ is unsatisfiable and 0 otherwise. Since ψ is valid iff $\neg\psi$ is unsatisfiable, the machine outputs 1 iff ψ is valid, and 0 otherwise, i.e., it would solve the decision problem. □

explanation

So there is no Turing machine which always gives a correct "yes" or "no" answer to the question "Is ψ a valid sentence of first-order logic?" However, there *is* a Turing machine that always gives a correct "yes" answer—but simply does not halt if the answer is "no." This follows from the soundness and completeness theorem of first-order logic, and the fact that derivations can be effectively enumerated.

tur:und:uns: thm:valid-ce **Theorem 32.18.** *Validity of first-order sentences is semi-decidable: There is a Turing machine E, which when started on a tape that contains a sentence ψ of first-order logic as input, E eventually halts and outputs 1 iff ψ is valid, but does not halt otherwise.*

Proof. All possible derivations of first-order logic can be generated, one after another, by an effective algorithm. The machine E does this, and when it finds a derivation that shows that $\vdash \psi$, it halts with output 1. By the soundness theorem, if E halts with output 1, it's because $\models \psi$. By the completeness theorem, if $\models \psi$ there is a derivation that shows that $\vdash \psi$. Since E systematically generates all possible derivations, it will eventually find one that shows $\vdash \psi$, so will eventually halt with output 1. □

---

content/turing-machines/undecidability/trakhtenbrot.tex

32.9 Trakhtenbrot's Theorem

tur:und:tra: sec

explanation

In section 32.6 we defined sentences $\tau(M, w)$ and $\alpha(M, w)$ for a Turing machine M and input string w. Then we showed in Lemma 32.14 and Lemma 32.15 that $\tau(M, w) \to \alpha(M, w)$ is valid iff M, started on input w, eventually halts. Since the Halting Problem is undecidable, this implies that validity and satisfiability of sentences of first-order logic is undecidable (Theorem 32.16 and Corollary 32.17).

But validity and satisfiability of sentences is defined for arbitrary structures, finite or infinite. You might suspect that it is easier to decide if a sentence is

satisfiable in a finite structure (or valid in all finite structures). We can adapt the proof of the unsolvability of the decision problem so that it shows this is not the case.

First, if you go back to the proof of Lemma 32.15, you'll see that what we did there is produce a model $\mathfrak{M}$ of $\tau(M, w)$ which describes exactly what machine M does when started on input w. The domain of that model was $\mathbb{N}$, i.e., infinite. But if M actually halts on input w, we can build a finite model $\mathfrak{M}'$ in the same way. Suppose M started on input w halts after k steps. Take as domain $|\mathfrak{M}'|$ the set $\{0, \ldots, n\}$, where n is the larger of k and the length of w, and let

$$\prime^{\mathfrak{M}'}(x) = \begin{cases} x+1 & \text{if } x < n \\ n & \text{otherwise,} \end{cases}$$

and $\langle x, y\rangle \in <^{\mathfrak{M}'}$ iff $x < y$ or $x = y = n$. Otherwise $\mathfrak{M}'$ is defined just like $\mathfrak{M}$. By the definition of $\mathfrak{M}'$, just like in the proof of Lemma 32.15, $\mathfrak{M}' \vDash \tau(M, w)$. And since we assumed that M halts on input w, $\mathfrak{M}' \vDash \alpha(M, w)$. So, $\mathfrak{M}'$ is a finite model of $\tau(M, w) \wedge \alpha(M, w)$ (note that we've replaced $\rightarrow$ with $\wedge$).

We are halfway to a proof: we've shown that if M halts on input w, then $\tau(M, w) \wedge \alpha(M, w)$ has a finite model. Unfortunately, the converse of this does not hold, i.e., there are Turing machines that don't halt on some input w, but $\tau(M, w) \wedge \alpha(M, w)$ still has a finite model. For instance, consider the machine M with the single state q_0 and instruction $\delta(q_0, 0) = \langle q_0, 0, N\rangle$. Started on empty input $w = \Lambda$, this machine never halts: it is in an infinite loop, but does not change the tape or move the head. All configurations are the same (same state, same head position, same tape contents). We can define a finite structure $\mathfrak{M}''$ that satisfies $\tau(M, \Lambda) \wedge \alpha(M, \Lambda)$ (exercise). We can, however, change $\tau(M, w)$ in a suitable way so that such structures are ruled out.

Problem 32.10. Let M be a Turing machine with the single state q_0 and single instruction $\delta(q_0, 0) = \langle q, 0, N\rangle$. Let $|\mathfrak{M}''| = \{0, 1, 2\}$, $\prime^{\mathfrak{M}''}(0) = \prime^{\mathfrak{M}'}(1) = 1$ and $\prime^{\mathfrak{M}''}(2) = 2$, and $<^{\mathfrak{M}''} = \{\langle 0, 1\rangle, \langle 1, 1\rangle, \langle 2, 2\rangle\}$. Define $Q_{q_0}^{\mathfrak{M}''}$, $S_0^{\mathfrak{M}''}$, and $S_\triangleright^{\mathfrak{M}''}$ so that $\tau(M, \Lambda)$ and $\alpha(M, \Lambda)$ become true and explain why they are. Hint: Observe that $\delta(q_0, \triangleright)$ is undefined. Ensure that

$$Q_{q_0}(\overline{1}, \overline{n}) \wedge S_\triangleright(\overline{0}, \overline{n}) \wedge \forall x\, (\overline{0} < x \rightarrow S_0(x, \overline{n})) \quad \text{for all } n \in \mathbb{N}$$
$$\exists y\, (Q_{q_0}(\overline{0}, y) \wedge S_\triangleright(\overline{0}, y))$$

are both true in $\mathfrak{M}''$.

Consider the sentences describing the operation of the Turing machine M on input $w = \sigma_{i_1} \ldots \sigma_{i_k}$:

1. Axioms describing numbers and $<$ (just like in the definition of $\tau(M, w)$ in section 32.6).

2. Axioms describing the input configuration: just like in the definition of $\tau(M, w)$.

3. Axioms describing the transition from one configuration to the next:

   For the following, let $\varphi(x, y)$ be as before, and let

   $$\psi(y) \equiv \forall x\, (x < y \to x \neq y).$$

tur:und:tra: rep-right

   a) For every instruction $\delta(q_i, \sigma) = \langle q_j, \sigma', R\rangle$, the sentence:

   $$\forall x\, \forall y\, ((Q_{q_i}(x, y) \wedge S_\sigma(x, y)) \to (Q_{q_j}(x', y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y) \wedge \psi(y')))$$

tur:und:tra: rep-left

   b) For every instruction $\delta(q_i, \sigma) = \langle q_j, \sigma', L\rangle$, the sentence

   $$\begin{aligned}&\forall x\, \forall y\, ((Q_{q_i}(x', y) \wedge S_\sigma(x', y)) \to \\ &\qquad (Q_{q_j}(x, y') \wedge S_{\sigma'}(x', y') \wedge \varphi(x, y))) \wedge \\ &\forall y\, ((Q_{q_i}(\overline{0}, y) \wedge S_\sigma(\overline{0}, y)) \to \\ &\qquad (Q_{q_j}(\overline{0}, y') \wedge S_{\sigma'}(\overline{0}, y') \wedge \varphi(\overline{0}, y) \wedge \psi(y')))\end{aligned}$$

tur:und:tra: rep-stay

   c) For every instruction $\delta(q_i, \sigma) = \langle q_j, \sigma', N\rangle$, the sentence:

   $$\forall x\, \forall y\, ((Q_{q_i}(x, y) \wedge S_\sigma(x, y)) \to (Q_{q_j}(x, y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y) \wedge \psi(y')))$$

   As you can see, the sentences describing the transitions of M are the same as the corresponding sentence in $\tau(M, w)$, except we add $\psi(y')$ at the end. $\psi(y')$ ensures that the number y' of the "next" configuration is different from all previous numbers $\mathsf{0}$, $\mathsf{0}'$, ...

Let $\tau'(M, w)$ be the conjunction of all the above sentences for Turing machine M and input w.

tur:und:tra: lem:halts-sat

Lemma 32.19. *If M started on input w halts, then $\tau'(M, w) \wedge \alpha(M, w)$ has a finite model.*

Proof. Let $\mathfrak{M}'$ be as in the proof of Lemma 32.15, except

$$|\mathfrak{M}'| = \{0, \ldots, n\},$$

$$\prime^{\mathfrak{M}'}(x) = \begin{cases} x + 1 & \text{if } x < n \\ n & \text{otherwise,} \end{cases}$$

$$\langle x, y\rangle \in <^{\mathfrak{M}'} \text{ iff } x < y \text{ or } x = y = n,$$

where $n = \max(k, \mathrm{len}(w))$ and k is the least number such that M started on input w has halted after k steps. We leave the verification that $\mathfrak{M}' \vDash \tau'(M, w) \wedge E(M, w)$ as an exercise. □

Problem 32.11. Complete the proof of Lemma 32.19 by proving that $\mathfrak{M}' \vDash \tau(M, w) \wedge E(M, w)$.

Lemma 32.20. *If $\tau'(M,w) \wedge \alpha(M,w)$ has a finite model, then M started on input w halts.* *tur:und:tra: lem:sat-halts*

Proof. We show the contrapositive. Suppose that M started on w does not halt. If $\tau'(M,w) \wedge \alpha(M,w)$ has no model at all, we are done. So assume $\mathfrak{M}$ is a model of $\tau(M,w) \wedge \alpha(M,w)$. We have to show that it cannot be finite.

We can prove, just like in Lemma 32.13, that if M, started on input w, has not halted after n steps, then $\tau'(M,w) \vDash \chi(M,w,n) \wedge \psi(\overline{n})$. Since M started on input w does not halt, $\tau'(M,w) \vDash \chi(M,w,n) \wedge \psi(\overline{n})$ for all $n \in \mathbb{N}$. Note that by Proposition 32.10, $\tau'(M,w) \vDash \overline{k} < \overline{n}$ for all $k < n$. Also $\psi(\overline{n}) \vDash \overline{k} < \overline{n} \to \overline{k} \neq \overline{n}$. So, $\mathfrak{M} \vDash \overline{k} \neq \overline{n}$ for all $k < n$, i.e., the infinitely many terms $\overline{k}$ must all have different values in $\mathfrak{M}$. But this requires that $|\mathfrak{M}|$ be infinite, so $\mathfrak{M}$ cannot be a finite model of $\tau'(M,w) \wedge \alpha(M,w)$. □

Problem 32.12. Complete the proof of Lemma 32.20 by proving that if M, started on input w, has not halted after n steps, then $\tau'(M,w) \vDash \psi(\overline{n})$.

Theorem 32.21 (Trakhtenbrot's Theorem). *It is undecidable if an arbitrary sentence of first-order logic has a finite model (i.e., is finitely satisfiable).* *tur:und:tra: thm:trakhtenbrodt*

Proof. Suppose there were a Turing machine F that decides the finite satisfiability problem. Then given any Turing machine M and input w, we could compute the sentence $\tau'(M,w) \wedge \alpha(M,w)$, and use F to decide if it has a finite model. By Lemmata 32.19 and 32.20, it does iff M started on input w halts. So we could use F to solve the halting problem, which we know is unsolvable. □

Corollary 32.22. *There can be no derivation system that is sound and complete for* finite *validity, i.e., a derivation system which has $\vdash \psi$ iff $\mathfrak{M} \vDash \psi$ for every finite structure $\mathfrak{M}$.* *tur:und:tra: cor:fproof-incomp*

Proof. Exercise. □

Problem 32.13. Prove Corollary 32.22. Observe that ψ is satisfied in every finite structure iff $\neg\psi$ is not finitely satisfiable. Explain why finite satisfiability is semi-decidable in the sense of Theorem 32.18. Use this to argue that if there were a derivation system for finite validity, then finite satisfiability would be decidable.

Part VII

Incompleteness

Material in this part covers the incompleteness theorems. It depends on material in the parts on first-order logic (esp., the proof system), the material on recursive functions (in the computability part). It is based on Jeremy Avigad's notes with revisions by Richard Zach.

Chapter 33

Introduction to Incompleteness

content/incompleteness/introduction/historical-background.tex

33.1 Historical Background

inc:int:bgr:sec

In this section, we will briefly discuss historical developments that will help put the incompleteness theorems in context. In particular, we will give a very sketchy overview of the history of mathematical logic; and then say a few words about the history of the foundations of mathematics.

digression

The phrase "mathematical logic" is ambiguous. One can interpret the word "mathematical" as describing the subject matter, as in, "the logic of mathematics," denoting the principles of mathematical reasoning; or as describing the methods, as in "the mathematics of logic," denoting a mathematical study of the principles of reasoning. The account that follows involves mathematical logic in both senses, often at the same time.

The study of logic began, essentially, with Aristotle, who lived approximately 384–322 BCE. His *Categories*, *Prior analytics*, and *Posterior analytics* include systematic studies of the principles of scientific reasoning, including a thorough and systematic study of the syllogism.

Aristotle's logic dominated scholastic philosophy through the middle ages; indeed, as late as the eighteenth century, Kant maintained that Aristotle's logic

was perfect and in no need of revision. But the theory of the syllogism is far too limited to model anything but the most superficial aspects of mathematical reasoning. A century earlier, Leibniz, a contemporary of Newton's, imagined a complete "calculus" for logical reasoning, and made some rudimentary steps towards designing such a calculus, essentially describing a version of propositional logic.

The nineteenth century was a watershed for logic. In 1854 George Boole wrote *The Laws of Thought*, with a thorough algebraic study of propositional logic that is not far from modern presentations. In 1879 Gottlob Frege published his *Begriffsschrift* (Concept writing) which extends propositional logic with quantifiers and relations, and thus includes first-order logic. In fact, Frege's logical systems included higher-order logic as well, and more. In his *Basic Laws of Arithmetic*, Frege set out to show that all of arithmetic could be derived in his Begriffsschrift from purely logical assumption. Unfortunately, these assumptions turned out to be inconsistent, as Russell showed in 1902. But setting aside the inconsistent axiom, Frege more or less invented modern logic singlehandedly, a startling achievement. Quantificational logic was also developed independently by algebraically-minded thinkers after Boole, including Peirce and Schröder.

Let us now turn to developments in the foundations of mathematics. Of course, since logic plays an important role in mathematics, there is a good deal of interaction with the developments just described. For example, Frege developed his logic with the explicit purpose of showing that all of mathematics could be based solely on his logical framework; in particular, he wished to show that mathematics consists of a priori *analytic* truths instead of, as Kant had maintained, a priori *synthetic* ones.

Many take the birth of mathematics proper to have occurred with the Greeks. Euclid's *Elements*, written around 300 B.C., is already a mature representative of Greek mathematics, with its emphasis on rigor and precision. The definitions and proofs in Euclid's *Elements* survive more or less intact in high school geometry textbooks today (to the extent that geometry is still taught in high schools). This model of mathematical reasoning has been held to be a paradigm for rigorous argumentation not only in mathematics but in branches of philosophy as well. (Spinoza even presented moral and religious arguments in the Euclidean style, which is strange to see!)

Calculus was invented by Newton and Leibniz in the seventeenth century. (A fierce priority dispute raged for centuries, but most scholars today hold that the two developments were for the most part independent.) Calculus involves reasoning about, for example, infinite sums of infinitely small quantities; these features fueled criticism by Bishop Berkeley, who argued that belief in God was no less rational than the mathematics of his time. The methods of calculus were widely used in the eighteenth century, for example by Leonhard Euler, who used calculations involving infinite sums with dramatic results.

In the nineteenth century, mathematicians tried to address Berkeley's criticisms by putting calculus on a firmer foundation. Efforts by Cauchy, Weierstrass, Bolzano, and others led to our contemporary definitions of limits, conti-

nuity, differentiation, and integration in terms of "epsilons and deltas," in other words, devoid of any reference to infinitesimals. Later in the century, mathematicians tried to push further, and explain all aspects of calculus, including the real numbers themselves, in terms of the natural numbers. (Kronecker: "God created the whole numbers, all else is the work of man.") In 1872, Dedekind wrote "Continuity and the irrational numbers," where he showed how to "construct" the real numbers as sets of rational numbers (which, as you know, can be viewed as pairs of natural numbers); in 1888 he wrote "Was sind und was sollen die Zahlen" (roughly, "What are the natural numbers, and what should they be?") which aimed to explain the natural numbers in purely "logical" terms. In 1887 Kronecker wrote "Über den Zahlbegriff" ("On the concept of number") where he spoke of representing all mathematical object in terms of the integers; in 1889 Giuseppe Peano gave formal, symbolic axioms for the natural numbers.

The end of the nineteenth century also brought a new boldness in dealing with the infinite. Before then, infinitary objects and structures (like the set of natural numbers) were treated gingerly; "infinitely many" was understood as "as many as you want," and "approaches in the limit" was understood as "gets as close as you want." But Georg Cantor showed that it was possible to take the infinite at face value. Work by Cantor, Dedekind, and others help to introduce the general set-theoretic understanding of mathematics that is now widely accepted.

This brings us to twentieth century developments in logic and foundations. In 1902 Russell discovered the paradox in Frege's logical system. In 1904 Zermelo proved Cantor's well-ordering principle, using the so-called "axiom of choice"; the legitimacy of this axiom prompted a good deal of debate. Between 1910 and 1913 the three volumes of Russell and Whitehead's *Principia Mathematica* appeared, extending the Fregean program of establishing mathematics on logical grounds. Unfortunately, Russell and Whitehead were forced to adopt two principles that seemed hard to justify as purely logical: an axiom of infinity and an axiom of "reducibility." In the 1900's Poincaré criticized the use of "impredicative definitions" in mathematics, and in the 1910's Brouwer began proposing to refound all of mathematics in an "intuitionistic" basis, which avoided the use of the law of the excluded middle ($\varphi \vee \neg\varphi$).

Strange days indeed! The program of reducing all of mathematics to logic is now referred to as "logicism," and is commonly viewed as having failed, due to the difficulties mentioned above. The program of developing mathematics in terms of intuitionistic mental constructions is called "intuitionism," and is viewed as posing overly severe restrictions on everyday mathematics. Around the turn of the century, David Hilbert, one of the most influential mathematicians of all time, was a strong supporter of the new, abstract methods introduced by Cantor and Dedekind: "no one will drive us from the paradise that Cantor has created for us." At the same time, he was sensitive to foundational criticisms of these new methods (oddly enough, now called "classical"). He proposed a way of having one's cake and eating it too:

1. Represent classical methods with formal axioms and rules; represent mathematical questions as formulas in an axiomatic system.
2. Use safe, "finitary" methods to prove that these formal deductive systems are consistent.

Hilbert's work went a long way toward accomplishing the first goal. In 1899, he had done this for geometry in his celebrated book *Foundations of geometry*. In subsequent years, he and a number of his students and collaborators worked on other areas of mathematics to do what Hilbert had done for geometry. Hilbert himself gave axiom systems for arithmetic and analysis. Zermelo gave an axiomatization of set theory, which was expanded on by Fraenkel, Skolem, von Neumann, and others. By the mid-1920s, there were two approaches that laid claim to the title of an axiomatization of "all" of mathematics, the *Principia mathematica* of Russell and Whitehead, and what came to be known as Zermelo–Fraenkel set theory.

In 1921, Hilbert set out on a research project to establish the goal of proving these systems to be consistent. He was aided in this project by several of his students, in particular Bernays, Ackermann, and later Gentzen. The basic idea for accomplishing this goal was to cast the question of the possibility of derivation of an inconsistency in mathematics as a combinatorial problem about possible sequences of symbols, namely possible sequences of sentences which meet the criterion of being a correct derivation of, say, $\varphi \wedge \neg\varphi$ from the axioms of an axiom system for arithmetic, analysis, or set theory. A proof of the impossibility of such a sequence of symbols would—since it is itself a mathematical proof—be formalizable in these axiomatic systems. In other words, there would be some sentence Con which states that, say, arithmetic is consistent. Moreover, this sentence should be provable in the systems in question, especially if its proof requires only very restricted, "finitary" means.

The second aim, that the axiom systems developed would settle every mathematical question, can be made precise in two ways. In one way, we can formulate it as follows: For any sentence φ in the language of an axiom system for mathematics, either φ or $\neg\varphi$ is provable from the axioms. If this were true, then there would be no sentences which can neither be proved nor refuted on the basis of the axioms, no questions which the axioms do not settle. An axiom system with this property is called *complete*. Of course, for any given sentence it might still be a difficult task to determine which of the two alternatives holds. But in principle there should be a method to do so. In fact, for the axiom and derivation systems considered by Hilbert, completeness would imply that such a method exists—although Hilbert did not realize this. The second way to interpret the question would be this stronger requirement: that there be a mechanical, computational method which would determine, for a given sentence φ, whether it is derivable from the axioms or not.

In 1931, Gödel proved the two "incompleteness theorems," which showed that this program could not succeed. There is no axiom system for mathematics which is complete, specifically, the sentence that expresses the consistency of the axioms is a sentence which can neither be proved nor refuted.

This struck a lethal blow to Hilbert's original program. However, as is so often the case in mathematics, it also opened up exciting new avenues for research. If there is no one, all-encompassing formal system of mathematics, it makes sense to develop more circumscribed systems and investigate what can be proved in them. It also makes sense to develop less restricted methods of proof for establishing the consistency of these systems, and to find ways to measure how hard it is to prove their consistency. Since Gödel showed that (almost) every formal system has questions it cannot settle, it makes sense to look for "interesting" questions a given formal system cannot settle, and to figure out how strong a formal system has to be to settle them. To the present day, logicians have been pursuing these questions in a new mathematical discipline, the theory of proofs.

---

content/incompleteness/introduction/definitions.tex

33.2 Definitions

inc:int:def: sec

In order to carry out Hilbert's project of formalizing mathematics and showing that such a formalization is consistent and complete, the first order of business would be that of picking a language, logical framework, and a system of axioms. For our purposes, let us suppose that mathematics can be formalized in a first-order language, i.e., that there is some set of constant symbols, function symbols, and predicate symbols which, together with the connectives and quantifiers of first-order logic, allow us to express the claims of mathematics. Most people agree that such a language exists: the language of set theory, in which $\in$ is the only non-logical symbol. That such a simple language is so expressive is of course a very implausible claim at first sight, and it took a lot of work to establish that practically all of mathematics can be expressed in this very austere vocabulary. To keep things simple, for now, let's restrict our discussion to arithmetic, so the part of mathematics that just deals with the natural numbers $\mathbb{N}$. The natural language in which to express facts of arithmetic is $\mathcal{L}_A$. $\mathcal{L}_A$ contains a single two-place predicate symbol $<$, a single constant symbol $\mathsf{0}$, one one-place function symbol $\prime$, and two two-place function symbols $+$ and $\times$.

Definition 33.1. A set of sentences Γ is a *theory* if it is closed under entailment, i.e., if $\Gamma = \{\varphi : \Gamma \vDash \varphi\}$.

There are two easy ways to specify theories. One is as the set of sentences true in some structure. For instance, consider the structure for $\mathcal{L}_A$ in which the domain is $\mathbb{N}$ and all non-logical symbols are interpreted as you would expect.

inc:int:def: def:standard-model

Definition 33.2. The *standard model of arithmetic* is the structure $\mathfrak{N}$ defined as follows:

1. $|\mathfrak{N}| = \mathbb{N}$

2. $\mathsf{o}^{\mathfrak{N}} = 0$

3. $\mathsf{'}^{\mathfrak{N}}(n) = n + 1$ for all $n \in \mathbb{N}$

4. $+^{\mathfrak{N}}(n, m) = n + m$ for all $n, m \in \mathbb{N}$

5. $\times^{\mathfrak{N}}(n, m) = n \cdot m$ for all $n, m \in \mathbb{N}$

6. $<^{\mathfrak{N}} = \{\langle n, m\rangle : n \in \mathbb{N}, m \in \mathbb{N}, n < m\}$

Note the difference between '$\times$' and '$\cdot$': $\times$ is a symbol in the language of arithmetic. Of course, we've chosen it to remind us of multiplication, but $\times$ is not the multiplication operation but a two-place function symbol (officially, f^2_1). By contrast, $\cdot$ *is* the ordinary multiplication function. When you see something like $n \cdot m$, we mean the product of the numbers n and m; when you see something like $x \times y$ we are talking about a term in the language of arithmetic. In the standard model, the function symbol $\times$ is interpreted as the function $\cdot$ on the natural numbers. For addition, we use $+$ as both the function symbol of the language of arithmetic, and the addition function on the natural numbers. Here you have to use the context to determine what is meant.

Definition 33.3. The theory of *true arithmetic* is the set of sentences satisfied in the standard model of arithmetic, i.e.,

$$\mathbf{TA} = \{\varphi : \mathfrak{N} \vDash \varphi\}.$$

TA is a theory, for whenever $\mathbf{TA} \vDash \varphi$, φ is satisfied in every structure which satisfies **TA**. Since $\mathfrak{N} \vDash \mathbf{TA}$, we have that $\mathfrak{N} \vDash \varphi$, and so $\varphi \in \mathbf{TA}$.

The other way to specify a theory Γ is as the set of sentences entailed by some set of sentences Γ_0. In that case, Γ is the "closure" of Γ_0 under entailment. Specifying a theory this way is only interesting if Γ_0 is explicitly specified, e.g., if the elements of Γ_0 are listed. At the very least, Γ_0 has to be decidable, i.e., there has to be a computable test for when a sentence counts as an element of Γ_0 or not. We call the sentences in Γ_0 *axioms* for Γ, and Γ *axiomatized* by Γ_0.

Definition 33.4. A theory Γ is *axiomatized* by Γ_0 iff

$$\Gamma = \{\varphi : \Gamma_0 \vDash \varphi\}$$

Definition 33.5. The theory **Q** axiomatized by the following sentences is known as "Robinson's **Q**" and is a very simple theory of arithmetic.

$$\forall x \forall y\, (x' = y' \to x = y) \tag{Q_1}$$
$$\forall x\, \mathsf{o} \neq x' \tag{Q_2}$$
$$\forall x\, (x = \mathsf{o} \vee \exists y\, x = y') \tag{Q_3}$$
$$\forall x\, (x + \mathsf{o}) = x \tag{Q_4}$$
$$\forall x \forall y\, (x + y') = (x + y)' \tag{Q_5}$$
$$\forall x\, (x \times \mathsf{o}) = \mathsf{o} \tag{Q_6}$$
$$\forall x \forall y\, (x \times y') = ((x \times y) + x) \tag{Q_7}$$
$$\forall x \forall y\, (x < y \leftrightarrow \exists z\, (z' + x) = y) \tag{Q_8}$$

The set of sentences $\{Q_1, \ldots, Q_8\}$ are the axioms of **Q**, so **Q** consists of all sentences entailed by them:

$$\mathbf{Q} = \{\varphi : \{Q_1, \ldots, Q_8\} \vDash \varphi\}.$$

Definition 33.6. Suppose $\varphi(x)$ is a formula in $\mathcal{L}_A$ with free variables x and $y_1, \ldots, y_n$. Then any sentence of the form

$$\forall y_1 \ldots \forall y_n\, ((\varphi(\mathsf{o}) \wedge \forall x\, (\varphi(x) \to \varphi(x'))) \to \forall x\, \varphi(x))$$

is an instance of the *induction schema*.

Peano arithmetic **PA** is the theory axiomatized by the axioms of **Q** together with all instances of the induction schema.

Every instance of the induction schema is true in $\mathfrak{N}$. This is easiest to see if the formula φ only has one free variable x. Then $\varphi(x)$ defines a subset X_φ of $\mathbb{N}$ in $\mathfrak{N}$. X_φ is the set of all $n \in \mathbb{N}$ such that $\mathfrak{N}, s \vDash \varphi(x)$ when $s(x) = n$. The corresponding instance of the induction schema is explanation

$$((\varphi(\mathsf{o}) \wedge \forall x\, (\varphi(x) \to \varphi(x'))) \to \forall x\, \varphi(x)).$$

If its antecedent is true in $\mathfrak{N}$, then $0 \in X_\varphi$ and, whenever $n \in X_\varphi$, so is $n + 1$. Since $0 \in X_\varphi$, we get $1 \in X_\varphi$. With $1 \in X_\varphi$ we get $2 \in X_\varphi$. And so on. So for every $n \in \mathbb{N}$, $n \in X_\varphi$. But this means that $\forall x\, \varphi(x)$ is satisfied in $\mathfrak{N}$.

Both **Q** and **PA** are axiomatized theories. The big question is, how strong are they? For instance, can **PA** prove all the truths about $\mathbb{N}$ that can be expressed in $\mathcal{L}_A$? Specifically, do the axioms of **PA** settle all the questions that can be formulated in $\mathcal{L}_A$?

Another way to put this is to ask: Is **PA** = **TA**? **TA** obviously does prove (i.e., it includes) all the truths about $\mathbb{N}$, and it settles all the questions that can be formulated in $\mathcal{L}_A$, since if φ is a sentence in $\mathcal{L}_A$, then either $\mathfrak{N} \vDash \varphi$ or $\mathfrak{N} \vDash \neg\varphi$, and so either **TA** $\vDash \varphi$ or **TA** $\vDash \neg\varphi$. Call such a theory *complete*.

Definition 33.7. A theory Γ is *complete* iff for every sentence φ in its language, either $\Gamma \vDash \varphi$ or $\Gamma \vDash \neg\varphi$.

explanation

By the Completeness Theorem, $\Gamma \vDash \varphi$ iff $\Gamma \vdash \varphi$, so Γ is complete iff for every sentence φ in its language, either $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$.

Another question we are led to ask is this: Is there a computational procedure we can use to test if a sentence is in **TA**, in **PA**, or even just in **Q**? We can make this more precise by defining when a set (e.g., a set of sentences) is decidable.

Definition 33.8. A set X is *decidable* iff there is a computational procedure which on input x returns 1 if $x \in X$ and 0 otherwise.

So our question becomes: Is **TA** (**PA**, **Q**) decidable?

The answer to all these questions will be: no. None of these theories are decidable. However, this phenomenon is not specific to these particular theories. In fact, *any* theory that satisfies certain conditions is subject to the same results. One of these conditions, which **Q** and **PA** satisfy, is that they are axiomatized by a decidable set of axioms.

Definition 33.9. A theory is *axiomatizable* if it is axiomatized by a decidable set of axioms.

Example 33.10. Any theory axiomatized by a finite set of sentences is axiomatizable, since any finite set is decidable. Thus, **Q**, for instance, is axiomatizable.

Schematically axiomatized theories like **PA** are also axiomatizable. For to test if ψ is among the axioms of **PA**, i.e., to compute the function χ_X where $\chi_X(\psi) = 1$ if ψ is an axiom of **PA** and $= 0$ otherwise, we can do the following: First, check if ψ is one of the axioms of **Q**. If it is, the answer is "yes" and the value of $\chi_X(\psi) = 1$. If not, test if it is an instance of the induction schema. This can be done systematically; in this case, perhaps it's easiest to see that it can be done as follows: Any instance of the induction schema begins with a number of universal quantifiers, and then a sub-formula that is a conditional. The consequent of that conditional is $\forall x\, \varphi(x, y_1, \ldots, y_n)$ where x and y_1, …, y_n are all the free variables of φ and the initial quantifiers of ψ bind the variables y_1, …, y_n. Once we have extracted this φ and checked that its free variables match the variables bound by the universal quantifiers at the front and $\forall x$, we go on to check that the antecedent of the conditional matches

$$\varphi(\mathsf{0}, y_1, \ldots, y_n) \wedge \forall x\, (\varphi(x, y_1, \ldots, y_n) \to \varphi(x', y_1, \ldots, y_n))$$

Again, if it does, ψ is an instance of the induction schema, and if it doesn't, ψ isn't.

In answering this question—and the more general question of which theories are complete or decidable—it will be useful to consider also the following definition. Recall that a set X is enumerable iff it is empty or if there is a surjective function $f\colon \mathbb{N} \to X$. Such a function is called an enumeration of X.

Definition 33.11. A set X is called *computably enumerable* (c.e. for short) iff it is empty or it has a computable enumeration.

In addition to axiomatizability, another condition on theories to which the incompleteness theorems apply will be that they are strong enough to prove basic facts about computable functions and decidable relations. By "basic facts," we mean sentences which express what the values of computable functions are for each of their arguments. And by "strong enough" we mean that the theories in question count these sentences among its theorems. For instance, consider a prototypical computable function: addition. The value of $+$ for arguments 2 and 3 is 5, i.e., $2 + 3 = 5$. A sentence in the language of arithmetic that expresses that the value of $+$ for arguments 2 and 3 is 5 is: $(\overline{2} + \overline{3}) = \overline{5}$. And, e.g., **Q** proves this sentence. More generally, we would like there to be, for each computable function $f(x_1, x_2)$ a formula $\varphi_f(x_1, x_2, y)$ in $\mathcal{L}_A$ such that $\mathbf{Q} \vdash \varphi_f(\overline{n_1}, \overline{n_2}, \overline{m})$ whenever $f(n_1, n_2) = m$. In this way, **Q** proves that the value of f for arguments n_1, n_2 is m. In fact, we require that it proves a bit more, namely that no other number is the value of f for arguments n_1, n_2. And the same goes for decidable relations. This is made precise in the following two definitions.

Definition 33.12. A formula $\varphi(x_1, \ldots, x_k, y)$ *represents* the function $f\colon \mathbb{N}^k \to \mathbb{N}$ in Γ iff whenever $f(n_1, \ldots, n_k) = m$, then

1. $\Gamma \vdash \varphi(\overline{n_1}, \ldots, \overline{n_k}, \overline{m})$, and
2. $\Gamma \vdash \forall y(\varphi(\overline{n_1}, \ldots, \overline{n_k}, y) \to y = \overline{m})$.

Definition 33.13. A formula $\varphi(x_1, \ldots, x_k)$ *represents* the relation $R \subseteq \mathbb{N}^k$ iff,

1. whenever $R(n_1, \ldots, n_k)$, $\Gamma \vdash \varphi(\overline{n_1}, \ldots, \overline{n_k})$, and
2. whenever not $R(n_1, \ldots, n_k)$, $\Gamma \vdash \neg\varphi(\overline{n_1}, \ldots, \overline{n_k})$.

A theory is "strong enough" for the incompleteness theorems to apply if it represents all computable functions and all decidable relations. **Q** and its extensions satisfy this condition, but it will take us a while to establish this—it's a non-trivial fact about the kinds of things **Q** can prove, and it's hard to show because **Q** has only a few axioms from which we'll have to prove all these facts. However, **Q** is a very weak theory. So although it's hard to prove that **Q** represents all computable functions, most interesting theories are stronger than **Q**, i.e., prove more than **Q** does. And if **Q** proves something, any stronger theory does; since **Q** represents all computable functions, every stronger theory does. This means that many interesting theories meet this condition of the incompleteness theorems. So our hard work will pay off, since it shows that the incompleteness theorems apply to a wide range of theories. Certainly, any theory aiming to formalize "all of mathematics" must prove everything that **Q** proves, since it should at the very least be able to capture the results of

elementary computations. So any theory that is a candidate for a theory of "all of mathematics" will be one to which the incompleteness theorems apply.

---

content/incompleteness/introduction/overview.tex

33.3 Overview of Incompleteness Results

inc:int:ovr:sec

Hilbert expected that mathematics could be formalized in an axiomatizable theory which it would be possible to prove complete and decidable. Moreover, he aimed to prove the consistency of this theory with very weak, "finitary," means, which would defend classical mathematics against the challenges of intuitionism. Gödel's incompleteness theorems showed that these goals cannot be achieved.

Gödel's first incompleteness theorem showed that a version of Russell and Whitehead's *Principia Mathematica* is not complete. But the proof was actually very general and applies to a wide variety of theories. This means that it wasn't just that *Principia Mathematica* did not manage to completely capture mathematics, but that *no* acceptable theory does. It took a while to isolate the features of theories that suffice for the incompleteness theorems to apply, and to generalize Gödel's proof to apply make it depend only on these features. But we are now in a position to state a very general version of the first incompleteness theorem for theories in the language $\mathcal{L}_A$ of arithmetic.

Theorem 33.14. *If Γ is a consistent and axiomatizable theory in $\mathcal{L}_A$ which represents all computable functions and decidable relations, then Γ is not complete.*

To say that Γ is not complete is to say that for at least one sentence φ, $\Gamma \nvdash \varphi$ and $\Gamma \nvdash \neg\varphi$. Such a sentence is called *independent* (of Γ). We can in fact relatively quickly prove that there must be independent sentences. But the power of Gödel's proof of the theorem lies in the fact that it exhibits a *specific example* of such an independent sentence. The intriguing construction produces a sentence γ_Γ, called a *Gödel sentence* for Γ, which is unprovable because in Γ, γ_Γ is equivalent to the claim that γ_Γ is unprovable in Γ. It does so *constructively*, i.e., given an axiomatization of Γ and a description of the derivation system, the proof gives a method for actually writing down γ_Γ.

The construction in Gödel's proof requires that we find a way to express in $\mathcal{L}_A$ the properties of and operations on terms and formulas of $\mathcal{L}_A$ itself. These include properties such as "φ is a sentence," "δ is a derivation of φ," and operations such as $\varphi[t/x]$. This way must (a) express these properties and relations via a "coding" of symbols and sequences thereof (which is what terms, formulas, derivations, etc. are) as natural numbers (which is what $\mathcal{L}_A$ can talk about). It must (b) do this in such a way that Γ will prove the relevant facts, so we must show that these properties are coded by decidable properties of natural numbers and the operations correspond to computable functions on natural numbers. This is called "arithmetization of syntax."

Before we investigate how syntax can be arithmetized, however, we will consider the condition that Γ is "strong enough," i.e., represents all computable functions and decidable relations. This requires that we give a precise definition of "computable." This can be done in a number of ways, e.g., via the model of Turing machines, or as those functions computable by programs in some general-purpose programming language. Since our aim is to represent these functions and relations in a theory in the language $\mathcal{L}_A$, however, it is best to pick a simple definition of computability of just numerical functions. This is the notion of *recursive function*. So we will first discuss the recursive functions. We will then show that **Q** already represents all recursive functions and relations. This will allow us to apply the incompleteness theorem to specific theories such as **Q** and **PA**, since we will have established that these are examples of theories that are "strong enough."

The end result of the arithmetization of syntax is a formula $\mathsf{Prov}_\Gamma(x)$ which, via the coding of formulas as numbers, expresses provability from the axioms of Γ. Specifically, if φ is coded by the number n, and $\Gamma \vdash \varphi$, then $\Gamma \vdash \mathrm{Prov}_\Gamma(\overline{n})$. This "provability predicate" for Γ allows us also to express, in a certain sense, the consistency of Γ as a sentence of $\mathcal{L}_A$: let the "consistency statement" for Γ be the sentence $\neg\mathrm{Prov}_\Gamma(\overline{n})$, where we take n to be the code of a contradiction, e.g., of $\bot$. The second incompleteness theorem states that consistent axiomatizable theories also do not prove their own consistency statements. The conditions required for this theorem to apply are a bit more stringent than just that the theory represents all computable functions and decidable relations, but we will show that **PA** satisfies them.

---

content/incompleteness/introduction/undecidability.tex

33.4 Undecidability and Incompleteness

inc:int:dec: sec

Gödel's proof of the incompleteness theorems require arithmetization of syntax. But even without that we can obtain some nice results just on the assumption that a theory represents all decidable relations. The proof is a diagonal argument similar to the proof of the undecidability of the halting problem.

Theorem 33.15. *If Γ is a consistent theory that represents every decidable relation, then Γ is not decidable.*

Proof. Suppose Γ were decidable. We show that if Γ represents every decidable relation, it must be inconsistent.

Decidable properties (one-place relations) are represented by formulas with one free variable. Let $\varphi_0(x)$, $\varphi_1(x)$, ..., be a computable enumeration of all such formulas. Now consider the following set $D \subseteq \mathbb{N}$:

$$D = \{n : \Gamma \vdash \neg\varphi_n(\overline{n})\}$$

The set D is decidable, since we can test if $n \in D$ by first computing $\varphi_n(x)$, and from this $\neg\varphi_n(\overline{n})$. Obviously, substituting the term $\overline{n}$ for every free occurrence

of x in $\varphi_n(x)$ and prefixing $\varphi(\overline{n})$ by $\neg$ is a mechanical matter. By assumption, Γ is decidable, so we can test if $\neg\varphi(\overline{n}) \in \Gamma$. If it is, $n \in D$, and if it isn't, $n \notin D$. So D is likewise decidable.

Since Γ represents all decidable properties, it represents D. And the formulas which represent D in Γ are all among $\varphi_0(x)$, $\varphi_1(x)$, ... So let d be a number such that $\varphi_d(x)$ represents D in Γ. If $d \notin D$, then, since $\varphi_d(x)$ represents D, $\Gamma \vdash \neg\varphi_d(\overline{d})$. But that means that d meets the defining condition of D, and so $d \in D$. This contradicts $d \notin D$. So by indirect proof, $d \in D$.

Since $d \in D$, by the definition of D, $\Gamma \vdash \neg\varphi_d(\overline{d})$. On the other hand, since $\varphi_d(x)$ represents D in Γ, $\Gamma \vdash \varphi_d(\overline{d})$. Hence, Γ is inconsistent. □

explanation

The preceding theorem shows that no consistent theory that represents all decidable relations can be decidable. We will show that **Q** does represent all decidable relations; this means that all theories that include **Q**, such as **PA** and **TA**, also do, and hence also are not decidable. (Since all these theories are true in the standard model, they are all consistent.)

We can also use this result to obtain a weak version of the first incompleteness theorem. Any theory that is axiomatizable and complete is decidable. Consistent theories that are axiomatizable and represent all decidable properties then cannot be complete.

Theorem 33.16. *If Γ is axiomatizable and complete it is decidable.*

Proof. Any inconsistent theory is decidable, since inconsistent theories contain all sentences, so the answer to the question "is $\varphi \in \Gamma$" is always "yes," i.e., can be decided.

So suppose Γ is consistent, and furthermore is axiomatizable, and complete. Since Γ is axiomatizable, it is computably enumerable. For we can enumerate all the correct derivations from the axioms of Γ by a computable function. From a correct derivation we can compute the sentence it derives, and so together there is a computable function that enumerates all theorems of Γ. A sentence is a theorem of Γ iff $\neg\varphi$ is not a theorem, since Γ is consistent and complete. We can therefore decide if $\varphi \in \Gamma$ as follows. Enumerate all theorems of Γ. When φ appears on this list, we know that $\Gamma \vdash \varphi$. When $\neg\varphi$ appears on this list, we know that $\Gamma \nvdash \varphi$. Since Γ is complete, one of these cases eventually obtains, so the procedure eventually produces an answer. □

Corollary 33.17. *If Γ is consistent, axiomatizable, and represents every decidable property, it is not complete.*

inc:int:dec: cor:incompleteness

Proof. If Γ were complete, it would be decidable by the previous theorem (since it is axiomatizable and consistent). But since Γ represents every decidable property, it is not decidable, by the first theorem. □

Problem 33.1. Show that $\mathbf{TA} = \{\varphi : \mathfrak{N} \vDash \varphi\}$ is not axiomatizable. You may assume that **TA** represents all decidable properties.

Once we have established that, e.g., **Q**, represents all decidable properties, the corollary tells us that **Q** must be incomplete. However, its proof does not provide an example of an independent sentence; it merely shows that such a sentence must exist. For this, we have to arithmetize syntax and follow Gödel's original proof idea. And of course, we still have to show the first claim, namely that **Q** does, in fact, represent all decidable properties.

It should be noted that not every *interesting* theory is incomplete or undecidable. There are many theories that are sufficiently strong to describe interesting mathematical facts that do not satisify the conditions of Gödel's result. For instance, $\mathbf{Pres} = \{\varphi \in \mathcal{L}_{A^+} : \mathfrak{N} \vDash \varphi\}$, the set of sentences of the language of arithmetic without $\times$ true in the standard model, is both complete and decidable. This theory is called Presburger arithmetic, and proves all the truths about natural numbers that can be formulated just with 0, $'$, and $+$.

Chapter 34

Arithmetization of Syntax

Note that arithmetization for signed tableaux is not yet available.

---

content/incompleteness/arithmetization-syntax/introduction.tex

34.1 Introduction

inc:art:int:sec

In order to connect computability and logic, we need a way to talk about the objects of logic (symbols, terms, formulas, derivations), operations on them, and their properties and relations, in a way amenable to computational treatment. We can do this directly, by considering computable functions and relations on symbols, sequences of symbols, and other objects built from them. Since the objects of logical syntax are all finite and built from an enumerable sets of symbols, this is possible for some models of computation. But other models of computation—such as the recursive functions—-are restricted to numbers, their relations and functions. Moreover, ultimately we also want to be able to deal with syntax within certain theories, specifically, in theories formulated in

the language of arithmetic. In these cases it is necessary to *arithmetize* syntax, i.e., to represent syntactic objects, operations on them, and their relations, as numbers, arithmetical functions, and arithmetical relations, respectively. The idea, which goes back to Leibniz, is to assign numbers to syntactic objects.

It is relatively straightforward to assign numbers to symbols as their "codes." Some symbols pose a bit of a challenge, since, e.g., there are infinitely many variables, and even infinitely many function symbols of each arity n. But of course it's possible to assign numbers to symbols systematically in such a way that, say, v_2 and v_3 are assigned different codes. Sequences of symbols (such as terms and formulas) are a bigger challenge. But if we can deal with sequences of numbers purely arithmetically (e.g., by the powers-of-primes coding of sequences), we can extend the coding of individual symbols to coding of sequences of symbols, and then further to sequences or other arrangements of formulas, such as derivations. This extended coding is called "Gödel numbering." Every term, formula, and derivation is assigned a Gödel number.

By coding sequences of symbols as sequences of their codes, and by choosing a system of coding sequences that can be dealt with using computable functions, we can then also deal with Gödel numbers using computable functions. In practice, all the relevant functions will be primitive recursive. For instance, computing the length of a sequence and computing the i-th element of a sequence from the code of the sequence are both primitive recursive. If the number coding the sequence is, e.g., the Gödel number of a formula φ, we immediately see that the length of a formula and the (code of the) i-th symbol in a formula can also be computed from the Gödel number of φ. It is a bit harder to prove that, e.g., the property of being the Gödel number of a correctly formed term or of a correct derivation is primitive recursive. It is nevertheless possible, because the sequences of interest (terms, formulas, derivations) are inductively defined.

As an example, consider the operation of substitution. If φ is a formula, x a variable, and t a term, then $\varphi[t/x]$ is the result of replacing every free occurrence of x in φ by t. Now suppose we have assigned Gödel numbers to φ, x, t—say, k, l, and m, respectively. The same scheme assigns a Gödel number to $\varphi[t/x]$, say, n. This mapping—of k, l, and m to n—is the arithmetical analog of the substitution operation. When the substitution operation maps φ, x, t to $\varphi[t/x]$, the arithmetized substitution functions maps the Gödel numbers k, l, m to the Gödel number n. We will see that this function is primitive recursive.

Arithmetization of syntax is not just of abstract interest, although it was originally a non-trivial insight that languages like the language of arithmetic, which do not come with mechanisms for "talking about" languages can, after all, formalize complex properties of expressions. It is then just a small step to ask what a theory in this language, such as Peano arithmetic, can *prove* about its own language (including, e.g., whether sentences are provable or true). This leads us to the famous limitative theorems of Gödel (about unprovability) and Tarski (the undefinability of truth). But the trick of arithmetizing syntax is also important in order to prove some important results in computability theory, e.g., about the computational power of theories or the relationship between

different models of computability. The arithmetization of syntax serves as a model for arithmetizing other objects and properties. For instance, it is similarly possible to arithmetize configurations and computations (say, of Turing machines). This makes it possible to simulate computations in one model (e.g., Turing machines) in another (e.g., recursive functions).

---

content/incompleteness/arithmetization-syntax/coding-symbols.tex

34.2 Coding Symbols

inc:art:cod:sec

The basic language $\mathcal{L}$ of first order logic makes use of the symbols

$$\bot \quad \neg \quad \vee \quad \wedge \quad \to \quad \forall \quad \exists \quad = \quad (\quad) \quad ,$$

together with enumerable sets of variables and constant symbols, and enumerable sets of function symbols and predicate symbols of arbitrary arity. We can assign *codes* to each of these symbols in such a way that every symbol is assigned a unique number as its code, and no two different symbols are assigned the same number. We know that this is possible since the set of all symbols is enumerable and so there is a bijection between it and the set of natural numbers. But we want to make sure that we can recover the symbol (as well as some information about it, e.g., the arity of a function symbol) from its code in a computable way. There are many possible ways of doing this, of course. Here is one such way, which uses primitive recursive functions. (Recall that $\langle n_0, \ldots, n_k \rangle$ is the number coding the sequence of numbers $n_0, \ldots, n_k$.)

Definition 34.1. If s is a symbol of $\mathcal{L}$, let the *symbol code* c_s be defined as follows:

1. If s is among the logical symbols, c_s is given by the following table:

$\bot$	$\neg$	$\vee$	$\wedge$	$\to$	$\forall$
$\langle 0, 0 \rangle$	$\langle 0, 1 \rangle$	$\langle 0, 2 \rangle$	$\langle 0, 3 \rangle$	$\langle 0, 4 \rangle$	$\langle 0, 5 \rangle$
$\exists$	$=$	$($	$)$	$,$	
$\langle 0, 6 \rangle$	$\langle 0, 7 \rangle$	$\langle 0, 8 \rangle$	$\langle 0, 9 \rangle$	$\langle 0, 10 \rangle$	

2. If s is the i-th variable v_i, then $\mathrm{c}_s = \langle 1, i \rangle$.
3. If s is the i-th constant symbol c_i, then $\mathrm{c}_s = \langle 2, i \rangle$.
4. If s is the i-th n-ary function symbol f^n_i, then $\mathrm{c}_s = \langle 3, n, i \rangle$.
5. If s is the i-th n-ary predicate symbol P^n_i, then $\mathrm{c}_s = \langle 4, n, i \rangle$.

Proposition 34.2. *The following relations are primitive recursive:*

1. $\mathrm{Fn}(x, n)$ *iff* x *is the code of* f^n_i *for some* i*, i.e.,* x *is the code of an* n*-ary function symbol.*

2. $\mathrm{Pred}(x, n)$ *iff* x *is the code of* P_i^n *for some* i *or* x *is the code of* $=$ *and* $n = 2$*, i.e.,* x *is the code of an* n*-ary predicate symbol.*

Definition 34.3. If $s_0, \ldots, s_{n-1}$ is a sequence of symbols, its *Gödel number* is $\langle \mathrm{c}_{s_0}, \ldots, \mathrm{c}_{s_{n-1}} \rangle$.

explanation

Note that *codes* and *Gödel numbers* are different things. For instance, the variable v_5 has a code $\mathrm{c}_{v_5} = \langle 1, 5 \rangle = 2^2 \cdot 3^6$. But the variable v_5 considered as a term is also a sequence of symbols (of length 1). The *Gödel number* ${}^{\ulcorner}v_5{}^{\urcorner}$ of the *term* v_5 is $\langle \mathrm{c}_{v_5} \rangle = 2^{\mathrm{c}_{v_5}+1} = 2^{2^2 \cdot 3^6 + 1}$.

Example 34.4. Recall that if $k_0, \ldots, k_{n-1}$ is a sequence of numbers, then the code of the sequence $\langle k_0, \ldots, k_{n-1} \rangle$ in the power-of-primes coding is

$$2^{k_0+1} \cdot 3^{k_1+1} \cdot \ldots \cdot p_{n-1}^{k_{n-1}+1},$$

where p_i is the i-th prime (starting with $p_0 = 2$). So for instance, the formula $v_0 = \mathsf{o}$, or, more explicitly, $=(v_0, c_0)$, has the Gödel number

$$\langle \mathrm{c}_=, \mathrm{c}_(, \mathrm{c}_{v_0}, \mathrm{c}_,, \mathrm{c}_{c_0}, \mathrm{c}_) \rangle.$$

Here, $\mathrm{c}_=$ is $\langle 0, 7 \rangle = 2^{0+1} \cdot 3^{7+1}$, c_{v_0} is $\langle 1, 0 \rangle = 2^{1+1} \cdot 3^{0+1}$, etc. So ${}^{\ulcorner}=(v_0, c_0){}^{\urcorner}$ is

$$\begin{aligned} &2^{\mathrm{c}_=+1} \cdot 3^{\mathrm{c}_(+1} \cdot 5^{\mathrm{c}_{v_0}+1} \cdot 7^{\mathrm{c}_,+1} \cdot 11^{\mathrm{c}_{c_0}+1} \cdot 13^{\mathrm{c}_)+1} = \\ &\qquad 2^{2^1 \cdot 3^8+1} \cdot 3^{2^1 \cdot 3^9+1} \cdot 5^{2^2 \cdot 3^1+1} \cdot 7^{2^1 \cdot 3^{11}+1} \cdot 11^{2^3 \cdot 3^1+1} \cdot 13^{2^1 \cdot 3^{10}+1} = \\ &\qquad\qquad 2^{13\,123} \cdot 3^{39\,367} \cdot 5^{13} \cdot 7^{354\,295} \cdot 11^{25} \cdot 13^{118\,099}. \end{aligned}$$

---

content/incompleteness/arithmetization-syntax/coding-terms.tex

34.3 Coding Terms

explanation

A term is simply a certain kind of sequence of symbols: it is built up inductively from constants and variables according to the formation rules for terms. Since sequences of symbols can be coded as numbers—using a coding scheme for the symbols plus a way to code sequences of numbers—assigning Gödel numbers to terms is not difficult. The challenge is rather to show that the property a number has if it is the Gödel number of a correctly formed term is computable, or in fact primitive recursive.

inc:art:trm:sec

Variables and constant symbols are the simplest terms, and testing whether x is the Gödel number of such a term is easy: $\mathrm{Var}(x)$ holds if x is ${}^{\ulcorner}v_i{}^{\urcorner}$ for some i. In other words, x is a sequence of length 1 and its single element $(x)_0$ is the code of some variable v_i, i.e., x is $\langle \langle 1, i \rangle \rangle$ for some i. Similarly, $\mathrm{Const}(x)$ holds if x is ${}^{\ulcorner}c_i{}^{\urcorner}$ for some i. Both of these relations are primitive recursive, since if such an i exists, it must be $< x$:

$$\begin{aligned} \mathrm{Var}(x) &\Leftrightarrow (\exists i < x)\ x = \langle \langle 1, i \rangle \rangle \\ \mathrm{Const}(x) &\Leftrightarrow (\exists i < x)\ x = \langle \langle 2, i \rangle \rangle \end{aligned}$$

inc:art:trm: prop:term-primrec **Proposition 34.5.** *The relations* Term(x) *and* ClTerm(x) *which hold iff x is the Gödel number of a term or a closed term, respectively, are primitive recursive.*

Proof. A sequence of symbols s is a term iff there is a sequence $s_0, \ldots, s_{k-1} = s$ of terms which records how the term s was formed from constant symbols and variables according to the formation rules for terms. To express that such a putative formation sequence follows the formation rules it has to be the case that, for each $i < k$, either

1. s_i is a variable v_j, or
2. s_i is a constant symbol c_j, or
3. s_i is built from n terms $t_1, \ldots, t_n$ occurring prior to place i using an n-place function symbol f^n_j.

To show that the corresponding relation on Gödel numbers is primitive recursive, we have to express this condition primitive recursively, i.e., using primitive recursive functions, relations, and bounded quantification.

Suppose y is the number that codes the sequence $s_0, \ldots, s_{k-1}$, i.e., $y = \langle {}^{\ulcorner}s_0{}^{\urcorner}, \ldots, {}^{\ulcorner}s_{k-1}{}^{\urcorner}\rangle$. It codes a formation sequence for the term with Gödel number x iff for all $i < k$:

1. $\mathrm{Var}((y)_i)$, or
2. $\mathrm{Const}((y)_i)$, or
3. there is an n and a number $z = \langle z_1, \ldots, z_n\rangle$ such that each z_l is equal to some $(y)_{i'}$ for $i' < i$ and

$$(y)_i = {}^{\ulcorner}\mathsf{f}^n_j({}^{\urcorner} \frown \mathrm{flatten}(z) \frown {}^{\ulcorner}){}^{\urcorner},$$

and moreover $(y)_{k-1} = x$. (The function flatten(z) turns the sequence $\langle {}^{\ulcorner}t_1{}^{\urcorner}, \ldots, {}^{\ulcorner}t_n{}^{\urcorner}\rangle$ into ${}^{\ulcorner}t_1, \ldots, t_n{}^{\urcorner}$ and is primitive recursive.)

The indices j, n, the Gödel numbers z_l of the terms t_l, and the code z of the sequence $\langle z_1, \ldots, z_n\rangle$, in (3) are all less than y. We can replace k above with len(y). Hence we can express "y is the code of a formation sequence of the term with Gödel number x" in a way that shows that this relation is primitive recursive.

We now just have to convince ourselves that there is a primitive recursive bound on y. But if x is the Gödel number of a term, it must have a formation sequence with at most len(x) terms (since every term in the formation sequence of s must start at some place in s, and no two subterms can start at the same place). The Gödel number of each subterm of s is of course $\leq x$. Hence, there always is a formation sequence with code $\leq p_{k-1}^{k(x+1)}$, where $k = \mathrm{len}(x)$.

For ClTerm, simply leave out the clause for variables. □

Problem 34.1. Show that the function flatten(z), which turns the sequence $\langle {}^{\ulcorner}t_1{}^{\urcorner}, \ldots, {}^{\ulcorner}t_n{}^{\urcorner}\rangle$ into ${}^{\ulcorner}t_1, \ldots, t_n{}^{\urcorner}$, is primitive recursive.

Proposition 34.6. *The function* $\mathrm{num}(n) = {}^{\ulcorner}\overline{n}^{\urcorner}$ *is primitive recursive.*

inc:art:trm: prop:num-primrec

Proof. We define num(n) by primitive recursion:

$$\begin{aligned}\mathrm{num}(0) &= {}^{\ulcorner}0{}^{\urcorner}\\ \mathrm{num}(n+1) &= {}^{\ulcorner}\prime({}^{\urcorner} \frown \mathrm{num}(n) \frown {}^{\ulcorner})^{\urcorner}.\end{aligned}$$

□

---

content/incompleteness/arithmetization-syntax/coding-formulas.tex

34.4 Coding Formulas

Once we have defined the relation Term(x) primitive recursively, we can use it to define the corresponding relation for formulas, Frm(x) primitive recursively.

inc:art:frm: sec

Proposition 34.7. *The relation* Atom(x) *which holds iff x is the Gödel number of an atomic formula, is primitive recursive.*

Proof. The number x is the Gödel number of an atomic formula iff one of the following holds:

1. There are n, $j < x$, and $z < x$ such that for each $i < n$, Term($(z)_i$) and $x =$
$$^{\ulcorner}P^n_j({}^{\urcorner} \frown \mathrm{flatten}(z) \frown {}^{\ulcorner})^{\urcorner}.$$
2. There are $z_1, z_2 < x$ such that Term(z_1), Term(z_2), and $x =$
$$^{\ulcorner}=({}^{\urcorner} \frown z_1 \frown {}^{\ulcorner},{}^{\urcorner} \frown z_2 \frown {}^{\ulcorner})^{\urcorner}.$$
3. $x = {}^{\ulcorner}\bot{}^{\urcorner}$.
4. $x = {}^{\ulcorner}\top{}^{\urcorner}$.

□

Proposition 34.8. *The relation* Frm(x) *which holds iff x is the Gödel number of a formula is primitive recursive.*

inc:art:frm: prop:frm-primrec

Proof. A sequence of symbols s is a formula iff there is formation sequence s_0, $\ldots$, $s_{k-1} = s$ of formula which records how s was formed from atomic formulas according to the formation rules. The code for each s_i (and indeed of the code of the sequence $\langle s_0, \ldots, s_{k-1}\rangle$) is less than the code x of s. □

Problem 34.2. Give a detailed proof of Proposition 34.8 along the lines of the first proof of Proposition 34.5.

inc:art:frm: prop:freeocc-primrec **Proposition 34.9.** *The relation* $\mathrm{FreeOcc}(x, z, i)$*, which holds iff the* i*-th symbol of the formula with Gödel number* x *is a free occurrence of the variable with Gödel number* z*, is primitive recursive.*

Proof. Exercise. □

Problem 34.3. Prove Proposition 34.9. You may make use of the fact that any substring of a formula which is a formula is a sub-formula of it.

Proposition 34.10. *The property* $\mathrm{Sent}(x)$ *which holds iff* x *is the Gödel number of a sentence is primitive recursive.*

Proof. A sentence is a formula without free occurrences of variables. So $\mathrm{Sent}(x)$ holds iff

$$(\forall i < \mathrm{len}(x))\ (\forall z < x)\ ((\exists j < z)\ z = \ulcorner v_j \urcorner \rightarrow \neg \mathrm{FreeOcc}(x, z, i)).$$ □

---

content/incompleteness/arithmetization-syntax/substitution.tex

34.5 Substitution

inc:art:sub: sec Recall that substitution is the operation of replacing all free occurrences of a variable u in a formula φ by a term t, written $\varphi[t/u]$. This operation, when carried out on Gödel numbers of variables, formulas, and terms, is primitive recursive.

inc:art:sub: prop:subst-primrec **Proposition 34.11.** *There is a primitive recursive function* $\mathrm{Subst}(x, y, z)$ *with the property that*

$$\mathrm{Subst}(\ulcorner \varphi \urcorner, \ulcorner t \urcorner, \ulcorner u \urcorner) = \ulcorner \varphi[t/u] \urcorner.$$

Proof. We can then define a function hSubst by primitive recursion as follows:

$$\begin{aligned} \mathrm{hSubst}(x, y, z, 0) &= \Lambda \\ \mathrm{hSubst}(x, y, z, i+1) &= \\ & \begin{cases} \mathrm{hSubst}(x, y, z, i) \frown y & \text{if } \mathrm{FreeOcc}(x, z, i) \\ \mathrm{append}(\mathrm{hSubst}(x, y, z, i), (x)_i) & \text{otherwise.} \end{cases} \end{aligned}$$

$\mathrm{Subst}(x, y, z)$ can now be defined as $\mathrm{hSubst}(x, y, z, \mathrm{len}(x))$. □

inc:art:sub: prop:free-for **Proposition 34.12.** *The relation* $\mathrm{FreeFor}(x, y, z)$*, which holds iff the term with Gödel number* y *is free for the variable with Gödel number* z *in the formula with Gödel number* x*, is primitive recursive.*

Proof. Exercise. □

Problem 34.4. Prove Proposition 34.12

---

content/incompleteness/arithmetization-syntax/proofs-in-lk.tex

34.6 Derivations in LK

explanation

In order to arithmetize derivations, we must represent derivations as numbers. Since derivations are trees of sequents where each inference carries also a label, a recursive representation is the most obvious approach: we represent a derivation as a tuple, the components of which are the end-sequent, the label, and the representations of the sub-derivations leading to the premises of the last inference.

inc:art:plk:sec

Definition 34.13. If Γ is a finite sequence of sentences, $\Gamma = \langle \varphi_1, \ldots, \varphi_n \rangle$, then ${}^{\ulcorner}\Gamma^{\urcorner} = \langle {}^{\ulcorner}\varphi_1{}^{\urcorner}, \ldots, {}^{\ulcorner}\varphi_n{}^{\urcorner} \rangle$.

If $\Gamma \Rightarrow \Delta$ is a sequent, then a Gödel number of $\Gamma \Rightarrow \Delta$ is

$$ {}^{\ulcorner}\Gamma \Rightarrow \Delta^{\urcorner} = \langle {}^{\ulcorner}\Gamma^{\urcorner}, {}^{\ulcorner}\Delta^{\urcorner} \rangle $$

If π is a derivation in **LK**, then ${}^{\ulcorner}\pi^{\urcorner}$ is defined as follows:

1. If π consists only of the initial sequent $\Gamma \Rightarrow \Delta$, then ${}^{\ulcorner}\pi^{\urcorner}$ is

   $$ \langle 0, {}^{\ulcorner}\Gamma \Rightarrow \Delta^{\urcorner} \rangle. $$

2. If π ends in an inference with one or two premises, has $\Gamma \Rightarrow \Delta$ as its conclusion, and π_1 and π_2 are the immediate subproof ending in the premise of the last inference, then ${}^{\ulcorner}\pi^{\urcorner}$ is

   $$ \langle 1, {}^{\ulcorner}\pi_1{}^{\urcorner}, {}^{\ulcorner}\Gamma \Rightarrow \Delta^{\urcorner}, k \rangle \text{ or} $$
   $$ \langle 2, {}^{\ulcorner}\pi_1{}^{\urcorner}, {}^{\ulcorner}\pi_2{}^{\urcorner}, {}^{\ulcorner}\Gamma \Rightarrow \Delta^{\urcorner}, k \rangle, $$

   respectively, where k is given by the following table according to which rule was used in the last inference:

Rule:	WL	WR	CL	CR	XL	XR
k:	1	2	3	4	5	6

Rule:	¬L	¬R	∧L	∧R	∨L	∨R
k:	7	8	9	10	11	12

Rule:	→L	→R	∀L	∀R	∃L	∃R
k:	13	14	15	16	17	18

Rule:	Cut	=
k:	19	20

Example 34.14. Consider the very simple derivation

$$\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi}\,\wedge\text{L}}{\Rightarrow (\varphi \wedge \psi) \to \varphi}\,\to\text{R}$$

The Gödel number of the initial sequent would be $p_0 = \langle 0, {}^{\ulcorner}\varphi \Rightarrow \varphi^{\#}\rangle$. The Gödel number of the derivation ending in the conclusion of $\wedge$L would be $p_1 = \langle 1, p_0, {}^{\ulcorner}\varphi \wedge \psi \Rightarrow \varphi^{\#}, 9\rangle$ (1 since $\wedge$L has one premise, the Gödel number of the conclusion $\varphi \wedge \psi \Rightarrow \varphi$, and 9 is the number coding $\wedge$L). The Gödel number of the entire derivation then is $\langle 1, p_1, {}^{\ulcorner}\Rightarrow (\varphi \wedge \psi) \to \varphi)^{\#}, 14\rangle$, i.e.,

$$\langle 1, \langle 1, \langle 0, {}^{\ulcorner}\varphi \Rightarrow \varphi)^{\#}\rangle, {}^{\ulcorner}\varphi \wedge \psi \Rightarrow \varphi^{\#}, 9\rangle, {}^{\ulcorner}\Rightarrow (\varphi \wedge \psi) \to \varphi^{\#}, 14\rangle.$$

Having settled on a representation of derivations, we must also show that we can manipulate such derivations primitive recursively, and express their essential properties and relations so. Some operations are simple: e.g., given a Gödel number p of a derivation, $\text{EndSeq}(p) = (p)_{(p)_0+1}$ gives us the Gödel number of its end-sequent and $\text{LastRule}(p) = (p)_{(p)_0+2}$ the code of its last rule. The property $\text{Sequent}(s)$ defined by explanation

$$\text{len}(s) = 2 \wedge (\forall i < \text{len}((s)_0) + \text{len}((s)_1))\ \text{Sent}(((s)_0 \frown (s)_1)_i)$$

holds of s iff s is the Gödel number of a sequent consisting of sentences. Some are much harder. We'll at least sketch how to do this. The goal is to show that the relation "π is a derivation of φ from Γ" is a primitive recursive relation of the Gödel numbers of π and φ.

inc:art:plk: prop:followsby **Proposition 34.15.** *The property* $\text{Correct}(p)$ *which holds iff the last inference in the derivation* π *with Gödel number* p *is correct, is primitive recursive.*

Proof. $\Gamma \Rightarrow \Delta$ is an initial sequent if either there is a sentence φ such that $\Gamma \Rightarrow \Delta$ is $\varphi \Rightarrow \varphi$, or there is a term t such that $\Gamma \Rightarrow \Delta$ is $\emptyset \Rightarrow t = t$. In terms of Gödel numbers, $\text{InitSeq}(s)$ holds iff

$$\begin{aligned}&(\exists x < s)\ (\text{Sent}(x) \wedge s = \langle\langle x\rangle, \langle x\rangle\rangle) \vee\\ &(\exists t < s)\ (\text{Term}(t) \wedge s = \langle 0, \langle {}^{\ulcorner}{=}({}^{\#} \frown t \frown {}^{\ulcorner}, {}^{\#} \frown t \frown {}^{\ulcorner})^{\#}\rangle\rangle).\end{aligned}$$

We also have to show that for each rule of inference R the relation $\text{FollowsBy}_R(p)$ is primitive recursive, where $\text{FollowsBy}_R(p)$ holds iff p is the Gödel number of derivation π, and the end-sequent of π follows by a correct application of R from the immediate sub-derivations of π.

A simple case is that of the $\wedge$R rule. If π ends in a correct $\wedge$R inference, it looks like this:

$$\dfrac{\begin{array}{cc}\vdots\,\pi_1 & \vdots\,\pi_2\\ \Gamma \Rightarrow \Delta, \varphi & \Gamma \Rightarrow \Delta, \psi\end{array}}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\,\wedge\text{R}$$

So, the last inference in the derivation π is a correct application of $\wedge$R iff there are sequences of sentences Γ and Δ as well as two sentences φ and ψ such that the end-sequent of π_1 is $\Gamma \Rightarrow \Delta, \varphi$, the end-sequent of π_2 is $\Gamma \Rightarrow \Delta, \psi$, and the end-sequent of π is $\Gamma \Rightarrow \Delta, \varphi \wedge \psi$. We just have to translate this into Gödel numbers. If $s = {}^{\ddagger}\Gamma \Rightarrow \Delta^{\#}$ then $(s)_0 = {}^{\ddagger}\Gamma^{\#}$ and $(s)_1 = {}^{\ddagger}\Delta^{\#}$. So, $\mathrm{FollowsBy}_{\wedge\mathrm{R}}(p)$ holds iff

$$\begin{aligned}
&(\exists g < p)\ (\exists d < p)\ (\exists a < p)\ (\exists b < p)\\
&\quad \mathrm{EndSequent}(p) = \langle g, d \frown \langle {}^{\ddagger}(^{\#} \frown a \frown {}^{\ddagger}\wedge^{\#} \frown b \frown {}^{\ddagger})^{\#}\rangle\rangle \wedge\\
&\quad \mathrm{EndSequent}((p)_1) = \langle g, d \frown \langle a\rangle\rangle \wedge\\
&\quad \mathrm{EndSequent}((p)_2) = \langle g, d \frown \langle b\rangle\rangle \wedge\\
&\quad (p)_0 = 2 \wedge \mathrm{LastRule}(p) = 10.
\end{aligned}$$

The individual lines express, respectively, "there is a sequence (Γ) with Gödel number g, there is a sequence (Δ) with Gödel number d, a formula (φ) with Gödel number a, and a formula (ψ) with Gödel number b," such that "the end-sequent of π is $\Gamma \Rightarrow \Delta, \varphi \wedge \psi$," "the end-sequent of π_1 is $\Gamma \Rightarrow \Delta, \varphi$," "the end-sequent of π_2 is $\Gamma \Rightarrow \Delta, \psi$," and "$\pi$ has two immediate subderivations and the last inference rule is $\wedge$R (with number 10)."

The last inference in π is a correct application of $\exists$R iff there are sequences Γ and Δ, a formula φ, a variable x, and a term t, such that the end-sequent of π is $\Gamma \Rightarrow \Delta, \exists x\, \varphi$ and the end-sequent of π_1 is $\Gamma \Rightarrow \Delta, \varphi[t/x]$. So in terms of Gödel numbers, we have $\mathrm{FollowsBy}_{\exists\mathrm{R}}(p)$ iff

$$\begin{aligned}
&(\exists g < p)\ (\exists d < p)\ (\exists a < p)\ (\exists x < p)\ (\exists t < p)\\
&\quad \mathrm{EndSequent}(p) = \langle g, d \frown \langle {}^{\ddagger}\exists^{\#} \frown x \frown a\rangle\rangle \wedge\\
&\quad \mathrm{EndSequent}((p)_1) = \langle g, d \frown \langle \mathrm{Subst}(a, t, x)\rangle\rangle \wedge\\
&\quad (p)_0 = 1 \wedge \mathrm{LastRule}(p) = 18.
\end{aligned}$$

We then define $\mathrm{Correct}(p)$ as

$$\begin{aligned}
&\mathrm{Sequent}(\mathrm{EndSequent}(p)) \wedge\\
&\qquad [(\mathrm{LastRule}(p) = 1 \wedge \mathrm{FollowsBy}_{\mathrm{WL}}(p)) \vee \cdots \vee\\
&\qquad\quad (\mathrm{LastRule}(p) = 20 \wedge \mathrm{FollowsBy}_{=}(p)) \vee\\
&\qquad\qquad (p)_0 = 0 \wedge \mathrm{InitialSeq}(\mathrm{EndSequent}(p))]
\end{aligned}$$

The first line ensures that the end-sequent of d is actually a sequent consisting of sentences. The last line covers the case where p is just an initial sequent. □

Problem 34.5. Define the following properties as in Proposition 34.15:

1. $\mathrm{FollowsBy}_{\mathrm{Cut}}(p)$,
2. $\mathrm{FollowsBy}_{\to\mathrm{L}}(p)$,
3. $\mathrm{FollowsBy}_{=}(p)$,

4. $\text{FollowsBy}_{\forall \text{R}}(p)$.

For the last one, you will have to also show that you can test primitive recursively if the last inference of the derivation with Gödel number p satisfies the eigenvariable condition, i.e., the eigenvariable a of the $\forall$R does not occur in the end-sequent.

inc:art:plk: prop:deriv

Proposition 34.16. *The relation* $\text{Deriv}(p)$ *which holds if* p *is the Gödel number of a correct derivation* π*, is primitive recursive.*

Proof. A derivation π is correct if every one of its inferences is a correct application of a rule, i.e., if every one of its sub-derivations ends in a correct inference. So, $\text{Deriv}(d)$ iff

$$(\forall i < \text{len}(\text{SubtreeSeq}(p)))\ \text{Correct}((\text{SubtreeSeq}(p))_i.$$ □

Proposition 34.17. *Suppose* Γ *is a primitive recursive set of sentences. Then the relation* $\text{Prf}_\Gamma(x, y)$ *expressing "*x *is the code of a derivation* π *of* $\Gamma_0 \Rightarrow \varphi$ *for some finite* $\Gamma_0 \subseteq \Gamma$ *and* y *is the Gödel number of* φ*" is primitive recursive.*

Proof. Suppose "$y \in \Gamma$" is given by the primitive recursive predicate $R_\Gamma(y)$. We have to show that $\text{Prf}_\Gamma(x, y)$ which holds iff y is the Gödel number of a sentence φ and x is the code of an **LK**-derivation with end-sequent $\Gamma_0 \Rightarrow \varphi$ is primitive recursive.

By the previous proposition, the property $\text{Deriv}(x)$ which holds iff x is the code of a correct derivation π in **LK** is primitive recursive. If x is such a code, then $\text{EndSequent}(x)$ is the code of the end-sequent of π, and so $(\text{EndSequent}(x))_0$ is the code of the left side of the end sequent and $(\text{EndSequent}(x))_1$ the right side. So we can express "the right side of the end-sequent of π is φ" as $\text{len}((\text{EndSequent}(x))_1) = 1 \wedge ((\text{EndSequent}(x))_1)_0 = x$. The left side of the end-sequent of π is of course automatically finite, we just have to express that every sentence in it is in Γ. Thus we can define $\text{Prf}_\Gamma(x, y)$ by

$$\begin{aligned}\text{Prf}_\Gamma(x, y) \Leftrightarrow\ & \text{Deriv}(x) \wedge \\ & (\forall i < \text{len}((\text{EndSequent}(x))_0))\ R_\Gamma(((\text{EndSequent}(x))_0)_i) \wedge \\ & \text{len}((\text{EndSequent}(x))_1) = 1 \wedge ((\text{EndSequent}(x))_1)_0 = y.\end{aligned}$$ □

---

content/incompleteness/arithmetization-syntax/proofs-in-nd.tex

34.7 Derivations in Natural Deduction

inc:art:pnd: sec

In order to arithmetize derivations, we must represent derivations as numbers. Since derivations are trees of formulas where each inference carries one or two labels, a recursive representation is the most obvious approach: we represent a derivation as a tuple, the components of which are the number of immediate

explanation

sub-derivations leading to the premises of the last inference, the representations of these sub-derivations, and the end-formula, the discharge label of the last inference, and a number indicating the type of the last inference.

Definition 34.18. If δ is a derivation in natural deduction, then ${}^{\ulcorner}\delta^{\urcorner}$ is defined inductively as follows:

1. If δ consists only of the assumption φ, then ${}^{\ulcorner}\delta^{\urcorner}$ is $\langle 0, {}^{\ulcorner}\varphi^{\urcorner}, n\rangle$. The number n is 0 if it is an undischarged assumption, and the numerical label otherwise.

2. If δ ends in an inference with zero, one, two, or three premises, then ${}^{\ulcorner}\delta^{\urcorner}$ is

$$\begin{aligned}&\langle 0, {}^{\ulcorner}\varphi^{\urcorner}, n, k\rangle,\\ &\langle 1, {}^{\ulcorner}\delta_1{}^{\urcorner}, {}^{\ulcorner}\varphi^{\urcorner}, n, k\rangle,\\ &\langle 2, {}^{\ulcorner}\delta_1{}^{\urcorner}, {}^{\ulcorner}\delta_2{}^{\urcorner}, {}^{\ulcorner}\varphi^{\urcorner}, n, k\rangle, \text{ or}\\ &\langle 3, {}^{\ulcorner}\delta_1{}^{\urcorner}, {}^{\ulcorner}\delta_2{}^{\urcorner}, {}^{\ulcorner}\delta_3{}^{\urcorner}, {}^{\ulcorner}\varphi^{\urcorner}, n, k\rangle,\end{aligned}$$

   respectively. Here $\delta_1, \delta_2, \delta_3$ are the sub-derivations ending in the premise(s) of the last inference in δ, φ is the conclusion of the last inference in δ, n is the discharge label of the last inference (0 if the inference does not discharge any assumptions), and k is given by the following table according to which rule was used in the last inference.

Rule:	∧Intro	∧Elim	∨Intro	∨Elim
k:	1	2	3	4

Rule:	→Intro	→Elim	¬Intro	¬Elim
k:	5	6	7	8

Rule:	$\bot_I$	$\bot_C$	∀Intro	∀Elim
k:	9	10	11	12

Rule:	∃Intro	∃Elim	=Intro	=Elim
k:	13	14	15	16

Example 34.19. Consider the very simple derivation

$$\dfrac{\dfrac{\dfrac{[\varphi \wedge \psi]^1}{\varphi}\wedge\text{Elim}}{(\varphi \wedge \psi) \to \varphi}\to\text{Intro}}{}\,{}^{1}$$

The Gödel number of the assumption would be $d_0 = \langle 0, {}^{\ulcorner}\varphi \wedge \psi^{\urcorner}, 1\rangle$. The Gödel number of the derivation ending in the conclusion of ∧Elim would be $d_1 = \langle 1, d_0, {}^{\ulcorner}\varphi^{\urcorner}, 0, 2\rangle$ (1 since ∧Elim has one premise, the Gödel number of conclusion φ, 0 because no assumption is discharged, and 2 is the number coding ∧Elim). The Gödel number of the entire derivation then is $\langle 1, d_1, {}^{\ulcorner}((\varphi \wedge \psi) \to \varphi)^{\urcorner}, 1, 5\rangle$, i.e.,

$$\langle 1, \langle 1, \langle 0, {}^{\ulcorner}(\varphi \wedge \psi)^{\urcorner}, 1\rangle, {}^{\ulcorner}\varphi^{\urcorner}, 0, 2\rangle, {}^{\ulcorner}((\varphi \wedge \psi) \to \varphi)^{\urcorner}, 1, 5\rangle.$$

explanation

Having settled on a representation of derivations, we must also show that we can manipulate Gödel numbers of such derivations primitive recursively, and express their essential properties and relations. Some operations are simple: e.g., given a Gödel number d of a derivation, $\text{EndFmla}(d) = (d)_{(d)_0+1}$ gives us the Gödel number of its end-formula, $\text{DischargeLabel}(d) = (d)_{(d)_0+2}$ gives us the discharge label and $\text{LastRule}(d) = (d)_{(d)_0+3}$ the number indicating the type of the last inference. Some are much harder. We'll at least sketch how to do this. The goal is to show that the relation "δ is a derivation of φ from Γ" is a primitive recursive relation of the Gödel numbers of δ and φ.

Proposition 34.20. *The following relations are primitive recursive:*

1. *φ occurs as an assumption in δ with label n.*

2. *All assumptions in δ with label n are of the form φ (i.e., we can discharge the assumption φ using label n in δ).*

Proof. We have to show that the corresponding relations between Gödel numbers of formulas and Gödel numbers of derivations are primitive recursive.

1. We want to show that $\text{Assum}(x, d, n)$, which holds if x is the Gödel number of an assumption of the derivation with Gödel number d labelled n, is primitive recursive. This is the case if the derivation with Gödel number $\langle 0, x, n\rangle$ is a sub-derivation of d. Note that the way we code derivations is a special case of the coding of trees introduced in section 29.12, so the primitive recursive function $\text{SubtreeSeq}(d)$ gives a sequence of Gödel numbers of all sub-derivations of d (of length at most d). So we can define

$$\text{Assum}(x, d, n) \Leftrightarrow (\exists i < d)\ (\text{SubtreeSeq}(d))_i = \langle 0, x, n\rangle.$$

2. We want to show that $\text{Discharge}(x, d, n)$, which holds if all assumptions with label n in the derivation with Gödel number d all are the formula with Gödel number x. But this relation holds iff $(\forall y < d)\ (\text{Assum}(y, d, n) \rightarrow y = x)$. □

inc:art:pnd: prop:followsby

Proposition 34.21. *The property* $\text{Correct}(d)$ *which holds iff the last inference in the derivation δ with Gödel number d is correct, is primitive recursive.*

Proof. Here we have to show that for each rule of inference R the relation $\text{FollowsBy}_R(d)$ is primitive recursive, where $\text{FollowsBy}_R(d)$ holds iff d is the Gödel number of derivation δ, and the end-formula of δ follows by a correct application of R from the immediate sub-derivations of δ.

A simple case is that of the ∧Intro rule. If δ ends in a correct ∧Intro inference, it looks like this:

$$\dfrac{\begin{array}{cc}\vdots & \vdots\\ \delta_1 & \delta_2\\ \varphi & \psi\end{array}}{\varphi \wedge \psi}\ \wedge\text{Intro}$$

Then the Gödel number d of δ is $\langle 2, d_1, d_2, {}^{\ulcorner}(\varphi \wedge \psi)^{\urcorner}, 0, k\rangle$ where $\text{EndFmla}(d_1) = {}^{\ulcorner}\varphi^{\urcorner}$, $\text{EndFmla}(d_2) = {}^{\ulcorner}\psi^{\urcorner}$, $n = 0$, and $k = 1$. So we can define $\text{FollowsBy}_{\wedge\text{Intro}}(d)$ as

$$\begin{aligned}(d)_0 = 2 \wedge \text{DischargeLabel}(d) = 0 \wedge \text{LastRule}(d) = 1 \wedge \\ \text{EndFmla}(d) = \\ {}^{\ulcorner}({}^{\urcorner} \frown \text{EndFmla}((d)_1) \frown {}^{\ulcorner}\wedge{}^{\urcorner} \frown \text{EndFmla}((d)_2) \frown {}^{\ulcorner})^{\urcorner}.\end{aligned}$$

Another simple example is the =Intro rule. This has no premises, so $(d)_0 = 0$, like assumptions. It also has no discharge label, i.e., $n = 0$. However, φ must be of the form $t = t$, for a closed term t. Here, a primitive recursive definition is

$$\begin{aligned}(d)_0 = 0 \wedge \text{DischargeLabel}(d) = 0 \wedge \\ (\exists t < d)\ (\text{ClTerm}(t) \wedge \text{EndFmla}(d) = \\ {}^{\ulcorner}={}^{\urcorner}({}^{\urcorner} \frown t \frown {}^{\ulcorner},{}^{\urcorner} \frown t \frown {}^{\ulcorner})^{\urcorner}).\end{aligned}$$

For a more complicated example, $\text{FollowsBy}_{\to\text{Intro}}(d)$ holds iff the end-formula of δ is of the form $(\varphi \to \psi)$, where the end-formula of δ_1 is ψ, and any assumption in δ labelled n is of the form φ. We can express this primitive recursively by

$$\begin{aligned}(d)_0 = 1 \wedge \\ (\exists a < d)\ (\text{Discharge}(a, (d)_1, \text{DischargeLabel}(d)) \wedge \\ \text{EndFmla}(d) = ({}^{\ulcorner}({}^{\urcorner} \frown a \frown {}^{\ulcorner}\to{}^{\urcorner} \frown \text{EndFmla}((d)_1) \frown {}^{\ulcorner})^{\urcorner}))\end{aligned}$$

(Think of a as the Gödel number of φ).

For another example, consider ∃Intro. Here, the last inference in δ is correct iff there is a formula φ, a closed term t and a variable x such that $\varphi[t/x]$ is the end-formula of the derivation δ_1 and $\exists x\, \varphi$ is the conclusion of the last inference. So, $\text{FollowsBy}_{\exists\text{Intro}}(d)$ holds iff

$$\begin{aligned}(d)_0 = 1 \wedge \text{DischargeLabel}(d) = 0 \wedge \\ (\exists a < d)\ (\exists x < d)\ (\exists t < d)\ (\text{ClTerm}(t) \wedge \text{Var}(x) \wedge \\ \text{Subst}(a, t, x) = \text{EndFmla}((d)_1) \wedge \text{EndFmla}(d) = ({}^{\ulcorner}\exists{}^{\urcorner} \frown x \frown a)).\end{aligned}$$

We then define $\mathrm{Correct}(d)$ as

$$\begin{aligned}&\mathrm{Sent}(\mathrm{EndFmla}(d)) \wedge \\ &\quad (\mathrm{LastRule}(d) = 1 \wedge \mathrm{FollowsBy}_{\wedge\mathrm{Intro}}(d)) \vee \cdots \vee \\ &\quad (\mathrm{LastRule}(d) = 16 \wedge \mathrm{FollowsBy}_{=\mathrm{Elim}}(d)) \vee \\ &\quad (\exists n < d)\ (\exists x < d)\ (d = \langle 0, x, n\rangle).\end{aligned}$$

The first line ensures that the end-formula of d is a sentence. The last line covers the case where d is just an assumption. □

Problem 34.6. Define the following properties as in Proposition 34.21:

1. $\mathrm{FollowsBy}_{\to\mathrm{Elim}}(d)$,
2. $\mathrm{FollowsBy}_{=\mathrm{Elim}}(d)$,
3. $\mathrm{FollowsBy}_{\vee\mathrm{Elim}}(d)$,
4. $\mathrm{FollowsBy}_{\forall\mathrm{Intro}}(d)$.

For the last one, you will have to also show that you can test primitive recursively if the last inference of the derivation with Gödel number d satisfies the eigenvariable condition, i.e., the eigenvariable a of the ∀Intro inference occurs neither in the end-formula of d nor in an open assumption of d. You may use the primitive recursive predicate OpenAssum from Proposition 34.23 for this.

inc:art:pnd: prop:deriv

Proposition 34.22. *The relation* $\mathrm{Deriv}(d)$ *which holds if d is the Gödel number of a correct derivation δ, is primitive recursive.*

Proof. A derivation δ is correct if every one of its inferences is a correct application of a rule, i.e., if every one of its sub-derivations ends in a correct inference. So, $\mathrm{Deriv}(d)$ iff

$$(\forall i < \mathrm{len}(\mathrm{SubtreeSeq}(d)))\ \mathrm{Correct}((\mathrm{SubtreeSeq}(d))_i)$$ □

inc:art:pnd: prop:openassum

Proposition 34.23. *The relation* $\mathrm{OpenAssum}(z, d)$ *that holds if z is the Gödel number of an undischarged assumption φ of the derivation δ with Gödel number d, is primitive recursive.*

Proof. An occurrence of an assumption is discharged if it occurs with label n in a sub-derivation of δ that ends in a rule with discharge label n. So φ is an undischarged assumption of δ if at least one of its occurrences is not discharged in δ. We must be careful: δ may contain both discharged and undischarged occurrences of φ.

Consider a sequence $\delta_0, \ldots, \delta_k$ where $\delta_0 = \delta$, δ_k is the assumption $[\varphi]^n$ (for some n), and δ_{i+1} is an immediate sub-derivation of δ_i. If such a sequence exists in which no δ_i ends in an inference with discharge label n, then φ is an undischarged assumption of δ.

The primitive recursive function $\text{SubtreeSeq}(d)$ provides us with a sequence of Gödel numbers of all sub-derivations of δ. Any sequence of Gödel numbers of sub-derivations of δ is a subsequence of it. Being a subsequence of is a primitive recursive relation: $\text{Subseq}(s, s')$ holds iff $(\forall i < \text{len}(s))\ \exists j < \text{len}(s')\, (s)_i = (s')_j$. Being an immediate sub-derivation is as well: $\text{Subderiv}(d, d')$ iff $(\exists j < (d')_0)\ d = (d')_j$. So we can define $\text{OpenAssum}(z, d)$ by

$$\begin{aligned}(\exists s < \text{SubtreeSeq}(d))\ (\text{Subseq}(s, \text{SubtreeSeq}(d)) \wedge (s)_0 = d \wedge \\ (\exists n < d)\ ((s)_{\text{len}(s) \dot{-} 1} = \langle 0, z, n\rangle \wedge \\ (\forall i < (\text{len}(s) \dot{-} 1))\ (\text{Subderiv}((s)_{i+1}, (s)_i) \wedge \\ \text{DischargeLabel}((s)_i) \neq n))).\end{aligned}$$ □

Proposition 34.24. *Suppose Γ is a primitive recursive set of sentences. Then the relation $\text{Prf}_\Gamma(x, y)$ expressing "x is the code of a derivation δ of φ from undischarged assumptions in Γ and y is the Gödel number of φ" is primitive recursive.* inc:art:pnd: prop:prf-prim-rec

Proof. Suppose "$y \in \Gamma$" is given by the primitive recursive predicate $R_\Gamma(y)$. We have to show that $\text{Prf}_\Gamma(x, y)$ which holds iff y is the Gödel number of a sentence φ and x is the code of a natural deduction derivation with end formula φ and all undischarged assumptions in Γ is primitive recursive.

By Proposition 34.22, the property $\text{Deriv}(x)$ which holds iff x is the Gödel number of a correct derivation δ in natural deduction is primitive recursive. Thus we can define $\text{Prf}_\Gamma(x, y)$ by

$$\begin{aligned}\text{Prf}_\Gamma(x, y) \Leftrightarrow \text{Deriv}(x) \wedge \text{EndFmla}(x) = y \wedge \\ (\forall z < x)\ (\text{OpenAssum}(z, x) \rightarrow R_\Gamma(z)).\end{aligned}$$ □

---

content/incompleteness/arithmetization-syntax/proofs-in-ax.tex

34.8 Axiomatic Derivations

explanation In order to arithmetize axiomatic derivations, we must represent derivations as numbers. Since derivations are simply sequences of formulas, the obvious approach is to code every derivation as the code of the sequence of codes of formulas in it. inc:art:pax: sec

Definition 34.25. If δ is an axiomatic derivation consisting of formulas $\varphi_1, \ldots, \varphi_n$, then ${}^{\ulcorner}\delta^{\urcorner}$ is

$$\langle {}^{\ulcorner}\varphi_1{}^{\urcorner}, \ldots, {}^{\ulcorner}\varphi_n{}^{\urcorner}\rangle.$$

Example 34.26. Consider the very simple derivation:

1. $\psi \to (\psi \vee \varphi)$
2. $(\psi \to (\psi \vee \varphi)) \to (\varphi \to (\psi \to (\psi \vee \varphi)))$
3. $\varphi \to (\psi \to (\psi \vee \varphi))$

The Gödel number of this derivation would be

$$\langle {}^{\ulcorner}\psi \to (\psi \vee \varphi)^{\urcorner}, \\ {}^{\ulcorner}(\psi \to (\psi \vee \varphi)) \to (\varphi \to (\psi \to (\psi \vee \varphi)))^{\urcorner}, \\ {}^{\ulcorner}\varphi \to (\psi \to (\psi \vee \varphi))^{\urcorner} \rangle.$$

explanation

Having settled on a representation of derivations, we must also show that we can manipulate such derivations primitive recursively, and express their essential properties and relations so. Some operations are simple: e.g., given a Gödel number d of a derivation, $(d)_{\mathrm{len}(d)-1}$ gives us the Gödel number of its end-formula. Some are much harder. We'll at least sketch how to do this. The goal is to show that the relation "δ is a derivation of φ from Γ" is primitive recursive in the Gödel numbers of δ and φ.

inc:art:pax: prop:followsby

Proposition 34.27. *The following relations are primitive recursive:*

1. *φ is an axiom.*
2. *The i-th line in δ is justified by modus ponens*
3. *The i-th line in δ is justified by* QR.
4. *δ is a correct derivation.*

Proof. We have to show that the corresponding relations between Gödel numbers of formulas and Gödel numbers of derivations are primitive recursive.

1. We have a given list of axiom schemas, and φ is an axiom if it is of the form given by one of these schemas. Since the list of schemas is finite, it suffices to show that we can test primitive recursively, for each axiom schema, if φ is of that form. For instance, consider the axiom schema

   $$\psi \to (\chi \to \psi).$$

   φ is an instance of this axiom schema if there are formulas ψ and χ such that we obtain φ when we concatenate '(' with ψ with '$\to$' with '(' with χ with '$\to$' with ψ and with '))'. We can test the corresponding property of the Gödel number n of φ, since concatenation of sequences is primitive recursive and the Gödel numbers of ψ and χ must be smaller than the Gödel number of φ, since when the relation holds, both ψ and χ are sub-formulas of φ. Hence, we can define:

   $$\mathrm{IsAx}_{\psi\to(\chi\to\psi)}(n) \Leftrightarrow (\exists b < n)\ (\exists c < n)\ (\mathrm{Sent}(b) \wedge \mathrm{Sent}(c) \wedge \\ n = {}^{\ulcorner}(^{\ulcorner} \frown b \frown {}^{\ulcorner}\to^{\urcorner} \frown {}^{\ulcorner}(^{\ulcorner} \frown c \frown {}^{\ulcorner}\to^{\urcorner} \frown b \frown {}^{\ulcorner}))^{\urcorner}).$$

   If we have such a definition for each axiom schema, their disjunction defines the property $\mathrm{IsAx}(n)$, "n is the Gödel number of an axiom."

2. The i-th line in δ is justified by modus ponens iff there are lines j and $k < i$ where the sentence on line j is some formula φ, the sentence on line k is $\varphi \to \psi$, and the sentence on line i is ψ.

$$\mathrm{MP}(d, i) \Leftrightarrow (\exists j < i)\ (\exists k < i)\ (d)_k = {}^{\#}(^{\#} \frown (d)_j \frown {}^{\#}{\to}^{\#} \frown (d)_i \frown {}^{\#})^{\#}$$

Since bounded quantification, concatenation, and $=$ are primitive recursive, this defines a primitive recursive relation.

3. A line in δ is justified by QR if it is of the form $\psi \to \forall x\, \varphi(x)$, a preceding line is $\psi \to \varphi(c)$ for some constant symbol c, and c does on occur in ψ. This is the case iff

   a) there is a sentence ψ and
   b) a formula $\varphi(x)$ with a single variable x free so that
   c) line i contains $\psi \to \forall x\, \varphi(x)$
   d) some line $j < i$ contains $\psi \to \varphi[c/x]$ for a constant c
   e) which does not occur in ψ.

   All of these can be tested primitive recursively, since the Gödel numbers of ψ, $\varphi(x)$, and x are less than the Gödel number of the formula on line i, and that of a less than the Gödel number of the formula on line j:

$$\begin{aligned}\mathrm{QR}_1(d, i) \Leftrightarrow (\exists b < (d)_i)\ (\exists x < (d)_i)\ (\exists a < (d)_i)\ (\exists c < (d)_j)\ (&\\ \mathrm{Var}(x) \wedge \mathrm{Const}(c) \wedge&\\ (d)_i = {}^{\#}(^{\#} \frown b \frown {}^{\#}{\to}^{\#} \frown {}^{\#}{\forall}^{\#} \frown x \frown a \frown {}^{\#})^{\#} \wedge&\\ (d)_j = {}^{\#}(^{\#} \frown b \frown {}^{\#}{\to}^{\#} \frown \mathrm{Subst}(a, c, x) \frown {}^{\#})^{\#} \wedge&\\ \mathrm{Sent}(b) \wedge \mathrm{Sent}(\mathrm{Subst}(a, c, x)) \wedge (\forall k < \mathrm{len}(b))\ (b)_k \neq (c)_0)&\end{aligned}$$

   Here we assume that c and x are the Gödel numbers of the variable and constant considered as terms (i.e., not their symbol codes). We test that x is the only free variable of $\varphi(x)$ by testing if $\varphi(x)[c/x]$ is a sentence, and ensure that c does not occur in ψ by requiring that every symbol of ψ is different from c.

   We leave the other version of QR as an exercise.

4. d is the Gödel number of a correct derivation iff every line in it is an axiom, or justified by modus ponens or QR. Hence:

$$\mathrm{Deriv}(d) \Leftrightarrow (\forall i < \mathrm{len}(d))\ (\mathrm{IsAx}((d)_i) \vee \mathrm{MP}(d, i) \vee \mathrm{QR}(d, i)) \qquad \square$$

Problem 34.7. Define the following relations as in Proposition 34.27:

1. $\mathrm{IsAx}_{\varphi \to (\psi \to (\varphi \wedge \psi))}(n)$,

2. $\text{IsAx}_{\forall x\, \varphi(x) \to \varphi(t)}(n)$,

3. $\text{QR}_2(d, i)$ (for the other version of QR).

Proposition 34.28. *Suppose Γ is a primitive recursive set of sentences. Then the relation $\text{Prf}_\Gamma(x, y)$ expressing "x is the code of a derivation δ of φ from Γ and y is the Gödel number of φ" is primitive recursive.*

Proof. Suppose "$y \in \Gamma$" is given by the primitive recursive predicate $R_\Gamma(y)$. We have to show that the relation $\text{Prf}_\Gamma(x, y)$ is primitive recursive, where $\text{Prf}_\Gamma(x, y)$ holds iff y is the Gödel number of a sentence φ and x is the code of a derivation of φ from Γ.

By the previous proposition, the property $\text{Deriv}(x)$ which holds iff x is the code of a correct derivation δ is primitive recursive. However, that definition did not take into account the set Γ as an additional way to justify lines in the derivation. Our primitive recursive test of whether a line is justified by QR also left out of consideration the requirement that the constant c is not allowed to occur in Γ. It is possible to amend our definition so that it takes into account Γ directly, but it is easier to use Deriv and the deduction theorem. $\Gamma \vdash \varphi$ iff there is some finite list of sentences $\psi_1, \ldots, \psi_n \in \Gamma$ such that $\{\psi_1, \ldots, \psi_n\} \vdash \varphi$. And by the deduction theorem, this is the case if $\vdash (\psi_1 \to (\psi_2 \to \cdots (\psi_n \to \varphi) \cdots))$. Whether a sentence with Gödel number z is of this form can be tested primitive recursively. So, instead of considering x as the Gödel number of a derivation of the sentence with Gödel number y *from* Γ, we consider x as the Gödel number of a derivation of a nested conditional of the above form from $\emptyset$.

First, if we have a sequence of sentences, we can primitive recursively form the conditional with all these sentences as antecedents and given sentence as consequent:

$$\begin{aligned}
\text{hCond}(s, y, 0) &= y \\
\text{hCond}(s, y, n+1) &= {}^{\frown}(^{\#} \frown (s)_n \frown {}^{\frown}\!\to^{\#} \frown \text{Cond}(s, y, n) \frown {}^{\frown})^{\#} \\
\text{Cond}(s, y) &= \text{hCond}(s, y, \text{len}(s))
\end{aligned}$$

So we can define $\text{Prf}_\Gamma(x, y)$ by

$$\begin{aligned}
\text{Prf}_\Gamma(x, y) \Leftrightarrow (\exists s < \text{sequenceBound}(x, x))\ (& \\
& (x)_{\text{len}(x)-1} = \text{Cond}(s, y) \wedge \\
& (\forall i < \text{len}(s))\ (s)_i \in \Gamma \wedge \\
& \text{Deriv}(x)).
\end{aligned}$$

The bound on s is given by considering that each $(s)_i$ is the Gödel number of a sub-formula of the last line of the derivation, i.e., is less than $(x)_{\text{len}(x)-1}$. The number of antecedents $\psi \in \Gamma$, i.e., the length of s, is less than the length of the last line of x. □

Chapter 35

Representability in Q

content/incompleteness/representability-in-q/introduction.tex

35.1 Introduction

inc:req:int:sec

The incompleteness theorems apply to theories in which basic facts about computable functions can be expressed and proved. We will describe a very minimal such theory called "**Q**" (or, sometimes, "Robinson's Q," after Raphael Robinson). We will say what it means for a function to be *representable* in **Q**, and then we will prove the following:

> A function is representable in **Q** if and only if it is computable.

For one thing, this provides us with another model of computability. But we will also use it to show that the set $\{\varphi : \mathbf{Q} \vdash \varphi\}$ is not decidable, by reducing the halting problem to it. By the time we are done, we will have proved much stronger things than this.

The language of **Q** is the language of arithmetic; **Q** consists of the following axioms (to be used in conjunction with the other axioms and rules of first-order logic with identity predicate):

$$\forall x\, \forall y\, (x' = y' \to x = y) \tag{Q_1}$$
$$\forall x\, \mathsf{0} \neq x' \tag{Q_2}$$
$$\forall x\, (x = \mathsf{0} \vee \exists y\, x = y') \tag{Q_3}$$
$$\forall x\, (x + \mathsf{0}) = x \tag{Q_4}$$
$$\forall x\, \forall y\, (x + y') = (x + y)' \tag{Q_5}$$
$$\forall x\, (x \times \mathsf{0}) = \mathsf{0} \tag{Q_6}$$
$$\forall x\, \forall y\, (x \times y') = ((x \times y) + x) \tag{Q_7}$$
$$\forall x\, \forall y\, (x < y \leftrightarrow \exists z\, (z' + x) = y) \tag{Q_8}$$

For each natural number n, define the numeral $\overline{n}$ to be the term $\mathsf{o}^{\prime\prime\cdots\prime}$ where there are n tick marks in all. So, $\overline{0}$ is the constant symbol o by itself, $\overline{1}$ is o', $\overline{2}$ is o'', etc.

As a theory of arithmetic, **Q** is *extremely* weak; for example, you can't even prove very simple facts like $\forall x\, x \neq x'$ or $\forall x \forall y\, (x + y) = (y + x)$. But we will see that much of the reason that **Q** is so interesting is *because* it is so weak. In fact, it is just barely strong enough for the incompleteness theorem to hold. Another reason **Q** is interesting is because it has a *finite* set of axioms.

A stronger theory than **Q** (called *Peano arithmetic* **PA**) is obtained by adding a schema of induction to **Q**:

$$(\varphi(\mathsf{o}) \wedge \forall x\, (\varphi(x) \to \varphi(x'))) \to \forall x\, \varphi(x)$$

where $\varphi(x)$ is any formula. If $\varphi(x)$ contains free variables other than x, we add universal quantifiers to the front to bind all of them (so that the corresponding instance of the induction schema is a sentence). For instance, if $\varphi(x, y)$ also contains the variable y free, the corresponding instance is

$$\forall y\, ((\varphi(\mathsf{o}) \wedge \forall x\, (\varphi(x) \to \varphi(x'))) \to \forall x\, \varphi(x))$$

Using instances of the induction schema, one can prove much more from the axioms of **PA** than from those of **Q**. In fact, it takes a good deal of work to find "natural" statements about the natural numbers that can't be proved in Peano arithmetic!

inc:req:int: defn:representable-fn

Definition 35.1. A function $f(x_0, \ldots, x_k)$ from the natural numbers to the natural numbers is said to be *representable in* **Q** if there is a formula $\varphi_f(x_0, \ldots, x_k, y)$ such that whenever $f(n_0, \ldots, n_k) = m$, **Q** proves

inc:req:int: defn:rep:a

1. $\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{m})$

inc:req:int: defn:rep:b

2. $\forall y\, (\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, y) \to \overline{m} = y)$.

There are other ways of stating the definition; for example, we could equivalently require that **Q** proves $\forall y\, (\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, y) \leftrightarrow y = \overline{m})$.

inc:req:int: thm:representable-iff-comp

Theorem 35.2. *A function is representable in* **Q** *if and only if it is computable.*

There are two directions to proving the theorem. The left-to-right direction is fairly straightforward once arithmetization of syntax is in place. The other direction requires more work. Here is the basic idea: we pick "general recursive" as a way of making "computable" precise, and show that every general recursive function is representable in **Q**. Recall that a function is general recursive if it can be defined from zero, the successor function succ, and the projection functions P^n_i, using composition, primitive recursion, and regular minimization. So one way of showing that every general recursive function is representable in **Q** is to show that the basic functions are representable, and

whenever some functions are representable, then so are the functions defined from them using composition, primitive recursion, and regular minimization. In other words, we might show that the basic functions are representable, and that the representable functions are "closed under" composition, primitive recursion, and regular minimization. This guarantees that every general recursive function is representable.

It turns out that the step where we would show that representable functions are closed under primitive recursion is hard. In order to avoid this step, we show first that in fact we can do without primitive recursion. That is, we show that every general recursive function can be defined from basic functions using composition and regular minimization alone. To do this, we show that primitive recursion can actually be done by a specific regular minimization. However, for this to work, we have to add some additional basic functions: addition, multiplication, and the characteristic function of the identity relation $\chi_=$. Then, we can prove the theorem by showing that all of *these* basic functions are representable in **Q**, and the representable functions are closed under composition and regular minimization.

---

content/incompleteness/representability-in-q/representable-comp.tex

35.2 Functions Representable in Q are Computable

We'll prove that every function that is representable in **Q** is computable. We first have to establish a lemma about functions representable in **Q**. inc:req:rpc:sec

Lemma 35.3. *If $f(x_0, \ldots, x_k)$ is representable in* **Q**, *there is a formula $\varphi(x_0, \ldots, x_k, y)$ such that* lem:rep-q

$$\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{m}) \quad \textit{iff} \quad m = f(n_0, \ldots, n_k).$$

Proof. The "if" part is Definition 35.1(1). The "only if" part is seen as follows: Suppose $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{m})$ but $m \neq f(n_0, \ldots, n_k)$. Let $l = f(n_0, \ldots, n_k)$. By Definition 35.1(1), $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{l})$. By Definition 35.1(2), $\forall y\,(\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, y) \rightarrow \overline{l} = y)$. Using logic and the assumption that $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{m})$, we get that $\mathbf{Q} \vdash \overline{l} = \overline{m}$. On the other hand, by Lemma 35.14, $\mathbf{Q} \vdash \overline{l} \neq \overline{m}$. So **Q** is inconsistent. But that is impossible, since **Q** is satisfied by the standard model (see Definition 33.2), $\mathfrak{N} \models \mathbf{Q}$, and satisfiable theories are always consistent by the Soundness Theorem (Corollaries 19.31, 20.29, 21.31 and 22.38). □

Lemma 35.4. *Every function that is representable in* **Q** *is computable.*

Proof. Let's first give the intuitive idea for why this is true. To compute f, we do the following. List all the possible derivations δ in the language of arithmetic. This is possible to do mechanically. For each one, check if it is a derivation of a formula of the form $\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{m})$ (the formula representing f in **Q** from Lemma 35.3). If it is, $m = f(n_0, \ldots, n_k)$ by Lemma 35.3, and we've found the value of f. The search terminates because $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{f(n_0, \ldots, n_k)})$, so eventually we find a δ of the right sort.

This is not quite precise because our procedure operates on derivations and formulas instead of just on numbers, and we haven't explained exactly why "listing all possible derivations" is mechanically possible. But as we've seen, it is possible to code terms, formulas, and derivations by Gödel numbers. We've also introduced a precise model of computation, the general recursive functions. And we've seen that the relation $\mathrm{Prf}_{\mathbf{Q}}(d, y)$, which holds iff d is the Gödel number of a derivation of the formula with Gödel number y from the axioms of $\mathbf{Q}$, is (primitive) recursive. Other primitive recursive functions we'll need are num (Proposition 34.6) and Subst (Proposition 34.11). From these, it is possible to define f by minimization; thus, f is recursive.

First, define

$$\begin{aligned} A(n_0, \ldots, n_k, m) = \\ \mathrm{Subst}(\mathrm{Subst}(\ldots \mathrm{Subst}(&{}^{\ulcorner}\varphi_f{}^{\urcorner}, \mathrm{num}(n_0), {}^{\ulcorner}x_0{}^{\urcorner}), \\ &\ldots), \mathrm{num}(n_k), {}^{\ulcorner}x_k{}^{\urcorner}), \mathrm{num}(m), {}^{\ulcorner}y{}^{\urcorner}) \end{aligned}$$

This looks complicated, but it's just the function $A(n_0, \ldots, n_k, m) = {}^{\ulcorner}\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{m})^{\urcorner}$.

Now, consider the relation $R(n_0, \ldots, n_k, s)$ which holds if $(s)_0$ is the Gödel number of a derivation from $\mathbf{Q}$ of $\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{(s)_1})$:

$$R(n_0, \ldots, n_k, s) \quad \text{iff} \quad \mathrm{Prf}_{\mathbf{Q}}((s)_0, A(n_0, \ldots, n_k, (s)_1))$$

If we can find an s such that $R(n_0, \ldots, n_k, s)$ holds, we have found a pair of numbers—$(s)_0$ and $(s)_1$—such that $(s)_0$ is the Gödel number of a derivation of $A_f(\overline{n_0}, \ldots, \overline{n_k}, (s)_1)$. So looking for s is like looking for the pair d and m in the informal proof. And a computable function that "looks for" such an s can be defined by regular minimization. Note that R is regular: for every $n_0, \ldots, n_k$, there is a derivation δ of $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{f(n_0, \ldots, n_k)})$, so $R(n_0, \ldots, n_k, s)$ holds for $s = \langle {}^{\ulcorner}\delta{}^{\urcorner}, f(n_0, \ldots, n_k)\rangle$. So, we can write f as

$$f(n_0, \ldots, n_k) = (\mu s\ R(n_0, \ldots, n_k, s))_1. \qquad \square$$

---

content/incompleteness/representability-in-q/beta-function.tex

35.3 The Beta Function Lemma

inc:req:bet: sec

In order to show that we can carry out primitive recursion if addition, multiplication, and $\chi_=$ are available, we need to develop functions that handle sequences. (If we had exponentiation as well, our task would be easier.) When we had primitive recursion, we could define things like the "n-th prime," and pick a fairly straightforward coding. But here we do not have primitive recursion—in fact we want to show that we can do primitive recursion using minimization—so we need to be more clever.

inc:req:bet: lem:beta

Lemma 35.5. *There is a function $\beta(d, i)$ such that for every sequence $a_0, \ldots, a_n$ there is a number d, such that for every $i \leq n$, $\beta(d, i) = a_i$. Moreover,*

β can be defined from the basic functions using just composition and regular minimization.

Think of d as coding the sequence $\langle a_0, \ldots, a_n \rangle$, and $\beta(d, i)$ returning the i-th element. (Note that this "coding" does *not* use the power-of-primes coding we're already familiar with!). The lemma is fairly minimal; it doesn't say we can concatenate sequences or append elements, or even that we can *compute* d from $a_0, \ldots, a_n$ using functions definable by composition and regular minimization. All it says is that there is a "decoding" function such that every sequence is "coded."

The use of the notation β is Gödel's. To repeat, the hard part of proving the lemma is defining a suitable β using the seemingly restricted resources, i.e., using just composition and minimization—however, we're allowed to use addition, multiplication, and $\chi_=$. There are various ways to prove this lemma, but one of the cleanest is still Gödel's original method, which used a number-theoretic fact called Sunzi's Theorem (traditionally, the "Chinese Remainder Theorem").

Definition 35.6. Two natural numbers a and b are *relatively prime* iff their greatest common divisor is 1; in other words, they have no other divisors in common.

Definition 35.7. Natural numbers a and b are *congruent modulo* c, $a \equiv b \mod c$, iff $c \mid (a - b)$, i.e., a and b have the same remainder when divided by c.

Here is Sunzi's Theorem:

Theorem 35.8. *Suppose $x_0, \ldots, x_n$ are (pairwise) relatively prime. Let $y_0, \ldots, y_n$ be any numbers. Then there is a number z such that*

$$\begin{aligned} z &\equiv y_0 \mod x_0 \\ z &\equiv y_1 \mod x_1 \\ &\vdots \\ z &\equiv y_n \mod x_n. \end{aligned}$$

Here is how we will use Sunzi's Theorem: if $x_0, \ldots, x_n$ are bigger than $y_0, \ldots, y_n$ respectively, then we can take z to code the sequence $\langle y_0, \ldots, y_n \rangle$. To recover y_i, we need only divide z by x_i and take the remainder. To use this coding, we will need to find suitable values for $x_0, \ldots, x_n$.

A couple of observations will help us in this regard. Given $y_0, \ldots, y_n$, let

$$\begin{aligned} j &= \max(n, y_0 + 1, \ldots, y_n + 1), \\ m &= \operatorname{lcm}(1, \ldots, j), \end{aligned}$$

and let

$$\begin{aligned} x_0 &= 1 + m \\ x_1 &= 1 + 2 \cdot m \\ x_2 &= 1 + 3 \cdot m \\ &\vdots \\ x_n &= 1 + (n+1) \cdot m \end{aligned}$$

Then two things are true:

inc:req:bet: rel-prime
1. $x_0, \ldots, x_n$ are relatively prime.

inc:req:bet: less
2. For each i, $y_i < x_i$.

To see that (1) is true, note that if p is a prime number and $p \mid x_i$ and $p \mid x_k$, then $p \mid 1 + (i+1)m$ and $p \mid 1 + (k+1)m$. But then p divides their difference,

$$(1 + (i+1)m) - (1 + (k+1)m) = (i-k)m.$$

Since p divides $1 + (i+1)m$, it can't divide m as well (otherwise, the first division would leave a remainder of 1). So p divides $i - k$, since p divides $(i-k)m$. But $|i-k|$ is at most n, and we have chosen $j \geq n$, so this implies that $p \mid m$, again a contradiction. So there is no prime number dividing both x_i and x_k. Clause (2) is easy: we have $y_i < j \leq m < x_i$.

Now let us prove the β function lemma. Remember that we can use 0, successor, plus, times, $\chi_=$, projections, and any function defined from them using composition and minimization applied to regular functions. We can also use a relation if its characteristic function is so definable. As before we can show that these relations are closed under Boolean combinations and bounded quantification; for example:

$$\begin{aligned} \text{not}(x) &= \chi_=(x, 0) \\ (\min x \leq z)\, R(x, y) &= \mu x\, (R(x, y) \vee x = z) \\ (\exists x \leq z)\, R(x, y) &\Leftrightarrow R((\min x \leq z)\, R(x, y), y) \end{aligned}$$

We can then show that all of the following are also definable without primitive recursion:

1. The pairing function, $J(x, y) = \frac{1}{2}[(x + y)(x + y + 1)] + x$;
2. the projection functions

$$\begin{aligned} K(z) &= (\min x \leq z)\, (\exists y \leq z)\, z = J(x, y), \\ L(z) &= (\min y \leq z)\, (\exists x \leq z)\, z = J(x, y); \end{aligned}$$

3. the less-than relation $x < y$;

4. the divisibility relation $x \mid y$;
5. the function $\text{rem}(x, y)$ which returns the remainder when y is divided by x.

Now define

$$\beta^*(d_0, d_1, i) = \text{rem}(1 + (i+1)d_1, d_0) \text{ and}$$
$$\beta(d, i) = \beta^*(K(d), L(d), i).$$

This is the function we want. Given $a_0, \ldots, a_n$ as above, let

$$j = \max(n, a_0 + 1, \ldots, a_n + 1),$$

and let $d_1 = \text{lcm}(1, \ldots, j)$. By (1) above, we know that $1 + d_1$, $1 + 2d_1$, $\ldots$, $1+(n+1)d_1$ are relatively prime, and by (2) that all are greater than $a_0, \ldots, a_n$. By Sunzi's Theorem there is a value d_0 such that for each i,

$$d_0 \equiv a_i \mod (1 + (i+1)d_1)$$

and so (because d_1 is greater than a_i),

$$a_i = \text{rem}(1 + (i+1)d_1, d_0).$$

Let $d = J(d_0, d_1)$. Then for each $i \leq n$, we have

$$\begin{aligned}\beta(d, i) &= \beta^*(d_0, d_1, i) \\ &= \text{rem}(1 + (i+1)d_1, d_0) \\ &= a_i\end{aligned}$$

which is what we need. This completes the proof of the β-function lemma.

Problem 35.1. Show that the relations $x < y$, $x \mid y$, and the function $\text{rem}(x, y)$ can be defined without primitive recursion. You may use 0, successor, plus, times, $\chi_=$, projections, and bounded minimization and quantification.

---

content/incompleteness/representability-in-q/prim-rec.tex

35.4 Simulating Primitive Recursion

inc:req:prim-sec

Now we can show that definition by primitive recursion can be "simulated" by regular minimization using the beta function. Suppose we have $f(\vec{x})$ and $g(\vec{x}, y, z)$. Then the function $h(x, \vec{z})$ defined from f and g by primitive recursion is

$$h(\vec{x}, 0) = f(\vec{x})$$
$$h(\vec{x}, y+1) = g(\vec{x}, y, h(\vec{x}, y)).$$

We need to show that h can be defined from f and g using just composition and regular minimization, using the basic functions and functions defined from them using composition and regular minimization (such as β).

inc:req:pri: lem:prim-rec **Lemma 35.9.** *If h can be defined from f and g using primitive recursion, it can be defined from f, g, the functions* zero, succ, P^n_i, add, mult, $\chi_=$, *using composition and regular minimization.*

Proof. First, define an auxiliary function $\hat{h}(\vec{x}, y)$ which returns the least number d such that d codes a sequence which satisfies

1. $(d)_0 = f(\vec{x})$, and
2. for each $i < y$, $(d)_{i+1} = g(\vec{x}, i, (d)_i)$,

where now $(d)_i$ is short for $\beta(d, i)$. In other words, $\hat{h}$ returns the sequence $\langle h(\vec{x}, 0), h(\vec{x}, 1), \ldots, h(\vec{x}, y)\rangle$. We can write $\hat{h}$ as

$$\hat{h}(\vec{x}, y) = \mu d\ (\beta(d, 0) = f(\vec{x}) \wedge (\forall i < y)\ \beta(d, i+1) = g(\vec{x}, i, \beta(d, i))).$$

Note: no primitive recursion is needed here, just minimization. The function we minimize is regular because of the beta function lemma Lemma 35.5.

But now we have

$$h(\vec{x}, y) = \beta(\hat{h}(\vec{x}, y), y),$$

so h can be defined from the basic functions using just composition and regular minimization. □

---

content/incompleteness/representability-in-q/basic-representable.tex

35.5 Basic Functions are Representable in Q

inc:req:bre: sec First we have to show that all the basic functions are representable in **Q**. In the end, we need to show how to assign to each k-ary basic function $f(x_0, \ldots, x_{k-1})$ a formula $\varphi_f(x_0, \ldots, x_{k-1}, y)$ that represents it.

We will be able to represent zero, successor, plus, times, the characteristic function for equality, and projections. In each case, the appropriate representing function is entirely straightforward; for example, zero is represented by the formula $y = 0$, successor is represented by the formula $x'_0 = y$, and addition is represented by the formula $(x_0 + x_1) = y$. The work involves showing that **Q** can prove the relevant sentences; for example, saying that addition is represented by the formula above involves showing that for every pair of natural numbers m and n, **Q** proves

$$\overline{n} + \overline{m} = \overline{n+m} \text{ and}$$
$$\forall y\,((\overline{n} + \overline{m}) = y \to y = \overline{n+m}).$$

inc:req:bre: prop:rep-zero **Proposition 35.10.** *The zero function* $\mathrm{zero}(x) = 0$ *is represented in* **Q** *by* $\varphi_{\mathrm{zero}}(x, y) \equiv y = 0$.

inc:req:bre: prop:rep-succ **Proposition 35.11.** *The successor function* $\mathrm{succ}(x) = x + 1$ *is represented in* **Q** *by* $\varphi_{\mathrm{succ}}(x, y) \equiv y = x'$.

Proposition 35.12. *The projection function $P^n_i(x_0, \ldots, x_{n-1}) = x_i$ is represented in* **Q** *by* inc:req:bre: prop:rep-proj

$$\varphi_{P^n_i}(x_0, \ldots, x_{n-1}, y) \equiv y = x_i.$$

Problem 35.2. Prove that $y = 0$, $y = x'$, and $y = x_i$ represent zero, succ, and P^n_i, respectively.

Proposition 35.13. *The characteristic function of $=$,* inc:req:bre: prop:rep-id

$$\chi_=(x_0, x_1) = \begin{cases} 1 & \text{if } x_0 = x_1 \\ 0 & \text{otherwise} \end{cases}$$

is represented in **Q** *by*

$$\varphi_{\chi_=}(x_0, x_1, y) \equiv (x_0 = x_1 \wedge y = \overline{1}) \vee (x_0 \neq x_1 \wedge y = \overline{0}).$$

The proof requires the following lemma.

Lemma 35.14. *Given natural numbers n and m, if $n \neq m$, then* $\mathbf{Q} \vdash \overline{n} \neq \overline{m}$. inc:req:bre: lem:q-proves-neq

Proof. Use induction on n to show that for every m, if $n \neq m$, then $Q \vdash \overline{n} \neq \overline{m}$.

In the base case, $n = 0$. If m is not equal to 0, then $m = k + 1$ for some natural number k. We have an axiom that says $\forall x\, 0 \neq x'$. By a quantifier axiom, replacing x by $\overline{k}$, we can conclude $0 \neq \overline{k}'$. But $\overline{k}'$ is just $\overline{m}$.

In the induction step, we can assume the claim is true for n, and consider $n+1$. Let m be any natural number. There are two possibilities: either $m = 0$ or for some k we have $m = k + 1$. The first case is handled as above. In the second case, suppose $n + 1 \neq k + 1$. Then $n \neq k$. By the induction hypothesis for n we have $\mathbf{Q} \vdash \overline{n} \neq \overline{k}$. We have an axiom that says $\forall x\, \forall y\, x' = y' \to x = y$. Using a quantifier axiom, we have $\overline{n}' = \overline{k}' \to \overline{n} = \overline{k}$. Using propositional logic, we can conclude, in **Q**, $\overline{n} \neq \overline{k} \to \overline{n}' \neq \overline{k}'$. Using modus ponens, we can conclude $\overline{n}' \neq \overline{k}'$, which is what we want, since $\overline{k}'$ is $\overline{m}$. □

explanation

Note that the lemma does not say much: in essence it says that **Q** can prove that different numerals denote different objects. For example, **Q** proves $0'' \neq 0'''$. But showing that this holds in general requires some care. Note also that although we are using induction, it is induction *outside* of **Q**.

Proof of Proposition 35.13. If $n = m$, then $\overline{n}$ and $\overline{m}$ are the same term, and $\chi_=(n, m) = 1$. But $\mathbf{Q} \vdash (\overline{n} = \overline{m} \wedge \overline{1} = \overline{1})$, so it proves $\varphi_=(\overline{n}, \overline{m}, \overline{1})$. If $n \neq m$, then $\chi_=(n, m) = 0$. By Lemma 35.14, $\mathbf{Q} \vdash \overline{n} \neq \overline{m}$ and so also $(\overline{n} \neq \overline{m} \wedge 0 = 0)$. Thus $\mathbf{Q} \vdash \varphi_=(\overline{n}, \overline{m}, \overline{0})$.

For the second part, we also have two cases. If $n = m$, we have to show that $\mathbf{Q} \vdash \forall y\, (\varphi_=(\overline{n}, \overline{m}, y) \to y = \overline{1})$. Arguing informally, suppose $\varphi_=(\overline{n}, \overline{m}, y)$, i.e.,

$$(\overline{n} = \overline{n} \wedge y = \overline{1}) \vee (\overline{n} \neq \overline{n} \wedge y = \overline{0})$$

The left disjunct implies $y = \overline{1}$ by logic; the right contradicts $\overline{n} = \overline{n}$ which is provable by logic.

Suppose, on the other hand, that $n \neq m$. Then $\varphi_=(\overline{n}, \overline{m}, y)$ is

$$(\overline{n} = \overline{m} \wedge y = \overline{1}) \vee (\overline{n} \neq \overline{m} \wedge y = \overline{0})$$

Here, the left disjunct contradicts $\overline{n} \neq \overline{m}$, which is provable in **Q** by Lemma 35.14; the right disjunct entails $y = \overline{0}$. □

inc:req:bre: prop:rep-add

Proposition 35.15. *The addition function* $\mathrm{add}(x_0, x_1) = x_0 + x_1$ *is represented in* **Q** *by*

$$\varphi_{\mathrm{add}}(x_0, x_1, y) \equiv y = (x_0 + x_1).$$

inc:req:bre: lem:q-proves-add

Lemma 35.16. $\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n + m}$

Proof. We prove this by induction on m. If $m = 0$, the claim is that $\mathbf{Q} \vdash (\overline{n} + \mathsf{o}) = \overline{n}$. This follows by axiom Q_4. Now suppose the claim for m; let's prove the claim for $m + 1$, i.e., prove that $\mathbf{Q} \vdash (\overline{n} + \overline{m+1}) = \overline{n+m+1}$. Note that $\overline{m+1}$ is just $\overline{m}'$, and $\overline{n+m+1}$ is just $\overline{n+m}'$. By axiom Q_5, $\mathbf{Q} \vdash (\overline{n} + \overline{m}') = (\overline{n} + \overline{m})'$. By induction hypothesis, $\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n+m}$. So $\mathbf{Q} \vdash (\overline{n} + \overline{m}') = \overline{n+m}'$. □

Proof of Proposition 35.15. The formula $\varphi_{\mathrm{add}}(x_0, x_1, y)$ representing add is $y = (x_0 + x_1)$. First we show that if $\mathrm{add}(n, m) = k$, then $\mathbf{Q} \vdash \varphi_{\mathrm{add}}(\overline{n}, \overline{m}, \overline{k})$, i.e., $\mathbf{Q} \vdash \overline{k} = (\overline{n} + \overline{m})$. But since $k = n + m$, $\overline{k}$ just is $\overline{n+m}$, and we've shown in Lemma 35.16 that $\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n+m}$.

We also have to show that if $\mathrm{add}(n, m) = k$, then

$$\mathbf{Q} \vdash \forall y\, (\varphi_{\mathrm{add}}(\overline{n}, \overline{m}, y) \to y = \overline{k}).$$

Suppose we have $(\overline{n} + \overline{m}) = y$. Since

$$\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n+m},$$

we can replace the left side with $\overline{n+m}$ and get $\overline{n+m} = y$, for arbitrary y. □

inc:req:bre: prop:rep-mult

Proposition 35.17. *The multiplication function* $\mathrm{mult}(x_0, x_1) = x_0 \cdot x_1$ *is represented in* **Q** *by*

$$\varphi_{\mathrm{mult}}(x_0, x_1, y) \equiv y = (x_0 \times x_1).$$

Proof. Exercise. □

inc:req:bre: lem:q-proves-mult

Lemma 35.18. $\mathbf{Q} \vdash (\overline{n} \times \overline{m}) = \overline{n \cdot m}$

Proof. Exercise. □

Problem 35.3. Prove Lemma 35.18.

Problem 35.4. Use Lemma 35.18 to prove Proposition 35.17.

explanation

Recall that we use $\times$ for the function symbol of the language of arithmetic, and $\cdot$ for the ordinary multiplication operation on numbers. So $\cdot$ can appear between expressions for numbers (such as in $m \cdot n$) while $\times$ appears only between terms of the language of arithmetic (such as in $(\overline{m} \times \overline{n})$). Even more confusingly, $+$ is used for both the function symbol and the addition operation. When it appears between terms—e.g., in $(\overline{n} + \overline{m})$—it is the 2-place function symbol of the language of arithmetic, and when it appears between numbers—e.g., in $n + m$—it is the addition operation. This includes the case $\overline{n + m}$: this is the standard numeral corresponding to the number $n + m$.

---

content/incompleteness/representability-in-q/composition-representable.tex

35.6 Composition is Representable in Q

Suppose h is defined by

inc:req:cmp:sec

$$h(x_0, \ldots, x_{l-1}) = f(g_0(x_0, \ldots, x_{l-1}), \ldots, g_{k-1}(x_0, \ldots, x_{l-1})).$$

where we have already found formulas $\varphi_f, \varphi_{g_0}, \ldots, \varphi_{g_{k-1}}$ representing the functions f, and g_0, …, g_{k-1}, respectively. We have to find a formula φ_h representing h.

Let's start with a simple case, where all functions are 1-place, i.e., consider $h(x) = f(g(x))$. If $\varphi_f(y, z)$ represents f, and $\varphi_g(x, y)$ represents g, we need a formula $\varphi_h(x, z)$ that represents h. Note that $h(x) = z$ iff there is a y such that both $z = f(y)$ and $y = g(x)$. (If $h(x) = z$, then $g(x)$ is such a y; if such a y exists, then since $y = g(x)$ and $z = f(y)$, $z = f(g(x))$.) This suggests that $\exists y\, (\varphi_g(x, y) \wedge \varphi_f(y, z))$ is a good candidate for $\varphi_h(x, z)$. We just have to verify that **Q** proves the relevant formulas.

Proposition 35.19. *If $h(n) = m$, then $\mathbf{Q} \vdash \varphi_h(\overline{n}, \overline{m})$.*

inc:req:cmp:prop:rep1

Proof. Suppose $h(n) = m$, i.e., $f(g(n)) = m$. Let $k = g(n)$. Then

$$\mathbf{Q} \vdash \varphi_g(\overline{n}, \overline{k})$$

since φ_g represents g, and

$$\mathbf{Q} \vdash \varphi_f(\overline{k}, \overline{m})$$

since φ_f represents f. Thus,

$$\mathbf{Q} \vdash \varphi_g(\overline{n}, \overline{k}) \wedge \varphi_f(\overline{k}, \overline{m})$$

and consequently also

$$\mathbf{Q} \vdash \exists y\, (\varphi_g(\overline{n}, y) \wedge \varphi_f(y, \overline{m})),$$

i.e., $\mathbf{Q} \vdash \varphi_h(\overline{n}, \overline{m})$. □

inc:req:cmp: prop:rep2

Proposition 35.20. *If $h(n) = m$, then $\mathbf{Q} \vdash \forall z\,(\varphi_h(\overline{n}, z) \to z = \overline{m})$.*

Proof. Suppose $h(n) = m$, i.e., $f(g(n)) = m$. Let $k = g(n)$. Then

$$\mathbf{Q} \vdash \forall y\,(\varphi_g(\overline{n}, y) \to y = \overline{k})$$

since φ_g represents g, and

$$\mathbf{Q} \vdash \forall z\,(\varphi_f(\overline{k}, z) \to z = \overline{m})$$

since φ_f represents f. Using just a little bit of logic, we can show that also

$$\mathbf{Q} \vdash \forall z\,(\exists y\,(\varphi_g(\overline{n}, y) \wedge \varphi_f(y, z)) \to z = \overline{m}).$$

i.e., $\mathbf{Q} \vdash \forall y\,(\varphi_h(\overline{n}, y) \to y = \overline{m})$. □

The same idea works in the more complex case where f and g_i have arity greater than 1.

inc:req:cmp: prop:rep-composition

Proposition 35.21. *If $\varphi_f(y_0, \ldots, y_{k-1}, z)$ represents $f(y_0, \ldots, y_{k-1})$ in $\mathbf{Q}$, and $\varphi_{g_i}(x_0, \ldots, x_{l-1}, y)$ represents $g_i(x_0, \ldots, x_{l-1})$ in $\mathbf{Q}$, then*

$$\exists y_0 \ldots \exists y_{k-1}\,(\varphi_{g_0}(x_0, \ldots, x_{l-1}, y_0) \wedge \cdots \wedge \varphi_{g_{k-1}}(x_0, \ldots, x_{l-1}, y_{k-1}) \wedge \varphi_f(y_0, \ldots, y_{k-1}, z))$$

represents

$$h(x_0, \ldots, x_{l-1}) = f(g_0(x_0, \ldots, x_{l-1}), \ldots, g_{k-1}(x_0, \ldots, x_{l-1})).$$

Proof. Exercise. □

Problem 35.5. Using the proofs of Proposition 35.20 and Proposition 35.20 as a guide, carry out the proof of Proposition 35.21 in detail.

---

content/incompleteness/representability-in-q/minimization-representable.tex

35.7 Regular Minimization is Representable in Q

inc:req:min: sec

Let's consider unbounded search. Suppose $g(x, z)$ is regular and representable in $\mathbf{Q}$, say by the formula $\varphi_g(x, z, y)$. Let f be defined by $f(z) = \mu x\,[g(x, z) = 0]$. We would like to find a formula $\varphi_f(z, y)$ representing f. The value of $f(z)$ is that number x which (a) satisfies $g(x, z) = 0$ and (b) is the least such, i.e., for any $w < x$, $g(w, z) \neq 0$. So the following is a natural choice:

$$\varphi_f(z, y) \equiv \varphi_g(y, z, 0) \wedge \forall w\,(w < y \to \neg\varphi_g(w, z, 0)).$$

In the general case, of course, we would have to replace z with $z_0, \ldots, z_k$.

The proof, again, will involve some lemmas about things $\mathbf{Q}$ is strong enough to prove.

Lemma 35.22. *For every constant symbol a and every natural number n,* inc:req:min: lem:succ

$$\mathbf{Q} \vdash (a' + \overline{n}) = (a + \overline{n})'.$$

Proof. The proof is, as usual, by induction on n. In the base case, $n = 0$, we need to show that **Q** proves $(a' + \mathsf{o}) = (a + \mathsf{o})'$. But we have:

$$\mathbf{Q} \vdash (a' + \mathsf{o}) = a' \quad \text{by axiom } Q_4 \tag{35.1}$$ inc:req:min: step1

$$\mathbf{Q} \vdash (a + \mathsf{o}) = a \quad \text{by axiom } Q_4 \tag{35.2}$$ inc:req:min: step2

$$\mathbf{Q} \vdash (a + \mathsf{o})' = a' \quad \text{by eq. (35.2)} \tag{35.3}$$ inc:req:min: step3

$$\mathbf{Q} \vdash (a' + \mathsf{o}) = (a + \mathsf{o})' \quad \text{by eq. (35.1) and eq. (35.3)}$$

In the induction step, we can assume that we have shown that $\mathbf{Q} \vdash (a' + \overline{n}) = (a + \overline{n})'$. Since $\overline{n+1}$ is $\overline{n}'$, we need to show that **Q** proves $(a' + \overline{n}') = (a + \overline{n}')'$. We have:

$$\mathbf{Q} \vdash (a' + \overline{n}') = (a' + \overline{n})' \quad \text{by axiom } Q_5 \tag{35.4}$$ inc:req:min: step5

$$\mathbf{Q} \vdash (a' + \overline{n}') = (a + \overline{n}')' \quad \text{inductive hypothesis} \tag{35.5}$$ inc:req:min: step6

$$\mathbf{Q} \vdash (a' + \overline{n})' = (a + \overline{n}')' \quad \text{by eq. (35.4) and eq. (35.5).}$$ □

It is again worth mentioning that this is weaker than saying that **Q** proves $\forall x \forall y\, (x' + y) = (x + y)'$. Although this sentence is true in $\mathfrak{N}$, **Q** does not prove it.

Lemma 35.23. $\mathbf{Q} \vdash \forall x \neg x < \mathsf{o}$. inc:req:min: lem:less-zero

Proof. We give the proof informally (i.e., only giving hints as to how to construct the formal derivation).

We have to prove $\neg a < \mathsf{o}$ for an arbitrary a. By the definition of $<$, we need to prove $\neg \exists y\, (y' + a) = \mathsf{o}$ in **Q**. We'll assume $\exists y\, (y' + a) = \mathsf{o}$ and prove a contradiction. Suppose $(b' + a) = \mathsf{o}$. Using Q_3, we have that $a = \mathsf{o} \vee \exists y\, a = y'$. We distinguish cases.

Case 1: $a = \mathsf{o}$ holds. From $(b' + a) = \mathsf{o}$, we have $(b' + \mathsf{o}) = \mathsf{o}$. By axiom Q_4 of **Q**, we have $(b' + \mathsf{o}) = b'$, and hence $b' = \mathsf{o}$. But by axiom Q_2 we also have $b' \neq \mathsf{o}$, a contradiction.

Case 2: For some c, $a = c'$. But then we have $(b' + c') = \mathsf{o}$. By axiom Q_5, we have $(b' + c)' = \mathsf{o}$, again contradicting axiom Q_2. □

Lemma 35.24. *For every natural number n,* inc:req:min: lem:less-nsucc

$$\mathbf{Q} \vdash \forall x\, (x < \overline{n+1} \to (x = \mathsf{o} \vee \cdots \vee x = \overline{n})).$$

Proof. We use induction on n. Let us consider the base case, when $n = 0$. In that case, we need to show $a < \overline{1} \to a = \mathsf{o}$, for arbitrary a. Suppose $a < \overline{1}$. Then by the defining axiom for $<$, we have $\exists y\, (y' + a) = \mathsf{o}'$ (since $\overline{1} \equiv \mathsf{o}'$).

Suppose b has that property, i.e., we have $(b' + a) = \mathsf{o}'$. We need to show $a = \mathsf{o}$. By axiom Q_3, we have either $a = \mathsf{o}$ or that there is a c such that $a = c'$.

In the former case, there is nothing to show. So suppose $a = c'$. Then we have $(b' + c') = \mathsf{o}'$. By axiom Q_5 of **Q**, we have $(b' + c)' = \mathsf{o}'$. By axiom Q_1, we have $(b' + c) = \mathsf{o}$. But this means, by axiom Q_8, that $c < \mathsf{o}$, contradicting Lemma 35.23.

Now for the inductive step. We prove the case for $n+1$, assuming the case for n. So suppose $a < \overline{n+2}$. Again using Q_3 we can distinguish two cases: $a = \mathsf{o}$ and for some b, $a = b'$. In the first case, $a = \mathsf{o} \vee \dots \vee a = \overline{n+1}$ follows trivially. In the second case, we have $b' < \overline{n+2}$, i.e., $b' < \overline{n+1}'$. By axiom Q_8, for some c, $(c' + b') = \overline{n+1}'$. By axiom Q_5, $(c' + b)' = \overline{n+1}'$. By axiom Q_1, $(c' + b) = \overline{n+1}$, and so $b < \overline{n+1}$ by axiom Q_8. By inductive hypothesis, $b = \mathsf{o} \vee \dots \vee b = \overline{n}$. From this, we get $b' = \mathsf{o}' \vee \dots \vee b' = \overline{n}'$ by logic, and so $a = \overline{1} \vee \dots \vee a = \overline{n+1}$ since $a = b'$. □

inc:req:min: lem:trichotomy

Lemma 35.25. *For every natural number m,*

$$\mathbf{Q} \vdash \forall y\,((y < \overline{m} \vee \overline{m} < y) \vee y = \overline{m}).$$

Proof. By induction on m. First, consider the case $m = 0$. $\mathbf{Q} \vdash \forall y\,(y = \mathsf{o} \vee \exists z\, y = z')$ by Q_3. Let a be arbitrary. Then either $a = \mathsf{o}$ or for some b, $a = b'$. In the former case, we also have $(a < \mathsf{o} \vee \mathsf{o} < a) \vee a = \mathsf{o}$. But if $a = b'$, then $(b' + \mathsf{o}) = (a + \mathsf{o})$ by the logic of $=$. By Q_4, $(a + \mathsf{o}) = a$, so we have $(b' + \mathsf{o}) = a$, and hence $\exists z\,(z' + \mathsf{o}) = a$. By the definition of $<$ in Q_8, $\mathsf{o} < a$. If $\mathsf{o} < a$, then also $(\mathsf{o} < a \vee a < \mathsf{o}) \vee a = \mathsf{o}$.

Now suppose we have

$$\mathbf{Q} \vdash \forall y\,((y < \overline{m} \vee \overline{m} < y) \vee y = \overline{m})$$

and we want to show

$$\mathbf{Q} \vdash \forall y\,((y < \overline{m+1} \vee \overline{m+1} < y) \vee y = \overline{m+1})$$

Let a be arbitrary. By Q_3, either $a = \mathsf{o}$ or for some b, $a = b'$. In the first case, we have $\overline{m}' + a = \overline{m+1}$ by Q_4, and so $a < \overline{m+1}$ by Q_8.

Now consider the second case, $a = b'$. By the induction hypothesis, $(b < \overline{m} \vee \overline{m} < b) \vee b = \overline{m}$.

The first disjunct $b < \overline{m}$ is equivalent (by Q_8) to $\exists z\,(z' + b) = \overline{m}$. Suppose c has this property. If $(c' + b) = \overline{m}$, then also $(c' + b)' = \overline{m}'$. By Q_5, $(c' + b)' = (c' + b')$. Hence, $(c' + b') = \overline{m}'$. We get $\exists u\,(u' + b') = \overline{m+1}$ by existentially generalizing on c' and keeping in mind that $\overline{m}' \equiv \overline{m+1}$. Hence, if $b < \overline{m}$ then $b' < \overline{m+1}$ and so $a < \overline{m+1}$.

Now suppose $\overline{m} < b$, i.e., $\exists z\,(z' + \overline{m}) = b$. Suppose c is such a z, i.e., $(c' + \overline{m}) = b$. By logic, $(c' + \overline{m})' = b'$. By Q_5, $(c' + \overline{m}') = b'$. Since $a = b'$ and $\overline{m}' \equiv \overline{m+1}$, $(c' + \overline{m+1}) = a$. By Q_8, $\overline{m+1} < a$.

Finally, assume $b = \overline{m}$. Then, by logic, $b' = \overline{m}'$, and so $a = \overline{m+1}$.

Hence, from each disjunct of the case for m and b, we can obtain the corresponding disjunct for $m+1$ and a. □

Proposition 35.26. *If $\varphi_g(x, z, y)$ represents $g(x, z)$ in* **Q***, then* inc:req:min: prop:rep-minimization

$$\varphi_f(z, y) \equiv \varphi_g(y, z, \mathsf{o}) \wedge \forall w\,(w < y \to \neg\varphi_g(w, z, \mathsf{o}))$$

represents $f(z) = \mu x\,[g(x, z) = 0]$.

Proof. First we show that if $f(n) = m$, then $\mathbf{Q} \vdash \varphi_f(\overline{n}, \overline{m})$, i.e.,

$$\mathbf{Q} \vdash \varphi_g(\overline{m}, \overline{n}, \mathsf{o}) \wedge \forall w\,(w < \overline{m} \to \neg\varphi_g(w, \overline{n}, \mathsf{o})).$$

Since $\varphi_g(x, z, y)$ represents $g(x, z)$ and $g(m, n) = 0$ if $f(n) = m$, we have

$$\mathbf{Q} \vdash \varphi_g(\overline{m}, \overline{n}, \mathsf{o}).$$

If $f(n) = m$, then for every $k < m$, $g(k, n) \neq 0$. So

$$\mathbf{Q} \vdash \neg\varphi_g(\overline{k}, \overline{n}, \mathsf{o}).$$

We get that

$$\mathbf{Q} \vdash \forall w\,(w < \overline{m} \to \neg\varphi_g(w, \overline{n}, \mathsf{o})). \tag{35.6}$$ inc:req:min: rep-less

by Lemma 35.23 in case $m = 0$ and by Lemma 35.24 otherwise.

Now let's show that if $f(n) = m$, then $\mathbf{Q} \vdash \forall y\,(\varphi_f(\overline{n}, y) \to y = \overline{m})$. We again sketch the argument informally, leaving the formalization to the reader.

Suppose $\varphi_f(\overline{n}, b)$. From this we get (a) $\varphi_g(b, \overline{n}, \mathsf{o})$ and (b) $\forall w\,(w < b \to \neg\varphi_g(w, \overline{n}, \mathsf{o}))$. By Lemma 35.25, $(b < \overline{m} \vee \overline{m} < b) \vee b = \overline{m}$. We'll show that both $b < \overline{m}$ and $\overline{m} < b$ leads to a contradiction.

If $\overline{m} < b$, then $\neg\varphi_g(\overline{m}, \overline{n}, \mathsf{o})$ from (b). But $m = f(n)$, so $g(m, n) = 0$, and so $\mathbf{Q} \vdash \varphi_g(\overline{m}, \overline{n}, \mathsf{o})$ since φ_g represents g. So we have a contradiction.

Now suppose $b < \overline{m}$. Then since $\mathbf{Q} \vdash \forall w\,(w < \overline{m} \to \neg\varphi_g(w, \overline{n}, \mathsf{o}))$ by eq. (35.6), we get $\neg\varphi_g(b, \overline{n}, \mathsf{o})$. This again contradicts (a). □

---

content/incompleteness/representability-in-q/comp-representable.tex

35.8 Computable Functions are Representable in Q

inc:req:crq: sec

Theorem 35.27. *Every computable function is representable in* **Q***.*

Proof. For definiteness, and using the Church–Turing Thesis, let's say that a function is computable iff it is general recursive. The general recursive functions are those which can be defined from the zero function zero, the successor function succ, and the projection function P^n_i using composition, primitive recursion, and regular minimization. By Lemma 35.9, any function h that can be defined from f and g can also be defined using composition and regular

minimization from f, g, and zero, succ, P^n_i, add, mult, $\chi_=$. Consequently, a function is general recursive iff it can be defined from zero, succ, P^n_i, add, mult, $\chi_=$ using composition and regular minimization.

We've furthermore shown that the basic functions in question are representable in **Q** (Propositions 35.10 to 35.13, 35.15 and 35.17), and that any function defined from representable functions by composition or regular minimization (Proposition 35.21, Proposition 35.26) is also representable. Thus every general recursive function is representable in **Q**. □

We have shown that the set of computable functions can be characterized as the set of functions representable in **Q**. In fact, the proof is more general. From the definition of representability, it is not hard to see that any theory extending **Q** (or in which one can interpret **Q**) can represent the computable functions. But, conversely, in any derivation system in which the notion of derivation is computable, every representable function is computable. So, for example, the set of computable functions can be characterized as the set of functions representable in Peano arithmetic, or even Zermelo–Fraenkel set theory. As Gödel noted, this is somewhat surprising. We will see that when it comes to provability, questions are very sensitive to which theory you consider; roughly, the stronger the axioms, the more you can prove. But across a wide range of axiomatic theories, the representable functions are exactly the computable ones; stronger theories do not represent more functions as long as they are axiomatizable.

explanation

---

content/incompleteness/representability-in-q/representing-relations.tex

35.9 Representing Relations

inc:req:rel: sec

Let us say what it means for a *relation* to be representable.

inc:req:rel: defn:representing-relations

Definition 35.28. A relation $R(x_0, \ldots, x_k)$ on the natural numbers is *representable in* **Q** if there is a formula $\varphi_R(x_0, \ldots, x_k)$ such that whenever $R(n_0, \ldots, n_k)$ is true, **Q** proves $\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$, and whenever $R(n_0, \ldots, n_k)$ is false, **Q** proves $\neg\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$.

inc:req:rel: thm:representing-rels

Theorem 35.29. *A relation is representable in* **Q** *if and only if it is computable.*

Proof. For the forwards direction, suppose $R(x_0, \ldots, x_k)$ is represented by the formula $\varphi_R(x_0, \ldots, x_k)$. Here is an algorithm for computing R: on input n_0, $\ldots$, n_k, simultaneously search for a proof of $\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$ and a proof of $\neg\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$. By our hypothesis, the search is bound to find one or the other; if it is the first, report "yes," and otherwise, report "no."

In the other direction, suppose $R(x_0, \ldots, x_k)$ is computable. By definition, this means that the function $\chi_R(x_0, \ldots, x_k)$ is computable. By Theorem 35.2, χ_R is represented by a formula, say $\varphi_{\chi_R}(x_0, \ldots, x_k, y)$. Let $\varphi_R(x_0, \ldots, x_k)$ be

the formula $\varphi_{\chi_R}(x_0, \ldots, x_k, \overline{1})$. Then for any n_0, ..., n_k, if $R(n_0, \ldots, n_k)$ is true, then $\chi_R(n_0, \ldots, n_k) = 1$, in which case $\mathbf{Q}$ proves $\varphi_{\chi_R}(\overline{n_0}, \ldots, \overline{n_k}, \overline{1})$, and so $\mathbf{Q}$ proves $\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$. On the other hand, if $R(n_0, \ldots, n_k)$ is false, then $\chi_R(n_0, \ldots, n_k) = 0$. This means that $\mathbf{Q}$ proves

$$\forall y\, (\varphi_{\chi_R}(\overline{n_0}, \ldots, \overline{n_k}, y) \to y = \overline{0}).$$

Since $\mathbf{Q}$ proves $\overline{0} \neq \overline{1}$, $\mathbf{Q}$ proves $\neg\varphi_{\chi_R}(\overline{n_0}, \ldots, \overline{n_k}, \overline{1})$, and so it proves $\neg\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$.□

Problem 35.6. Show that if R is representable in $\mathbf{Q}$, so is χ_R.

---

content/incompleteness/representability-in-q/undecidability.tex

35.10 Undecidability

inc:req:und:sec

We call a theory $\mathbf{T}$ *undecidable* if there is no computational procedure which, after finitely many steps and unfailingly, provides a correct answer to the question "does $\mathbf{T}$ prove φ?" for any sentence φ in the language of $\mathbf{T}$. So $\mathbf{Q}$ would be decidable iff there were a computational procedure which decides, given a sentence φ in the language of arithmetic, whether $\mathbf{Q} \vdash \varphi$ or not. We can make this more precise by asking: Is the relation $\mathrm{Prov}_{\mathbf{Q}}(y)$, which holds of y iff y is the Gödel number of a sentence provable in $\mathbf{Q}$, recursive? The answer is: no.

Theorem 35.30. $\mathbf{Q}$ *is undecidable, i.e., the relation*

$$\mathrm{Prov}_{\mathbf{Q}}(y) \Leftrightarrow \mathrm{Sent}(y) \wedge \exists x\, \mathrm{Prf}_{\mathbf{Q}}(x, y)$$

is not recursive.

Proof. Suppose it were. Then we could solve the halting problem as follows: Given e and n, we know that $\varphi_e(n) \downarrow$ iff there is an s such that $T(e, n, s)$, where T is Kleene's predicate from Theorem 29.28. Since T is primitive recursive it is representable in $\mathbf{Q}$ by a formula ψ_T, that is, $\mathbf{Q} \vdash \psi_T(\overline{e}, \overline{n}, \overline{s})$ iff $T(e, n, s)$. If $\mathbf{Q} \vdash \psi_T(\overline{e}, \overline{n}, \overline{s})$ then also $\mathbf{Q} \vdash \exists y\, \psi_T(\overline{e}, \overline{n}, y)$. If no such s exists, then $\mathbf{Q} \vdash \neg\psi_T(\overline{e}, \overline{n}, \overline{s})$ for every s. But $\mathbf{Q}$ is ω-consistent, i.e., if $\mathbf{Q} \vdash \neg\varphi(\overline{n})$ for every $n \in \mathbb{N}$, then $\mathbf{Q} \nvdash \exists y\, \varphi(y)$. We know this because the axioms of $\mathbf{Q}$ are true in the standard model $\mathfrak{N}$. So, $\mathbf{Q} \nvdash \exists y\, \psi_T(\overline{e}, \overline{n}, y)$. In other words, $\mathbf{Q} \vdash \exists y\, \psi_T(\overline{e}, \overline{n}, y)$ iff there is an s such that $T(e, n, s)$, i.e., iff $\varphi_e(n) \downarrow$. From e and n we can compute ${}^{\#}\exists y\, \psi_T(\overline{e}, \overline{n}, y)^{\#}$, let $g(e, n)$ be the primitive recursive function which does that. So

$$h(e, n) = \begin{cases} 1 & \text{if } \mathrm{Prov}_{\mathbf{Q}}(g(e, n)) \\ 0 & \text{otherwise.} \end{cases}$$

This would show that h is recursive if $\mathrm{Prov}_{\mathbf{Q}}$ is. But h is not recursive, by Theorem 29.29, so $\mathrm{Prov}_{\mathbf{Q}}$ cannot be either. □

Corollary 35.31. *First-order logic is undecidable.*

Proof. If first-order logic were decidable, provability in $\mathbf{Q}$ would be as well, since $\mathbf{Q} \vdash \varphi$ iff $\vdash \tau \to \varphi$, where τ is the conjunction of the axioms of $\mathbf{Q}$. □

---

content/incompleteness/representability-in-q/sigma1-completeness.tex

35.11 Σ_1 completeness

inc:inp:s1c: sec

Despite the incompleteness of $\mathbf{Q}$ and its consistent, axiomatizable extensions, we have seen that $\mathbf{Q}$ does prove many basic facts about numerals. In fact, this can be extended quite considerably. To understand the scope of what can be proved in $\mathbf{Q}$, we introduce the notions of Δ_0, Σ_1, and Π_1 formulas. Roughly speaking, a Σ_1 formula is one of the form $\exists x\, \psi(x)$, where ψ is constructed using only propositional connectives and bounded quantifiers. We shall show that if φ is a Σ_1 sentence which is true in $\mathfrak{N}$, then $\mathbf{Q} \vdash \varphi$ (Theorem 35.38).

inc:inp:s1c: defn:bd-quant

Definition 35.32. A *bounded existential formula* is one of the form $\exists x\, (x < t \wedge \varphi(x))$ where t is any term, which we conventionally write as $(\exists x < t)\, \varphi(x)$. A *bounded universal formula* is one of the form $\forall x\, (x < t \to \varphi(x))$ where t is any term, which we conventionally write as $(\forall x < t)\, \varphi(x)$.

inc:inp:s1c: defn:delta0-sigma1-pi1-frm

Definition 35.33. A formula ψ is Δ_0 if it is built up from atomic formulas using only propositional connectives and bounded quantification. A formula φ is Σ_1 if $\varphi \equiv \exists x\, \psi(x)$ where ψ is Δ_0. A formula φ is Π_1 if $\varphi \equiv \forall x\, \psi(x)$ where ψ is Δ_0.

inc:inp:s1c: lem:q-proves-clterm-id

Lemma 35.34. *Suppose t is a closed term such that $\mathrm{Val}^{\mathfrak{N}}(t) = n$. Then $\mathbf{Q} \vdash t = \overline{n}$.*

Proof. We prove this by induction on the complexity of t. For the base case, $\mathrm{Val}^{\mathfrak{N}}(\mathsf{o}) = 0$, and $\mathbf{Q} \vdash \mathsf{o} = \overline{0}$ since $\overline{0} \equiv \mathsf{o}$. For the inductive case, let t_1 and t_2 be terms such that $\mathrm{Val}^{\mathfrak{N}}(t_1) = n_1$, $\mathrm{Val}^{\mathfrak{N}}(t_2) = n_2$, $\mathbf{Q} \vdash t_1 = \overline{n}_1$, and $\mathbf{Q} \vdash t_2 = \overline{n}_2$.

Then $\mathrm{Val}^{\mathfrak{N}}((t_1')) = n_1 + 1$, and we have that $\mathbf{Q} \vdash t_1' = \overline{n}_1{}'$ by the first-order rules for identity applied to the induction hypothesis and the formula $\overline{n_1}' = \overline{n_1}'$, so we have $\mathbf{Q} \vdash t_1' = \overline{n_1 + 1}$ by the definition of numerals.

For sums we have

$$\mathrm{Val}^{\mathfrak{N}}((t_1 + t_2)) = \mathrm{Val}^{\mathfrak{N}}(t_1) + \mathrm{Val}^{\mathfrak{N}}(t_2) = n_1 + n_2.$$

By the induction hypothesis and the rules for identity, $\mathbf{Q} \vdash t_1 + t_2 = \overline{n_1} + t_2$, and then $\mathbf{Q} \vdash t_1 + t_2 = \overline{n_1} + \overline{n_2}$ by a second application of the rules for identity. By Lemma 35.16, $\mathbf{Q} \vdash \overline{n_1} + \overline{n_2} = \overline{n_1 + n_2}$, so $\mathbf{Q} \vdash t_1 + t_2 = \overline{n_1 + n_2}$.

Similar reasoning also works for $\times$, using Lemma 35.18. Since this exhausts the closed terms of arithmetic, we have that $\mathbf{Q} \vdash t = \overline{n}$ for all closed terms t such that $\mathrm{Val}^{\mathfrak{N}}(t) = n$. □

Problem 35.7. Prove in detail the part of Lemma 35.34 involving $\times$.

Lemma 35.35. *Suppose t_1 and t_2 are closed terms. Then*

inc:inp:s1c: lem:atomic-completeness

1. *If* $\mathrm{Val}^{\mathfrak{N}}(t_1) = \mathrm{Val}^{\mathfrak{N}}(t_2)$, *then* $\mathbf{Q} \vdash t_1 = t_2$.
2. *If* $\mathrm{Val}^{\mathfrak{N}}(t_1) \neq \mathrm{Val}^{\mathfrak{N}}(t_2)$, *then* $\mathbf{Q} \vdash t_1 \neq t_2$.
3. *If* $\mathrm{Val}^{\mathfrak{N}}(t_1) < \mathrm{Val}^{\mathfrak{N}}(t_2)$, *then* $\mathbf{Q} \vdash t_1 < t_2$.
4. *If* $\mathrm{Val}^{\mathfrak{N}}(t_2) \leq \mathrm{Val}^{\mathfrak{N}}(t_1)$, *then* $\mathbf{Q} \vdash \neg(t_1 < t_2)$.

Proof. Given terms t_1 and t_2, we fix $n = \mathrm{Val}^{\mathfrak{N}}(t_1)$ and $m = \mathrm{Val}^{\mathfrak{N}}(t_2)$.

Suppose $\varphi \equiv t_1 = t_2$. By Lemma 35.34, $\mathbf{Q} \vdash t_1 = \overline{n}$ and $\mathbf{Q} \vdash t_2 = \overline{n}$. If $n = m$, then $\mathbf{Q} \vdash \overline{n} = \overline{m}$ and hence $\mathbf{Q} \vdash t_1 = t_2$ by the transitivity of identity. If $n \neq m$ then $\mathbf{Q} \vdash \overline{n} \neq \overline{m}$, and by the transitivity of identity again, $\mathbf{Q} \vdash t_1 \neq t_2$.

Now let $\varphi \equiv t_1 < t_2$. For both cases, we rely on axiom Q_8, which states that $x < y \leftrightarrow \exists z\, z' + x = y$ for all x, y.

Suppose $\mathfrak{N} \vDash t_1 < t_2$. Then there exists some $k \in \mathbb{N}$ such that $n + k + 1 = m$. By Lemma 35.34, $\mathbf{Q} \vdash t_1 = \overline{n}$ and $\mathbf{Q} \vdash t_2 = \overline{m}$, and by the first part of this lemma, $\mathbf{Q} \vdash \overline{n} + \overline{k}' = \overline{m}$. By the transitivity of identity it follows that $\mathbf{Q} \vdash \overline{k}' + t_1 = t_2$, so $\mathbf{Q} \vdash \exists z\, z' + t_1 = t_2$. By the right-to-left direction of Q_8, $\mathbf{Q} \vdash t_1 < t_2$.

Suppose instead that $\mathfrak{N} \nvDash t_1 < t_2$, i.e., $m \leq n$. We work in $\mathbf{Q}$ and assume that $t_1 < t_2$. By the left-to-right direction of Q_8, there is some z such that $z' + t_1 = t_2$. Since $\mathbf{Q} \vdash t_1 = \overline{n}$ and $\mathbf{Q} \vdash t_2 = \overline{m}$, $z' + \overline{n} = \overline{m}$. By an external induction on m using Q_5, $z' + \overline{n - m} = \mathsf{0}$. If $m = n$ then $z' \neq \mathsf{0}$, giving a contradiction via Q_3. If $m < n$ then $(z' + \overline{n - m - 1})' = \mathsf{0}$ by Q_5 again, giving a contradiction via Q_3. So $\mathbf{Q} \vdash \neg(t_1 < t_2)$. □

Lemma 35.36. *Suppose φ is a formula, t a closed term, and $k = \mathrm{Val}^{\mathfrak{N}}(t)$. Then*

inc:inp:s1c: lem:bounded-quant-equiv

1. $\mathbf{Q} \vdash (\forall x < t)\ \varphi(x)$ *iff* $\mathbf{Q} \vdash \varphi(\overline{0}) \wedge \cdots \wedge \varphi(\overline{k - 1})$.
2. $\mathbf{Q} \vdash (\exists x < t)\ \varphi(x)$ *iff* $\mathbf{Q} \vdash \varphi(\overline{0}) \vee \cdots \vee \varphi(\overline{k - 1})$.

Proof. We prove the case for the bounded universal quantifier. If $\mathrm{Val}^{\mathfrak{N}}(t) = 0$ then the left-hand side of the equivalence is provable in $\mathbf{Q}$, because there is no $x < \overline{0}$ by Lemma 35.23. Similarly, we can take an empty disjunction to be simply $\top$, which is also provable in $\mathbf{Q}$. We therefore suppose that $\mathrm{Val}^{\mathfrak{N}}(t) = k + 1$ for some natural number k. By Lemma 35.34 we can assume that we are working with a formula of the form $(\forall x < \overline{k + 1})\ \varphi(x)$.

Suppose that $\mathbf{Q} \vdash (\forall x < \overline{k + 1})\ \varphi(x)$, and let $n \leq k$. Since $\mathbf{Q} \vdash \overline{n} < \overline{k + 1}$ by Lemma 35.35, it follows by logic that $\mathbf{Q} \vdash \varphi(\overline{n})$. Applying this fact $k + 1$ times for each $n \leq k$, we get that $\mathbf{Q} \vdash \varphi(\overline{0}) \wedge \cdots \wedge \varphi(\overline{k})$ as desired.

For the other direction, suppose that $\mathbf{Q} \vdash \varphi(\overline{0}) \wedge \cdots \wedge \varphi(\overline{k})$. Working in $\mathbf{Q}$, suppose that $x < \overline{k+1}$. By Lemma 35.24 we have that $x = \overline{0} \vee \cdots \vee x = \overline{k}$, so by logic it follows that $\varphi(x)$, and hence the universal claim $(\forall x < \overline{k+1})\ \varphi(x)$ follows.

The proof of the equivalence for bounded existentially quantified formulas is similar. □

Problem 35.8. Give a detailed proof of the existential case in Lemma 35.36.

inc:inp:s1c: lem:delta0-completeness

Lemma 35.37. *If φ is a Δ_0 sentence which is true in $\mathfrak{N}$, then $\mathbf{Q} \vdash \varphi$.*

Proof. We prove this by induction on formula complexity. The base case is given by Lemma 35.35, so we move to the induction step. For simplicity we split the case of negation into subcases depending on the structure of the formula to which the negation is applied.

1. Suppose $(\varphi \wedge \psi)$ is true in $\mathfrak{N}$, so φ and ψ are true in $\mathfrak{N}$. By the induction hypothesis, $\mathbf{Q} \vdash \varphi$ and $\mathbf{Q} \vdash \psi$, so $\mathbf{Q} \vdash (\varphi \wedge \psi)$ by logic.

2. Suppose $\neg(\varphi \wedge \psi)$ is true in $\mathfrak{N}$, so either $\neg\varphi$ or $\neg\psi$ is true in $\mathfrak{N}$. Without loss of generality, suppose the former. By the induction hypothesis $\mathbf{Q} \vdash \neg\varphi$, and hence $\mathbf{Q} \vdash \neg(\varphi \wedge \psi)$ by logic.

3. Suppose $(\varphi \vee \psi)$ is true in $\mathfrak{N}$, so either φ is true in $\mathfrak{N}$ or ψ is true in $\mathfrak{N}$. Without loss of generality, suppose the former holds. By the induction hypothesis $\mathbf{Q} \vdash \varphi$, and hence $\mathbf{Q} \vdash (\varphi \vee \psi)$ by logic.

4. Suppose $\neg(\varphi \vee \psi)$ is true in $\mathfrak{N}$, so $\neg\varphi$ and $\neg\psi$ are true in $\mathfrak{N}$. Then $\mathbf{Q} \vdash \neg\varphi$ and $\mathbf{Q} \vdash \neg\psi$ by the induction hypothesis. Consequently, $\mathbf{Q} \vdash \neg(\varphi \vee \psi)$ by logic.

5. Suppose that $(\forall x < t)\ \varphi(x)$ is true in $\mathfrak{N}$, where t is a closed term and $k = \mathrm{Val}^{\mathfrak{N}}(t)$. By the induction hypothesis and logic, if $\varphi(\overline{n})$ is true in $\mathfrak{N}$ for all $n < \mathrm{Val}^{\mathfrak{N}}(t)$ then $\mathbf{Q} \vdash \varphi(\overline{0}) \wedge \cdots \wedge \varphi(\overline{k-1})$. By Lemma 35.36 it follows that $\mathbf{Q} \vdash (\forall x < t)\ \varphi(x)$.

6. The case for the bounded existential quantifier, where we have a sentence of the form $(\exists x < t)\ \varphi(x)$, is similar to that for the bounded universal quantifier.

7. Suppose that $\neg(\forall x < t)\ \varphi(x)$ is true in $\mathfrak{N}$, where t is a closed term. This sentence is equivalent to the sentence $(\exists x < t)\ \neg\varphi(x)$, with the equivalence derivable in $\mathbf{Q}$, so we may apply the reasoning for bounded existential quantifiers.

8. Similarly, suppose that $\neg(\exists x < t)\ \varphi(x)$ is true in $\mathfrak{N}$, where t is a closed term. This sentence is equivalent in $\mathbf{Q}$ to $(\forall x < t)\ \neg\varphi(x)$, and so we may apply the reasoning for bounded universal quantifiers.

9. Finally, suppose $\neg\varphi$ is true in $\mathfrak{N}$. The only cases remaining are when φ is atomic and when $\neg\varphi \equiv \neg\neg\psi$ for some Δ_0 sentence ψ. If φ is atomic then by Lemma 35.35, $\mathbf{Q} \vdash \neg\varphi$. If $\neg\varphi \equiv \neg\neg\psi$, then by logic it is provably equivalent in $\mathbf{Q}$ to ψ, which is true in $\mathfrak{N}$ since $\neg\varphi$ is true in $\mathfrak{N}$. By the induction hypothesis we therefore have that $\mathbf{Q} \vdash \neg\varphi$. □

Problem 35.9. Give a detailed proof of the existential case in Lemma 35.37.

Theorem 35.38. *If φ is a Σ_1 sentence which is true in $\mathfrak{N}$, then $\mathbf{Q} \vdash \varphi$.*

inc:inp:s1c:thm:sigma1-completeness

Proof. If $\exists x \varphi(x)$ is a Σ_1 sentence which is true in $\mathfrak{N}$, then there exists a natural number n and a variable assignment s such that $s(x) = n$ and $\mathfrak{N}, s \vDash \varphi(x)$. By standard facts about the satisfaction relation it follows that $\mathfrak{N} \vDash \varphi(\overline{n})$. But $\varphi(\overline{n})$ is a Δ_0 formula, so by Lemma 35.37 we have that $\mathbf{Q} \vdash \varphi(\overline{n})$, and hence by logic we also have that $\mathbf{Q} \vdash \exists x\, \varphi(x)$. □

This chapter depends on material in the chapter on computability theory, but can be left out if that hasn't been covered. It's currently a basic conversion of Jeremy Avigad's notes, has not been revised, and is missing exercises.

Chapter 36

Theories and Computability

content/incompleteness/theories-computability/introduction.tex

36.1 Introduction

inc:tcp:int:sec

This section should be rewritten.

We have the following:

1. A definition of what it means for a function to be representable in **Q** (Definition 35.1)
2. a definition of what it means for a relation to be representable in **Q** (Definition 35.28)
3. a theorem asserting that the representable functions of **Q** are exactly the computable ones (Theorem 35.2)
4. a theorem asserting that the representable relations of **Q** are exactly the computable ones Theorem 35.29)

A *theory* is a set of sentences that is deductively closed, that is, with the property that whenever T proves φ then φ is in T. It is probably best to think of a theory as being a collection of sentences, together with all the things that these sentences imply. From now on, we will use **Q** to refer to the *theory* consisting of the set of sentences derivable from the eight axioms in section 35.1. Remember that we can code formula of **Q** as numbers; if φ is such a formula, let ${}^{\#}\varphi^{\#}$ denote the number coding φ. Modulo this coding, we can now ask whether various sets of formulas are computable or not.

---

content/incompleteness/theories-computability/q-is-ce.tex

36.2 Q is C.e.-Complete

inc:tcp:qce:sec

Theorem 36.1. **Q** *is c.e. but not decidable. In fact, it is a complete c.e. set.*

Proof. It is not hard to see that **Q** is c.e., since it is the set of (codes for) sentences y such that there is a proof x of y in **Q**:

$$Q = \{y : \exists x\, \mathrm{Prf}_{\mathbf{Q}}(x, y)\}.$$

But we know that $\mathrm{Prf}_{\mathbf{Q}}(x, y)$ is computable (in fact, primitive recursive), and any set that can be written in the above form is c.e.

Saying that it is a complete c.e. set is equivalent to saying that $K \leq_m Q$, where $K = \{x : \varphi_x(x)\downarrow\}$. So let us show that K is reducible to **Q**. Since Kleene's predicate $T(e, x, s)$ is primitive recursive, it is representable in **Q**, say, by φ_T. Then for every x, we have

$$\begin{aligned} x \in K &\to \exists s\, T(x, x, s) \\ &\to \exists s\, (\mathbf{Q} \vdash \varphi_T(\overline{x}, \overline{x}, \overline{s})) \\ &\to \mathbf{Q} \vdash \exists s\, \varphi_T(\overline{x}, \overline{x}, s). \end{aligned}$$

Conversely, if $\mathbf{Q} \vdash \exists s\, \varphi_T(\overline{x}, \overline{x}, s)$, then, in fact, for some natural number n the formula $\varphi_T(\overline{x}, \overline{x}, \overline{n})$ must be true. Now, if $T(x, x, n)$ were false, **Q** would prove $\neg\varphi_T(\overline{x}, \overline{x}, \overline{n})$, since φ_T represents T. But then **Q** proves a false formula, which is a contradiction. So $T(x, x, n)$ must be true, which implies $\varphi_x(x)\downarrow$.

In short, we have that for every x, x is in K if and only if **Q** proves $\exists s\, T(\overline{x}, \overline{x}, s)$. So the function f which takes x to (a code for) the sentence $\exists s\, T(\overline{x}, \overline{x}, s)$ is a reduction of K to **Q**. □

---

content/incompleteness/theories-computability/oconsis-ext-of-q-undec.tex

36.3 ω-Consistent Extensions of Q are Undecidable

explanation

The proof that **Q** is c.e.-complete relied on the fact that any sentence provable in **Q** is "true" of the natural numbers. The next definition and theorem strengthen this theorem, by pinpointing just those aspects of "truth" that were needed in the proof above. Don't dwell on this theorem too long, though, because we will soon strengthen it even further. We include it mainly for historical purposes: Gödel's original paper used the notion of ω-consistency, but his result was strengthened by replacing ω-consistency with ordinary consistency soon after.

inc:tcp:oqn:sec

Definition 36.2. A theory **T** is ω-consistent if the following holds: if $\exists x\, \varphi(x)$ is any sentence and **T** proves $\neg\varphi(\overline{0})$, $\neg\varphi(\overline{1})$, $\neg\varphi(\overline{2})$, ... then **T** does not prove $\exists x\, \varphi(x)$.

inc:tcp:oqn:thm:oconsis-q

Theorem 36.3. *Let* **T** *be any ω-consistent theory that includes* **Q**. *Then* **T** *is not decidable.*

Proof. If **T** includes **Q**, then **T** represents the computable functions and relations. We need only modify the previous proof. As above, if $x \in K$, then **T** proves $\exists s\, \varphi_T(\overline{x}, \overline{x}, s)$. Conversely, suppose **T** proves $\exists s\, \varphi_T(\overline{x}, \overline{x}, s)$. Then x must be in K: otherwise, there is no halting computation of machine x on input x; since φ_T represents Kleene's T relation, **T** proves $\neg\varphi_T(\overline{x}, \overline{x}, \overline{0})$, $\neg\varphi_T(\overline{x}, \overline{x}, \overline{1})$, ..., making **T** ω-inconsistent. □

---

content/incompleteness/theories-computability/extensions-of-q-not-decidable.tex

36.4 Consistent Extensions of Q are Undecidable

explanation

Remember that a theory is *consistent* if it does not prove both φ and $\neg\varphi$ for any formula φ. Since anything follows from a contradiction, an inconsistent theory is trivial: every sentence is provable. Clearly, if a theory if ω-consistent, then it is consistent. But being consistent is a weaker requirement (i.e., there are theories that are consistent but not ω-consistent.). We can weaken the assumption in Definition 36.2 to simple consistency to obtain a stronger theorem.

inc:tcp:cqn:sec

Lemma 36.4. *There is no "universal computable relation." That is, there is no binary computable relation $R(x, y)$, with the following property: whenever*

$S(y)$ is a unary computable relation, there is some k such that for every y, $S(y)$ is true if and only if $R(k, y)$ is true.

Proof. Suppose $R(x, y)$ is a universal computable relation. Let $S(y)$ be the relation $\neg R(y, y)$. Since $S(y)$ is computable, for some k, $S(y)$ is equivalent to $R(k, y)$. But then we have that $S(k)$ is equivalent to both $R(k, k)$ and $\neg R(k, k)$, which is a contradiction. □

Theorem 36.5. *Let* **T** *be any consistent theory that includes* **Q**. *Then* **T** *is not decidable.*

Proof. Suppose **T** is a consistent, decidable extension of **Q**. We will obtain a contradiction by using **T** to define a universal computable relation.

Let $R(x, y)$ hold if and only if

x codes a formula $\theta(u)$, and **T** proves $\theta(\overline{y})$.

Since we are assuming that **T** is decidable, R is computable. Let us show that R is universal. If $S(y)$ is any computable relation, then it is representable in **Q** (and hence **T**) by a formula $\theta_S(u)$. Then for every n, we have

$$\begin{aligned} S(\overline{n}) &\to T \vdash \theta_S(\overline{n}) \\ &\to R(^{\#}\theta_S(u)^{\#}, n) \end{aligned}$$

and

$$\begin{aligned} \neg S(\overline{n}) &\to T \vdash \neg\theta_S(\overline{n}) \\ &\to T \nvdash \theta_S(\overline{n}) \quad \text{(since } \mathbf{T} \text{ is consistent)} \\ &\to \neg R(^{\#}\theta_S(u)^{\#}, n). \end{aligned}$$

That is, for every y, $S(y)$ is true if and only if $R(^{\#}\theta_S(u)^{\#}, y)$ is. So R is universal, and we have the contradiction we were looking for. □

Let "true arithmetic" be the theory $\{\varphi : N \vDash \varphi\}$, that is, the set of sentences in the language of arithmetic that are true in the standard interpretation.

Corollary 36.6. *True arithmetic is not decidable.*

---

content/incompleteness/theories-computability/computably-axiomatizable.tex

36.5 Axiomatizable Theories

inc:tcp:cax:sec

A theory **T** is said to be *axiomatizable* if it has a computable set of axioms A. (Saying that A is a set of axioms for **T** means $T = \{\varphi : A \vdash \varphi\}$.) Any "reasonable" axiomatization of the natural numbers will have this property. In particular, any theory with a finite set of axioms is axiomatizable.

Lemma 36.7. *Suppose* **T** *is axiomatizable. Then* **T** *is computably enumerable.*

Proof. Suppose A is a computable set of axioms for **T**. To determine if $\varphi \in T$, just search for a derivation of φ from the axioms.

Put slightly differently, φ is in **T** if and only if there is a finite list of axioms $\psi_1, \ldots, \psi_k$ in A and a derivation of $(\psi_1 \wedge \cdots \wedge \psi_k) \to \varphi$ in first-order logic. But we already know that any set with a definition of the form "there exists ... such that ..." is c.e., provided the second "..." is computable. □

---

content/incompleteness/theories-computability/complete-decidable.tex

36.6 Axiomatizable Complete Theories are Decidable

A theory is said to be *complete* if for every sentence φ, either φ or $\neg\varphi$ is provable. inc:tcp:cdc:sec

Lemma 36.8. *Suppose a theory* **T** *is complete and axiomatizable. Then* **T** *is decidable.*

Proof. Suppose **T** is complete and A is a computable set of axioms. If **T** is inconsistent, it is clearly computable. (Algorithm: "just say yes.") So we can assume that **T** is also consistent.

To decide whether or not a sentence φ is in **T**, simultaneously search for a derivation of φ from **T** and a derivation of $\neg\varphi$. Since **T** is complete, you are bound to find one or the other; and since **T** is consistent, if you find a derivation of $\neg\varphi$, there is no derivation of φ.

Put in different terms, we already know that **T** is c.e.; so by a theorem we proved before, it suffices to show that the complement of **T** is c.e. also. But a formula φ is in $\bar{\mathbf{T}}$ if and only if $\neg\varphi$ is in **T**; so $\bar{\mathbf{T}} \leq_m \mathbf{T}$. □

---

content/incompleteness/theories-computability/first-incompleteness.tex

36.7 Q has no Complete, Consistent, Axiomatizable Extensions

Theorem 36.9. *There is no complete, consistent, axiomatizable extension of* **Q**. inc:tcp:inc:sec inc:tcp:inc:thm:first-incompleteness

Proof. We already know that there is no consistent, decidable extension of **Q**. But if **T** is complete and axiomatized, then it is decidable. □

This theorems is not that far from Gödel's original 1931 formulation of the First Incompleteness Theorem. Aside from the more modern terminology, the key differences are this: Gödel has "ω-consistent" instead of "consistent"; and he could not say "axiomatizable" in full generality, since the formal notion of computability was not in place yet. (The formal models of computability were developed over the following decade, including by Gödel, and in large part to be able to characterize the kinds of theories that are susceptible to the Gödel phenomenon.) explanation

The theorem says you can't have it all, namely, completeness, consistency, and axiomatizability. If you give up any one of these, though, you can have the other two: **Q** is consistent and computably axiomatized, but not complete; the inconsistent theory is complete, and computably axiomatized (say, by $\{0 \neq 0\}$), but not consistent; and the set of true sentence of arithmetic is complete and consistent, but it is not computably axiomatized.

---

content/incompleteness/theories-computability/inseparability.tex

36.8 Sentences Provable and Refutable in Q are Computably Inseparable

inc:tcp:ins:sec

Let $\bar{\mathbf{Q}}$ be the set of sentences whose *negations* are provable in **Q**, i.e., $\bar{\mathbf{Q}} = \{\varphi : \mathbf{Q} \vdash \neg\varphi\}$. Remember that disjoint sets A and B are said to be computably inseparable if there is no computable set C such that $A \subseteq C$ and $B \subseteq \overline{C}$.

Lemma 36.10. **Q** *and* $\bar{\mathbf{Q}}$ *are computably inseparable.*

Proof. Suppose C is a computable set such that $\mathbf{Q} \subseteq C$ and $\bar{\mathbf{Q}} \subseteq \overline{C}$. Let $R(x, y)$ be the relation

> x codes a formula $\theta(u)$ and $\theta(\overline{y})$ is in C.

We will show that $R(x, y)$ is a universal computable relation, yielding a contradiction.

Suppose $S(y)$ is computable, represented by $\theta_S(u)$ in **Q**. Then

$$\begin{aligned} S(\overline{n}) &\rightarrow \mathbf{Q} \vdash \theta_S(\overline{n}) \\ &\rightarrow \theta_S(\overline{n}) \in C \end{aligned}$$

and

$$\begin{aligned} \neg S(\overline{n}) &\rightarrow \mathbf{Q} \vdash \neg\theta_S(\overline{n}) \\ &\rightarrow \theta_S(\overline{n}) \in \bar{\mathbf{Q}} \\ &\rightarrow \theta_S(\overline{n}) \notin C \end{aligned}$$

So $S(y)$ is equivalent to $R(\#(\theta_S(\overline{u})), y)$. □

---

content/incompleteness/theories-computability/consis-with-q.tex

36.9 Theories Consistent with Q are Undecidable

inc:tcp:con:sec

The following theorem says that not only is **Q** undecidable, but, in fact, any theory that does not disagree with **Q** is undecidable.

Theorem 36.11. *Let* **T** *be any theory in the language of arithmetic that is consistent with* **Q** *(i.e.,* $\mathbf{T} \cup \mathbf{Q}$ *is consistent). Then* **T** *is undecidable.*

Proof. Remember that **Q** has a finite set of axioms, $Q_1, \ldots, Q_8$. We can even replace these by a single axiom, $\alpha = Q_1 \wedge \cdots \wedge Q_8$.

Suppose **T** is a decidable theory consistent with **Q**. Let

$$C = \{\varphi : \mathbf{T} \vdash \alpha \to \varphi\}.$$

We show that C would be a computable separation of **Q** and $\bar{\mathbf{Q}}$, a contradiction. First, if φ is in **Q**, then φ is provable from the axioms of **Q**; by the deduction theorem, there is a derivation of $\alpha \to \varphi$ in first-order logic. So φ is in C.

On the other hand, if φ is in $\bar{\mathbf{Q}}$, then there is a proof of $\alpha \to \neg\varphi$ in first-order logic. If **T** also proves $\alpha \to \varphi$, then **T** proves $\neg\alpha$, in which case $\mathbf{T} \cup \mathbf{Q}$ is inconsistent. But we are assuming $\mathbf{T} \cup \mathbf{Q}$ is consistent, so **T** does not prove $\alpha \to \varphi$, and so φ is not in C.

We've shown that if φ is in **Q**, then it is in C, and if φ is in $\bar{\mathbf{Q}}$, then it is in $\overline{C}$. So C is a computable separation, which is the contradiction we were looking for. □

This theorem is very powerful. For example, it implies:

Corollary 36.12. *First-order logic for the language of arithmetic (that is, the set* $\{\varphi : \varphi$ *is provable in first-order logic*$\}$*) is undecidable.*

Proof. First-order logic is the set of consequences of $\emptyset$, which is consistent with **Q**. □

---

content/incompleteness/theories-computability/interpretability.tex

36.10 Theories in which Q is Interpretable are Undecidable

inc:tcp:itp:sec

We can strengthen these results even more. Informally, an interpretation of a language $\mathcal{L}_1$ in another language $\mathcal{L}_2$ involves defining the universe, relation symbols, and function symbols of $\mathcal{L}_1$ with formulas in $\mathcal{L}_2$. Though we won't take the time to do this, one can make this definition precise.

Theorem 36.13. *Suppose* **T** *is a theory in a language in which one can interpret the language of arithmetic, in such a way that* **T** *is consistent with the interpretation of* **Q**. *Then* **T** *is undecidable. If* **T** *proves the interpretation of the axioms of* **Q**, *then no consistent extension of* **T** *is decidable.*

The proof is just a small modification of the proof of the last theorem; one could use a counterexample to get a separation of **Q** and $\bar{\mathbf{Q}}$. One can take **ZFC**, Zermelo–Fraenkel set theory with the axiom of choice, to be an axiomatic foundation that is powerful enough to carry out a good deal of ordinary mathematics. In **ZFC** one can define the natural numbers, and via this interpretation, the axioms of **Q** are true. So we have

Corollary 36.14. *There is no decidable extension of* **ZFC**.

Corollary 36.15. *There is no complete, consistent, computably axiomatizable extension of* **ZFC**.

The language of **ZFC** has only a single binary relation, $\in$. (In fact, you don't even need equality.) So we have

Corollary 36.16. *First-order logic for any language with a binary relation symbol is undecidable.*

This result extends to any language with two unary function symbols, since one can use these to simulate a binary relation symbol. The results just cited are tight: it turns out that first-order logic for a language with only *unary* relation symbols and at most one *unary* function symbol is decidable.

One more bit of trivia. We know that the set of sentences in the language $\mathsf{0}$, $'$, $+$, $\times$, $<$ true in the standard model is undecidable. In fact, one can define $<$ in terms of the other symbols, and then one can define $+$ in terms of $\times$ and $'$. So the set of true sentences in the language $\mathsf{0}$, $'$, $\times$ is undecidable. On the other hand, Presburger has shown that the set of sentences in the language $\mathsf{0}$, $'$, $+$ true in the language of arithmetic is decidable. The procedure is computationally infeasible, however.

Chapter 37

Incompleteness and Provability

---

content/incompleteness/incompleteness-provability/introduction.tex

37.1 Introduction

inc:inp:int: sec

Hilbert thought that a system of axioms for a mathematical structure, such as the natural numbers, is inadequate unless it allows one to derive all true statements about the structure. Combined with his later interest in formal systems of deduction, this suggests that he thought that we should guarantee that, say, the formal systems we are using to reason about the natural numbers is not only consistent, but also *complete*, i.e., every statement in its language is either derivable or its negation is. Gödel's first incompleteness theorem shows that no such system of axioms exists: there is no complete, consistent, axiomatizable formal system for arithmetic. In fact, no "sufficiently strong," consistent, axiomatizable mathematical theory is complete.

A more important goal of Hilbert's, the centerpiece of his program for the justification of modern ("classical") mathematics, was to find finitary consistency proofs for formal systems representing classical reasoning. With regard to Hilbert's program, then, Gödel's second incompleteness theorem was a much bigger blow. The second incompleteness theorem can be stated in vague terms, like the first incompleteness theorem. Roughly speaking, it says that no sufficiently strong theory of arithmetic can prove its own consistency. We will have to take "sufficiently strong" to include a little bit more than **Q**.

The idea behind Gödel's original proof of the incompleteness theorem can be found in the Epimenides paradox. Epimenides, a Cretan, asserted that all Cretans are liars; a more direct form of the paradox is the assertion "this sentence is false." Essentially, by replacing truth with derivability, Gödel was able to formalize a sentence which, in a roundabout way, asserts that it itself is not derivable. If that sentence were derivable, the theory would then be inconsistent. Gödel showed that the negation of that sentence is also not derivable from the system of axioms he was considering. (For this second part, Gödel had to assume that the theory **T** is what's called "ω-consistent." ω-Consistency is related to consistency, but is a stronger property.[1] A few years after Gödel, Rosser showed that assuming simple consistency of **T** is enough.)

The first challenge is to understand how one can construct a sentence that refers to itself. For every formula φ in the language of **Q**, let $\ulcorner\varphi\urcorner$ denote the numeral corresponding to ${}^{\#}\varphi^{\#}$. Think about what this means: φ is a formula in the language of **Q**, ${}^{\#}\varphi^{\#}$ is a natural number, and $\ulcorner\varphi\urcorner$ is a *term* in the language of **Q**. So every formula φ in the language of **Q** has a *name*, $\ulcorner\varphi\urcorner$, which is a term in the language of **Q**; this provides us with a conceptual framework in which formulas in the language of **Q** can "say" things about other formulas. The following lemma is known as the fixed-point lemma.

Lemma 37.1. *Let* **T** *be any theory extending* **Q**, *and let* $\psi(x)$ *be any formula with only the variable* x *free. Then there is a sentence* φ *such that* $\mathbf{T} \vdash \varphi \leftrightarrow \psi(\ulcorner\varphi\urcorner)$.

[1] That is, any ω-consistent theory is consistent, but not vice versa.

The lemma asserts that given any property $\psi(x)$, there is a sentence φ that asserts "$\psi(x)$ is true of me," and $\mathbf{T}$ "knows" this.

How can we construct such a sentence? Consider the following version of the Epimenides paradox, due to Quine:

> "Yields falsehood when preceded by its quotation" yields falsehood when preceded by its quotation.

This sentence is not directly self-referential. It simply makes an assertion about the syntactic objects between quotes, and, in doing so, it is on par with sentences like

1. "Robert" is a nice name.
2. "I ran." is a short sentence.
3. "Has three words" has three words.

But what happens when one takes the phrase "yields falsehood when preceded by its quotation," and precedes it with a quoted version of itself? Then one has the original sentence! In short, the sentence asserts that it is false.

---

content/incompleteness/incompleteness-provability/fixed-point-lemma.tex

37.2 The Fixed-Point Lemma

inc:inp:fix:sec

The fixed-point lemma says that for any formula $\psi(x)$, there is a sentence φ such that $\mathbf{T} \vdash \varphi \leftrightarrow \psi(\ulcorner\varphi\urcorner)$, provided $\mathbf{T}$ extends $\mathbf{Q}$. In the case of the liar sentence, we'd want φ to be equivalent (provably in $\mathbf{T}$) to "$\ulcorner\varphi\urcorner$ is false," i.e., the statement that ${}^{\#}\varphi^{\#}$ is the Gödel number of a false sentence. To understand the idea of the proof, it will be useful to compare it with Quine's informal gloss of φ as, "'yields a falsehood when preceded by its own quotation' yields a falsehood when preceded by its own quotation." The operation of taking an expression, and then forming a sentence by preceding this expression by its own quotation may be called *diagonalizing* the expression, and the result its diagonalization. So, the diagonalization of 'yields a falsehood when preceded by its own quotation' is "'yields a falsehood when preceded by its own quotation' yields a falsehood when preceded by its own quotation." Now note that Quine's liar sentence is not the diagonalization of 'yields a falsehood' but of 'yields a falsehood when preceded by its own quotation.' So the property being diagonalized to yield the liar sentence itself involves diagonalization!

explanation

In the language of arithmetic, we form quotations of a formula with one free variable by computing its Gödel numbers and then substituting the standard numeral for that Gödel number into the free variable. The diagonalization of $\alpha(x)$ is $\alpha(\overline{n})$, where $n = {}^{\#}\alpha(x)^{\#}$. (From now on, let's abbreviate $\overline{{}^{\#}\alpha(x)^{\#}}$ as $\ulcorner\alpha(x)\urcorner$.) So if $\psi(x)$ is "is a falsehood," then "yields a falsehood if preceded by its own quotation," would be "yields a falsehood when applied to the Gödel number of its diagonalization." If we had a symbol diag for the function $\mathrm{diag}(n)$

which computes the Gödel number of the diagonalization of the formula with Gödel number n, we could write $\alpha(x)$ as $\psi(\mathit{diag}(x))$. And Quine's version of the liar sentence would then be the diagonalization of it, i.e., $\alpha(\ulcorner\alpha(x)\urcorner)$ or $\psi(\mathit{diag}(\ulcorner\psi(\mathit{diag}(x))\urcorner))$. Of course, $\psi(x)$ could now be any other property, and the same construction would work. For the incompleteness theorem, we'll take $\psi(x)$ to be "x is not derivable in $\mathbf{T}$." Then $\alpha(x)$ would be "yields a sentence not derivable in $\mathbf{T}$ when applied to the Gödel number of its diagonalization."

To formalize this in $\mathbf{T}$, we have to find a way to formalize diag. The function $\mathrm{diag}(n)$ is computable, in fact, it is primitive recursive: if n is the Gödel number of a formula $\alpha(x)$, $\mathrm{diag}(n)$ returns the Gödel number of $\alpha(\ulcorner\alpha(x)\urcorner)$. (Recall, $\ulcorner\alpha(x)\urcorner$ is the standard numeral of the Gödel number of $\alpha(x)$, i.e., $\overline{\alpha(x)^\#}$). If diag were a function symbol in $\mathbf{T}$ representing the function diag, we could take φ to be the formula $\psi(\mathit{diag}(\ulcorner\psi(\mathit{diag}(x))\urcorner))$. Notice that

$$\begin{aligned}\mathrm{diag}(\overline{\psi(\mathit{diag}(x))^\#}) &= \overline{\psi(\mathit{diag}(\ulcorner\psi(\mathit{diag}(x))\urcorner))^\#}\\ &= \overline{\varphi^\#}.\end{aligned}$$

Assuming $\mathbf{T}$ can derive

$$\mathit{diag}(\ulcorner\psi(\mathit{diag}(x))\urcorner) = \ulcorner\varphi\urcorner,$$

it can derive $\psi(\mathit{diag}(\ulcorner\psi(\mathit{diag}(x))\urcorner)) \leftrightarrow \psi(\ulcorner\varphi\urcorner)$. But the left hand side is, by definition, φ.

Of course, diag will in general not be a function symbol of $\mathbf{T}$, and certainly is not one of $\mathbf{Q}$. But, since diag is computable, it is *representable* in $\mathbf{Q}$ by some formula $\theta_{\mathrm{diag}}(x, y)$. So instead of writing $\psi(\mathit{diag}(x))$ we can write $\exists y\,(\theta_{\mathrm{diag}}(x, y) \wedge \psi(y))$. Otherwise, the proof sketched above goes through, and in fact, it goes through already in $\mathbf{Q}$.

Lemma 37.2. *Let $\psi(x)$ be any formula with one free variable x. Then there is a sentence φ such that $\mathbf{Q} \vdash \varphi \leftrightarrow \psi(\ulcorner\varphi\urcorner)$.* inc:inp:fix: lem:fixed-point

Proof. Given $\psi(x)$, let $\alpha(x)$ be the formula $\exists y\,(\theta_{\mathrm{diag}}(x, y) \wedge \psi(y))$ and let φ be its diagonalization, i.e., the formula $\alpha(\ulcorner\alpha(x)\urcorner)$.

Since θ_{diag} represents diag, and $\mathrm{diag}(\overline{\alpha(x)^\#}) = \overline{\varphi^\#}$, $\mathbf{Q}$ can derive

$$\theta_{\mathrm{diag}}(\ulcorner\alpha(x)\urcorner, \ulcorner\varphi\urcorner) \tag{37.1}$$ inc:inp:fix: repdiag1

$$\forall y\,(\theta_{\mathrm{diag}}(\ulcorner\alpha(x)\urcorner, y) \to y = \ulcorner\varphi\urcorner). \tag{37.2}$$ inc:inp:fix: repdiag2

Now we show that $\mathbf{Q} \vdash \varphi \leftrightarrow \psi(\ulcorner\varphi\urcorner)$. We argue informally, using just logic and facts derivable in $\mathbf{Q}$.

First, suppose φ, i.e., $\alpha(\ulcorner\alpha(x)\urcorner)$. Going back to the definition of $\alpha(x)$, we see that $\alpha(\ulcorner\alpha(x)\urcorner)$ just is

$$\exists y\,(\theta_{\mathrm{diag}}(\ulcorner\alpha(x)\urcorner, y) \wedge \psi(y)).$$

Consider such a y. Since $\theta_{\mathrm{diag}}(\ulcorner\alpha(x)\urcorner, y)$, by eq. (37.2), $y = \ulcorner\varphi\urcorner$. So, from $\psi(y)$ we have $\psi(\ulcorner\varphi\urcorner)$.

Now suppose $\psi(\ulcorner\varphi\urcorner)$. By eq. (37.1), we have

$$\theta_{\text{diag}}(\ulcorner\alpha(x)\urcorner, \ulcorner\varphi\urcorner) \wedge \psi(\ulcorner\varphi\urcorner).$$

It follows that

$$\exists y\, (\theta_{\text{diag}}(\ulcorner\alpha(x)\urcorner, y) \wedge \psi(y)).$$

But that's just $\alpha(\ulcorner\alpha(x)\urcorner)$, i.e., φ. □

digression

You should compare this to the proof of the fixed-point lemma in computability theory. The difference is that here we want to define a *statement* in terms of itself, whereas there we wanted to define a *function* in terms of itself; this difference aside, it is really the same idea.

Problem 37.1. A formula $\varphi(x)$ is a *truth definition* if $\mathbf{Q} \vdash \psi \leftrightarrow \varphi(\ulcorner\psi\urcorner)$ for all sentences ψ. Show that no formula is a truth definition by using the fixed-point lemma.

---

content/incompleteness/incompleteness-provability/first-incompleteness-thm.tex

37.3 The First Incompleteness Theorem

inc:inp:1in:sec

We can now describe Gödel's original proof of the first incompleteness theorem. Let $\mathbf{T}$ be any computably axiomatized theory in a language extending the language of arithmetic, such that $\mathbf{T}$ includes the axioms of $\mathbf{Q}$. This means that, in particular, $\mathbf{T}$ represents computable functions and relations.

We have argued that, given a reasonable coding of formulas and proofs as numbers, the relation $\text{Prf}_{\mathbf{T}}(x, y)$ is computable, where $\text{Prf}_{\mathbf{T}}(x, y)$ holds if and only if x is the Gödel number of a derivation of the formula with Gödel number y in $\mathbf{T}$. In fact, for the particular theory that Gödel had in mind, Gödel was able to show that this relation is primitive recursive, using the list of 45 functions and relations in his paper. The 45th relation, xBy, is just $\text{Prf}_{\mathbf{T}}(x, y)$ for his particular choice of $\mathbf{T}$. Remember that where Gödel uses the word "recursive" in his paper, we would now use the phrase "primitive recursive."

Since $\text{Prf}_{\mathbf{T}}(x, y)$ is computable, it is representable in $\mathbf{T}$. We will use $\mathsf{Prf}_{\mathbf{T}}(x, y)$ to refer to the formula that represents it. Let $\mathsf{Prov}_{\mathbf{T}}(y)$ be the formula $\exists x\, \mathsf{Prf}_{\mathbf{T}}(x, y)$. This describes the 46th relation, $\text{Bew}(y)$, on Gödel's list. As Gödel notes, this is the only relation that "cannot be asserted to be recursive." What he probably meant is this: from the definition, it is not clear that it is computable; and later developments, in fact, show that it isn't.

Let $\mathbf{T}$ be an axiomatizable theory containing $\mathbf{Q}$. Then $\text{Prf}_{\mathbf{T}}(x, y)$ is decidable, hence representable in $\mathbf{Q}$ by a formula $\mathsf{Prf}_{\mathbf{T}}(x, y)$. Let $\mathsf{Prov}_{\mathbf{T}}(y)$ be the formula we described above. By the fixed-point lemma, there is a formula $\gamma_{\mathbf{T}}$ such that $\mathbf{Q}$ (and hence $\mathbf{T}$) derives

inc:inp:1in:eqn:qpf

$$\gamma_{\mathbf{T}} \leftrightarrow \neg\mathsf{Prov}_{\mathbf{T}}(\ulcorner\gamma_{\mathbf{T}}\urcorner). \tag{37.3}$$

Note that $\gamma_{\mathbf{T}}$ says, in essence, "$\gamma_{\mathbf{T}}$ is not derivable in $\mathbf{T}$."

Lemma 37.3. *If $\mathbf{T}$ is a consistent, axiomatizable theory extending $\mathbf{Q}$, then $\mathbf{T} \nvdash \gamma_{\mathbf{T}}$.* inc:inp:1in: lem:cons-G-unprov

Proof. Suppose $\mathbf{T}$ derives $\gamma_{\mathbf{T}}$. Then there *is* a derivation, and so, for some number m, the relation $\mathsf{Prf}_{\mathbf{T}}(m, {}^{\ulcorner}\gamma_{\mathbf{T}}{}^{\urcorner})$ holds. But then $\mathbf{Q}$ derives the sentence $\mathsf{Prf}_{\mathbf{T}}(\overline{m}, {}^{\ulcorner}\gamma_{\mathbf{T}}{}^{\urcorner})$. So $\mathbf{Q}$ derives $\exists x\, \mathsf{Prf}_{\mathbf{T}}(x, {}^{\ulcorner}\gamma_{\mathbf{T}}{}^{\urcorner})$, which is, by definition, $\mathsf{Prov}_{\mathbf{T}}({}^{\ulcorner}\gamma_{\mathbf{T}}{}^{\urcorner})$. By eq. (37.3), $\mathbf{Q}$ derives $\neg\gamma_{\mathbf{T}}$, and since $\mathbf{T}$ extends $\mathbf{Q}$, so does $\mathbf{T}$. We have shown that if $\mathbf{T}$ derives $\gamma_{\mathbf{T}}$, then it also derives $\neg\gamma_{\mathbf{T}}$, and hence it would be inconsistent. □

Definition 37.4. A theory $\mathbf{T}$ is *ω-consistent* if the following holds: if $\exists x\, \varphi(x)$ is any sentence and $\mathbf{T}$ derives $\neg\varphi(\overline{0})$, $\neg\varphi(\overline{1})$, $\neg\varphi(\overline{2})$, ... then $\mathbf{T}$ does not prove $\exists x\, \varphi(x)$. inc:inp:1in: thm:oconsis-q

Note that every ω-consistent theory is also consistent. This follows simply from the fact that if $\mathbf{T}$ is inconsistent, then $\mathbf{T} \vdash \varphi$ for every φ. In particular, if $\mathbf{T}$ is inconsistent, it derives both $\neg\varphi(\overline{n})$ for every n and also derives $\exists x\, \varphi(x)$. So, if $\mathbf{T}$ is inconsistent, it is ω-inconsistent. By contraposition, if $\mathbf{T}$ is ω-consistent, it must be consistent.

Lemma 37.5. *If $\mathbf{T}$ is an ω-consistent, axiomatizable theory extending $\mathbf{Q}$, then $\mathbf{T} \nvdash \neg\gamma_{\mathbf{T}}$.* inc:inp:1in: lem:omega-cons-G-unref

Proof. We show that if $\mathbf{T}$ derives $\neg\gamma_{\mathbf{T}}$, then it is ω-inconsistent. Suppose $\mathbf{T}$ derives $\neg\gamma_{\mathbf{T}}$. If $\mathbf{T}$ is inconsistent, it is ω-inconsistent, and we are done. Otherwise, $\mathbf{T}$ is consistent, so it does not derive $\gamma_{\mathbf{T}}$ by Lemma 37.3. Since there is no derivation of $\gamma_{\mathbf{T}}$ in $\mathbf{T}$, $\mathbf{Q}$ derives

$$\neg\mathsf{Prf}_{\mathbf{T}}(\overline{0}, {}^{\ulcorner}\gamma_{\mathbf{T}}{}^{\urcorner}), \neg\mathsf{Prf}_{\mathbf{T}}(\overline{1}, {}^{\ulcorner}\gamma_{\mathbf{T}}{}^{\urcorner}), \neg\mathsf{Prf}_{\mathbf{T}}(\overline{2}, {}^{\ulcorner}\gamma_{\mathbf{T}}{}^{\urcorner}), \ldots$$

and so does $\mathbf{T}$. On the other hand, by eq. (37.3), $\neg\gamma_{\mathbf{T}}$ is equivalent to $\exists x\, \mathsf{Prf}_{\mathbf{T}}(x, {}^{\ulcorner}\gamma_{\mathbf{T}}{}^{\urcorner})$. So $\mathbf{T}$ is ω-inconsistent. □

Problem 37.2. Every ω-consistent theory is consistent. Show that the converse does not hold, i.e., that there are consistent but ω-inconsistent theories. Do this by showing that $\mathbf{Q} \cup \{\neg\gamma_{\mathbf{Q}}\}$ is consistent but ω-inconsistent.

Theorem 37.6. *Let $\mathbf{T}$ be any ω-consistent, axiomatizable theory extending $\mathbf{Q}$. Then $\mathbf{T}$ is not complete.* inc:inp:1in: thm:first-incompleteness

Proof. If $\mathbf{T}$ is ω-consistent, it is consistent, so $\mathbf{T} \nvdash \gamma_{\mathbf{T}}$ by Lemma 37.3. By Lemma 37.5, $\mathbf{T} \nvdash \neg\gamma_{\mathbf{T}}$. This means that $\mathbf{T}$ is incomplete, since it derives neither $\gamma_{\mathbf{T}}$ nor $\neg\gamma_{\mathbf{T}}$. □

---

content/incompleteness/incompleteness-provability/rosser-thm.tex

37.4 Rosser's Theorem

inc:inp:ros: sec

Can we modify Gödel's proof to get a stronger result, replacing "ω-consistent" with simply "consistent"? The answer is "yes," using a trick discovered by Rosser. Rosser's trick is to use a "modified" derivability predicate $\mathsf{RProv}_T(y)$ instead of $\mathsf{Prov}_{\mathbf{T}}(y)$.

inc:inp:ros: thm:rosser

Theorem 37.7. *Let* $\mathbf{T}$ *be any consistent, axiomatizable theory extending* $\mathbf{Q}$*. Then* $\mathbf{T}$ *is not complete.*

Proof. Recall that $\mathsf{Prov}_{\mathbf{T}}(y)$ is defined as $\exists x\, \mathsf{Prf}_{\mathbf{T}}(x, y)$, where $\mathsf{Prf}_{\mathbf{T}}(x, y)$ represents the decidable relation which holds iff x is the Gödel number of a derivation of the sentence with Gödel number y. The relation that holds between x and y if x is the Gödel number of a *refutation* of the sentence with Gödel number y is also decidable. Let $\mathrm{not}(x)$ be the primitive recursive function which does the following: if x is the code of a formula φ, $\mathrm{not}(x)$ is a code of $\neg\varphi$. Then $\mathrm{Ref}_{\mathbf{T}}(x, y)$ holds iff $\mathrm{Prf}_{\mathbf{T}}(x, \mathrm{not}(y))$. Let $\mathsf{Ref}_{\mathbf{T}}(x, y)$ represent it. Then, if $\mathbf{T} \vdash \neg\varphi$ and δ is a corresponding derivation, $\mathbf{Q} \vdash \mathsf{Ref}_{\mathbf{T}}(\ulcorner\delta\urcorner, \ulcorner\varphi\urcorner)$. We define $\mathsf{RProv}_{\mathbf{T}}(y)$ as

$$\exists x\, (\mathsf{Prf}_{\mathbf{T}}(x, y) \wedge \forall z\, (z < x \to \neg\mathsf{Ref}_{\mathbf{T}}(z, y))).$$

Roughly, $\mathsf{RProv}_{\mathbf{T}}(y)$ says "there is a proof of y in $\mathbf{T}$, and there is no shorter refutation of y." Assuming $\mathbf{T}$ is consistent, $\mathsf{RProv}_{\mathbf{T}}(y)$ is true of the same numbers as $\mathsf{Prov}_{\mathbf{T}}(y)$; but from the point of view of *provability* in $\mathbf{T}$ (and we now know that there is a difference between truth and provability!) the two have different properties. If $\mathbf{T}$ is *in*consistent, then the two do *not* hold of the same numbers! ($\mathsf{RProv}_{\mathbf{T}}(y)$ is often read as "y is Rosser provable." Since, as just discussed, Rosser provability is not some special kind of provability—in inconsistent theories, there are sentences that are provable but not Rosser provable—this may be confusing. To avoid the confusion, you could instead read it as "y is shmovable.")

By the fixed-point lemma, there is a formula $\rho_{\mathbf{T}}$ such that

inc:inp:ros: RT

$$\mathbf{Q} \vdash \rho_{\mathbf{T}} \leftrightarrow \neg\mathsf{RProv}_{\mathbf{T}}(\ulcorner\rho_{\mathbf{T}}\urcorner). \tag{37.4}$$

In contrast to the proof of Theorem 37.6, here we claim that if $\mathbf{T}$ is consistent, $\mathbf{T}$ doesn't derive $\rho_{\mathbf{T}}$, and $\mathbf{T}$ also doesn't derive $\neg\rho_{\mathbf{T}}$. (In other words, we don't need the assumption of ω-consistency.)

First, let's show that $\mathbf{T} \nvdash \rho_{\mathbf{T}}$. Suppose it did, so there is a derivation of $\rho_{\mathbf{T}}$ from T; let n be its Gödel number. Then $\mathbf{Q} \vdash \mathsf{Prf}_{\mathbf{T}}(\overline{n}, \ulcorner\rho_{\mathbf{T}}\urcorner)$, since $\mathsf{Prf}_{\mathbf{T}}$ represents $\mathrm{Prf}_{\mathbf{T}}$ in $\mathbf{Q}$. Also, for each $k < n$, k is not the Gödel number of a derivation of $\neg\rho_{\mathbf{T}}$, since $\mathbf{T}$ is consistent. So for each $k < n$, $\mathbf{Q} \vdash \neg\mathsf{Ref}_{\mathbf{T}}(\overline{k}, \ulcorner\rho_{\mathbf{T}}\urcorner)$. By Lemma 35.24, $\mathbf{Q} \vdash \forall z\, (z < \overline{n} \to \neg\mathsf{Ref}_{\mathbf{T}}(z, \ulcorner\rho_{\mathbf{T}}\urcorner))$. Thus,

$$\mathbf{Q} \vdash \exists x\, (\mathsf{Prf}_{\mathbf{T}}(x, \ulcorner\rho_{\mathbf{T}}\urcorner) \wedge \forall z\, (z < x \to \neg\mathsf{Ref}_{\mathbf{T}}(z, \ulcorner\rho_{\mathbf{T}}\urcorner))),$$

but that's just $\mathsf{RProv}_{\mathbf{T}}(\ulcorner \rho_{\mathbf{T}} \urcorner)$. By eq. (37.4), $\mathbf{Q} \vdash \neg\rho_{\mathbf{T}}$. Since $\mathbf{T}$ extends $\mathbf{Q}$, also $\mathbf{T} \vdash \neg\rho_{\mathbf{T}}$. We've assumed that $\mathbf{T} \vdash \rho_{\mathbf{T}}$, so $\mathbf{T}$ would be inconsistent, contrary to the assumption of the theorem.

Now, let's show that $\mathbf{T} \nvdash \neg\rho_{\mathbf{T}}$. Again, suppose it did, and suppose n is the Gödel number of a derivation of $\neg\rho_{\mathbf{T}}$. Then $\mathsf{Ref}_{\mathbf{T}}(n, {}^{\#}\rho_{\mathbf{T}}{}^{\#})$ holds, and since $\mathsf{Ref}_{\mathbf{T}}$ represents $\mathrm{Ref}_{\mathbf{T}}$ in $\mathbf{Q}$, $\mathbf{Q} \vdash \mathsf{Ref}_{\mathbf{T}}(\overline{n}, \ulcorner \rho_{\mathbf{T}} \urcorner)$. We'll again show that $\mathbf{T}$ would then be inconsistent because it would also derive $\rho_{\mathbf{T}}$. Since

$$\mathbf{Q} \vdash \rho_{\mathbf{T}} \leftrightarrow \neg\mathsf{RProv}_{\mathbf{T}}(\ulcorner \rho_{\mathbf{T}} \urcorner),$$

and since $\mathbf{T}$ extends $\mathbf{Q}$, it suffices to show that

$$\mathbf{Q} \vdash \neg\mathsf{RProv}_{\mathbf{T}}(\ulcorner \rho_{\mathbf{T}} \urcorner).$$

The sentence $\neg\mathsf{RProv}_{\mathbf{T}}(\ulcorner \rho_{\mathbf{T}} \urcorner)$, i.e.,

$$\neg\exists x\, (\mathsf{Prf}_{\mathbf{T}}(x, \ulcorner \rho_{\mathbf{T}} \urcorner) \wedge \forall z\, (z < x \to \neg\mathsf{Ref}_{\mathbf{T}}(z, \ulcorner \rho_{\mathbf{T}} \urcorner))),$$

is logically equivalent to

$$\forall x\, (\mathsf{Prf}_{\mathbf{T}}(x, \ulcorner \rho_{\mathbf{T}} \urcorner) \to \exists z\, (z < x \wedge \mathsf{Ref}_{\mathbf{T}}(z, \ulcorner \rho_{\mathbf{T}} \urcorner))).$$

We argue informally using logic, making use of facts about what $\mathbf{Q}$ derives. Suppose x is arbitrary and $\mathsf{Prf}_{\mathbf{T}}(x, \ulcorner \rho_{\mathbf{T}} \urcorner)$. We already know that $\mathbf{T} \nvdash \rho_{\mathbf{T}}$, and so for every k, $\mathbf{Q} \vdash \neg\mathsf{Prf}_{\mathbf{T}}(\overline{k}, \ulcorner \rho_{\mathbf{T}} \urcorner)$. Thus, for every k it follows that $x \neq \overline{k}$. In particular, we have (a) that $x \neq \overline{n}$. We also have $\neg(x = \overline{0} \vee x = \overline{1} \vee \cdots \vee x = \overline{n-1})$ and so by Lemma 35.24, (b) $\neg(x < \overline{n})$. By Lemma 35.25, $\overline{n} < x$. Since $\mathbf{Q} \vdash \mathsf{Ref}_{\mathbf{T}}(\overline{n}, \ulcorner \rho_{\mathbf{T}} \urcorner)$, we have $\overline{n} < x \wedge \mathsf{Ref}_{\mathbf{T}}(\overline{n}, \ulcorner \rho_{\mathbf{T}} \urcorner)$, and from that $\exists z\, (z < x \wedge \mathsf{Ref}_{\mathbf{T}}(z, \ulcorner \rho_{\mathbf{T}} \urcorner))$. Since x was arbitrary we get, as required, that

$$\forall x\, (\mathsf{Prf}_{\mathbf{T}}(x, \ulcorner \rho_{\mathbf{T}} \urcorner) \to \exists z\, (z < x \wedge \mathsf{Ref}_{\mathbf{T}}(z, \ulcorner \rho_{\mathbf{T}} \urcorner))). \qquad \square$$

Problem 37.3. Two sets A and B of natural numbers are said to be *computably inseparable* if there is no decidable set X such that $A \subseteq X$ and $B \subseteq \overline{X}$ ($\overline{X}$ is the complement, $\mathbb{N} \setminus X$, of X). Let $\mathbf{T}$ be a consistent axiomatizable extension of $\mathbf{Q}$. Suppose A is the set of Gödel numbers of sentences provable in $\mathbf{T}$ and B the set of Gödel numbers of sentences refutable in $\mathbf{T}$. Prove that A and B are computably inseparable.

---

content/incompleteness/incompleteness-provability/godels-paper.tex

37.5 Comparison with Gödel's Original Paper

inc:inp:gop:sec

It is worthwhile to spend some time with Gödel's 1931 paper. The introduction sketches the ideas we have just discussed. Even if you just skim through the

paper, it is easy to see what is going on at each stage: first Gödel describes the formal system P (syntax, axioms, proof rules); then he defines the primitive recursive functions and relations; then he shows that xBy is primitive recursive, and argues that the primitive recursive functions and relations are represented in **P**. He then goes on to prove the incompleteness theorem, as above. In Section 3, he shows that one can take the unprovable assertion to be a sentence in the language of arithmetic. This is the origin of the β-lemma, which is what we also used to handle sequences in showing that the recursive functions are representable in **Q**. Gödel doesn't go so far to isolate a minimal set of axioms that suffice, but we now know that **Q** will do the trick. Finally, in Section 4, he sketches a proof of the second incompleteness theorem.

---

content/incompleteness/incompleteness-provability/provability-conditions.tex

37.6 The Derivability Conditions for PA

inc:inp:prc:sec

Peano arithmetic, or **PA**, is the theory extending **Q** with induction axioms for all formulas. In other words, one adds to **Q** axioms of the form

$$(\varphi(0) \wedge \forall x\,(\varphi(x) \to \varphi(x'))) \to \forall x\,\varphi(x)$$

for every formula φ. Notice that this is really a *schema*, which is to say, infinitely many axioms (and it turns out that **PA** is *not* finitely axiomatizable). But since one can effectively determine whether or not a string of symbols is an instance of an induction axiom, the set of axioms for **PA** is computable. **PA** is a much more robust theory than **Q**. For example, one can easily prove that addition and multiplication are commutative, using induction in the usual way. In fact, most finitary number-theoretic and combinatorial arguments can be carried out in **PA**.

Since **PA** is computably axiomatized, the derivability predicate $\mathrm{Prf}_{\mathbf{PA}}(x, y)$ is computable and hence represented in **Q** (and so, in **PA**). As before, we will take $\mathsf{Prf}_{\mathbf{PA}}(x, y)$ to denote the formula representing the relation. Let $\mathsf{Prov}_{\mathbf{PA}}(y)$ be the formula $\exists x\,\mathrm{Prf}_{\mathbf{PA}}(x, y)$, which, intuitively says, "y is derivable from the axioms of **PA**." The reason we need a little bit more than the axioms of **Q** is we need to know that the theory we are using is strong enough to derive a few basic facts about this derivability predicate. In fact, what we need are the following facts:

P1. If $\mathbf{PA} \vdash \varphi$, then $\mathbf{PA} \vdash \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi\urcorner)$.

P2. For all formulas φ and ψ,

$$\mathbf{PA} \vdash \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi \to \psi\urcorner) \to (\mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi\urcorner) \to \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\psi\urcorner)).$$

P3. For every formula φ,

$$\mathbf{PA} \vdash \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi\urcorner) \to \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi\urcorner)\urcorner).$$

The only way to verify that these three properties hold is to describe the formula $\mathsf{Prov}_{\mathbf{PA}}(y)$ carefully and use the axioms of **PA** to describe the relevant formal derivations. Conditions (1) and (2) are easy; it is really condition (3) that requires work. (Think about what kind of work it entails ...) Carrying out the details would be tedious and uninteresting, so here we will ask you to take it on faith that **PA** has the three properties listed above. A reasonable choice of $\mathsf{Prov}_{\mathbf{PA}}(y)$ will also satisfy

P4. If $\mathbf{PA} \vdash \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi\urcorner)$, then $\mathbf{PA} \vdash \varphi$.

But we will not need this fact.

digression

Incidentally, Gödel was lazy in the same way we are being now. At the end of the 1931 paper, he sketches the proof of the second incompleteness theorem, and promises the details in a later paper. He never got around to it; since everyone who understood the argument believed that it could be carried out (he did not need to fill in the details.)

---

content/incompleteness/incompleteness-provability/second-incompleteness-thm.tex

37.7 The Second Incompleteness Theorem

How can we express the assertion that **PA** doesn't prove its own consistency? Saying **PA** is inconsistent amounts to saying that $\mathbf{PA} \vdash 0 = 1$. So we can take the consistency statement $\mathsf{Con}_{\mathbf{PA}}$ to be the sentence $\neg\mathsf{Prov}_{\mathbf{PA}}(\ulcorner 0 = 1\urcorner)$, and then the following theorem does the job:

inc:inp:2in:sec

Theorem 37.8. *Assuming* **PA** *is consistent, then* **PA** *does not derive* $\mathsf{Con}_{\mathbf{PA}}$.

inc:inp:2in:thm:second-incompleteness

It is important to note that the theorem depends on the particular representation of $\mathsf{Con}_{\mathbf{PA}}$ (i.e., the particular representation of $\mathsf{Prov}_{\mathbf{PA}}(y)$). All we will use is that the representation of $\mathsf{Prov}_{\mathbf{PA}}(y)$ satisfies the three derivability conditions, so the theorem generalizes to any theory with a derivability predicate having these properties.

It is informative to read Gödel's sketch of an argument, since the theorem follows like a good punch line. It goes like this. Let $\gamma_{\mathbf{PA}}$ be the Gödel sentence that we constructed in the proof of Theorem 37.6. We have shown "If **PA** is consistent, then **PA** does not derive $\gamma_{\mathbf{PA}}$." If we formalize this *in* **PA**, we have a proof of

$$\mathsf{Con}_{\mathbf{PA}} \to \neg\mathsf{Prov}_{\mathbf{PA}}(\ulcorner\gamma_{\mathbf{PA}}\urcorner).$$

Now suppose **PA** derives $\mathsf{Con}_{\mathbf{PA}}$. Then it derives $\neg\mathsf{Prov}_{\mathbf{PA}}(\ulcorner\gamma_{\mathbf{PA}}\urcorner)$. But since $\gamma_{\mathbf{PA}}$ is a Gödel sentence, this is equivalent to $\gamma_{\mathbf{PA}}$. So **PA** derives $\gamma_{\mathbf{PA}}$.

But: we know that if **PA** is consistent, it doesn't derive $\gamma_{\mathbf{PA}}$! So if **PA** is consistent, it can't derive $\mathsf{Con}_{\mathbf{PA}}$.

To make the argument more precise, we will let $\gamma_{\mathbf{PA}}$ be the Gödel sentence for **PA** and use the derivability conditions (P1)–(P3) to show that **PA** derives

$\mathsf{Con}_{\mathbf{PA}} \to \gamma_{\mathbf{PA}}$. This will show that **PA** doesn't derive $\mathsf{Con}_{\mathbf{PA}}$. Here is a sketch of the proof, in **PA**. (For simplicity, we drop the **PA** subscripts.)

inc:inp:2in:G2-1
$$\gamma \leftrightarrow \neg\mathsf{Prov}(\ulcorner\gamma\urcorner) \tag{37.5}$$
γ is a Gödel sentence

inc:inp:2in:G2-2
$$\gamma \to \neg\mathsf{Prov}(\ulcorner\gamma\urcorner) \tag{37.6}$$
from eq. (37.5)

inc:inp:2in:G2-3
$$\gamma \to (\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \bot) \tag{37.7}$$
from eq. (37.6) by logic

inc:inp:2in:G2-4
$$\mathsf{Prov}(\ulcorner\gamma \to (\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \bot)\urcorner) \tag{37.8}$$
by from eq. (37.7) by condition P1

inc:inp:2in:G2-5
$$\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \mathsf{Prov}(\ulcorner(\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \bot)\urcorner) \tag{37.9}$$
from eq. (37.8) by condition P2

inc:inp:2in:G2-6
$$\mathsf{Prov}(\ulcorner\gamma\urcorner) \to (\mathsf{Prov}(\ulcorner\mathsf{Prov}(\ulcorner\gamma\urcorner)\urcorner) \to \mathsf{Prov}(\ulcorner\bot\urcorner)) \tag{37.10}$$
from eq. (37.9) by condition P2 and logic

inc:inp:2in:G2-7
$$\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \mathsf{Prov}(\ulcorner\mathsf{Prov}(\ulcorner\gamma\urcorner)\urcorner) \tag{37.11}$$
by P3

inc:inp:2in:G2-8
$$\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \mathsf{Prov}(\ulcorner\bot\urcorner) \tag{37.12}$$
from eq. (37.10) and eq. (37.11) by logic

inc:inp:2in:G2-9
$$\mathsf{Con} \to \neg\mathsf{Prov}(\ulcorner\gamma\urcorner) \tag{37.13}$$
contraposition of eq. (37.12) and $\mathsf{Con} \equiv \neg\mathsf{Prov}(\ulcorner\bot\urcorner)$

$\mathsf{Con} \to \gamma$
from eq. (37.5) and eq. (37.13) by logic

The use of logic in the above just elementary facts from propositional logic, e.g., eq. (37.7) uses $\vdash \neg\varphi \leftrightarrow (\varphi \to \bot)$ and eq. (37.12) uses $\varphi \to (\psi \to \chi), \varphi \to \psi \vdash \varphi \to \chi$. The use of condition P2 in eq. (37.9) and eq. (37.10) relies on instances of P2, $\mathsf{Prov}(\ulcorner\varphi \to \psi\urcorner) \to (\mathsf{Prov}(\ulcorner\varphi\urcorner) \to \mathsf{Prov}(\ulcorner\psi\urcorner))$. In the first one, $\varphi \equiv \gamma$ and $\psi \equiv \mathsf{Prov}(\ulcorner\gamma\urcorner) \to \bot$; in the second, $\varphi \equiv \mathsf{Prov}(\ulcorner G\urcorner)$ and $\psi \equiv \bot$.

The more abstract version of the second incompleteness theorem is as follows:

inc:inp:2in:thm:second-incompleteness-gen
Theorem 37.9. *Let* **T** *be any consistent, axiomatized theory extending* **Q** *and let* $\mathsf{Prov}_{\mathbf{T}}(y)$ *be any formula satisfying derivability conditions P1–P3 for* **T**. *Then* **T** *does not derive* Con_T.

Problem 37.4. Show that **PA** derives $\gamma_{\mathbf{PA}} \to \mathsf{Con}_{\mathbf{PA}}$.

digression
The moral of the story is that no "reasonable" consistent theory for mathematics can derive its own consistency statement. Suppose **T** is a theory of mathematics that includes **Q** and Hilbert's "finitary" reasoning (whatever that may be). Then, the whole of **T** cannot derive the consistency statement of **T**,

and so, a fortiori, the finitary fragment can't derive the consistency statement of $\mathbf{T}$ either. In that sense, there cannot be a finitary consistency proof for "all of mathematics."

There is some leeway in interpreting the term "finitary," and Gödel, in the 1931 paper, grants the possibility that something we may consider "finitary" may lie outside the kinds of mathematics Hilbert wanted to formalize. But Gödel was being charitable; today, it is hard to see how we might find something that can reasonably be called finitary but is not formalizable in, say, **ZFC**, Zermelo–Fraenkel set theory with the axiom of choice.

---

content/incompleteness/incompleteness-provability/lob-thm.tex

37.8 Löb's Theorem

inc:inp:lob:sec

The Gödel sentence for a theory $\mathbf{T}$ is a fixed point of $\neg\mathsf{Prov}_{\mathbf{T}}(y)$, i.e., a sentence γ such that

$$\mathbf{T} \vdash \neg\mathsf{Prov}_{\mathbf{T}}(\ulcorner\gamma\urcorner) \leftrightarrow \gamma.$$

It is not derivable, because if $\mathbf{T} \vdash \gamma$, (a) by derivability condition (1), $\mathbf{T} \vdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\gamma\urcorner)$, and (b) $\mathbf{T} \vdash \gamma$ together with $\mathbf{T} \vdash \neg\mathsf{Prov}_{\mathbf{T}}(\ulcorner\gamma\urcorner) \leftrightarrow \gamma$ gives $\mathbf{T} \vdash \neg\mathsf{Prov}_{\mathbf{T}}(\ulcorner\gamma\urcorner)$, and so $\mathbf{T}$ would be inconsistent. Now it is natural to ask about the status of a fixed point of $\mathsf{Prov}_{\mathbf{T}}(y)$, i.e., a sentence δ such that

$$\mathbf{T} \vdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\delta\urcorner) \leftrightarrow \delta.$$

If it were derivable, $\mathbf{T} \vdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\delta\urcorner)$ by condition (1), but the same conclusion follows if we apply modus ponens to the equivalence above. Hence, we don't get that $\mathbf{T}$ is inconsistent, at least not by the same argument as in the case of the Gödel sentence. This of course does not show that $\mathbf{T}$ *does* derive δ.

We can make headway on this question if we generalize it a bit. The left-to-right direction of the fixed point equivalence, $\mathsf{Prov}_{\mathbf{T}}(\ulcorner\delta\urcorner) \to \delta$, is an instance of a general schema called a *reflection principle*: $\mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner) \to \varphi$. It is called that because it expresses, in a sense, that $\mathbf{T}$ can "reflect" about what it can derive; basically it says, "If $\mathbf{T}$ can derive φ, then φ is true," for any φ. This is true for sound theories only, of course, and this suggests that theories will in general not derive every instance of it. So which instances can a theory (strong enough, and satisfying the derivability conditions) derive? Certainly all those where φ itself is derivable. And that's it, as the next result shows.

inc:inp:lob:thm:lob

Theorem 37.10. *Let $\mathbf{T}$ be an axiomatizable theory extending $\mathbf{Q}$, and suppose $\mathsf{Prov}_{\mathbf{T}}(y)$ is a formula satisfying conditions P1–P3 from section 37.7. If $\mathbf{T}$ derives $\mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner) \to \varphi$, then in fact $\mathbf{T}$ derives φ.*

Put differently, if $\mathbf{T} \nvdash \varphi$, then $\mathbf{T} \nvdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner) \to \varphi$. This result is known as Löb's theorem.

explanation

The heuristic for the proof of Löb's theorem is a clever proof that Santa Claus exists. (If you don't like that conclusion, you are free to substitute any other conclusion you would like.) Here it is:

1. Let X be the sentence, "If X is true, then Santa Claus exists."

2. Suppose X is true.

3. Then what it says holds; i.e., we have: if X is true, then Santa Claus exists.

4. Since we are assuming X is true, we can conclude that Santa Claus exists, by modus ponens from (2) and (3).

5. We have succeeded in deriving (4), "Santa Claus exists," from the assumption (2), "X is true." By conditional proof, we have shown: "If X is true, then Santa Claus exists."

6. But this is just the sentence X. So we have shown that X is true.

7. But then, by the argument (2)–(4) above, Santa Claus exists.

A formalization of this idea, replacing "is true" with "is derivable," and "Santa Claus exists" with φ, yields the proof of Löb's theorem. The trick is to apply the fixed-point lemma to the formula $\mathsf{Prov}_{\mathbf{T}}(y) \to \varphi$. The fixed point of that corresponds to the sentence X in the preceding sketch.

Proof of Theorem 37.10. Suppose φ is a sentence such that $\mathbf{T}$ derives $\mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner) \to \varphi$. Let $\psi(y)$ be the formula $\mathsf{Prov}_{\mathbf{T}}(y) \to \varphi$, and use the fixed-point lemma to find a sentence θ such that $\mathbf{T}$ derives $\theta \leftrightarrow \psi(\ulcorner\theta\urcorner)$. Then each of the following

is derivable in $\mathbf{T}$:

$$\theta \leftrightarrow (\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \to \varphi) \tag{37.14}$$ inc:inp:lob:L-1

θ is a fixed point of $\psi(y)$

$$\theta \to (\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \to \varphi) \tag{37.15}$$ inc:inp:lob:L-2

from eq. (37.14)

$$\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta \to (\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \to \varphi)\urcorner) \tag{37.16}$$ inc:inp:lob:L-3

from eq. (37.15) by condition P1

$$\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \to \mathsf{Prov}_{\mathbf{T}}(\ulcorner\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \to \varphi\urcorner) \tag{37.17}$$ inc:inp:lob:L-4

from eq. (37.16) using condition P2

$$\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \to (\mathsf{Prov}_{\mathbf{T}}(\ulcorner\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner)\urcorner) \to \mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner)) \tag{37.18}$$ inc:inp:lob:L-5

from eq. (37.17) using P2 again

$$\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \to \mathsf{Prov}_{\mathbf{T}}(\ulcorner\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner)\urcorner) \tag{37.19}$$ inc:inp:lob:L-6

by derivability condition P3

$$\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \to \mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner) \tag{37.20}$$ inc:inp:lob:L-7

from eq. (37.18) and eq. (37.19)

$$\mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner) \to \varphi \tag{37.21}$$ inc:inp:lob:L-8

by assumption of the theorem

$$\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \to \varphi \tag{37.22}$$ inc:inp:lob:L-9

from eq. (37.20) and eq. (37.21)

$$(\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \to \varphi) \to \theta \tag{37.23}$$ inc:inp:lob:L-10

from eq. (37.14)

$$\theta \tag{37.24}$$ inc:inp:lob:L-11

from eq. (37.22) and eq. (37.23)

$$\mathsf{Prov}_{\mathbf{T}}(\ulcorner\theta\urcorner) \tag{37.25}$$ inc:inp:lob:L-12

from eq. (37.24) by condition P1

φ from eq. (37.21) and eq. (37.25) □

With Löb's theorem in hand, there is a short proof of the second incompleteness theorem (for theories having a derivability predicate satisfying conditions P1–P3): if $\mathbf{T} \vdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\bot\urcorner) \to \bot$, then $\mathbf{T} \vdash \bot$. If $\mathbf{T}$ is consistent, $\mathbf{T} \nvdash \bot$. So, $\mathbf{T} \nvdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\bot\urcorner) \to \bot$, i.e., $\mathbf{T} \nvdash \mathsf{Con}_{\mathbf{T}}$. We can also apply it to show that δ, the fixed point of $\mathsf{Prov}_{\mathbf{T}}(x)$, is derivable. For since

$$\mathbf{T} \vdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\delta\urcorner) \leftrightarrow \delta$$

in particular

$$\mathbf{T} \vdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\delta\urcorner) \to \delta$$

and so by Löb's theorem, $\mathbf{T} \vdash \delta$.

Problem 37.5. Let $\mathbf{T}$ be a computably axiomatized theory, and let $\mathsf{Prov}_{\mathbf{T}}$ be a derivability predicate for $\mathbf{T}$. Consider the following four statements:

1. If $T \vdash \varphi$, then $T \vdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner)$.
2. $T \vdash \varphi \to \mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner)$.
3. If $T \vdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner)$, then $T \vdash \varphi$.
4. $T \vdash \mathsf{Prov}_{\mathbf{T}}(\ulcorner\varphi\urcorner) \to \varphi$

Under what conditions are each of these statements true?

---

content/incompleteness/incompleteness-provability/tarski-thm.tex

37.9 The Undefinability of Truth

inc:inp:tar:sec

The notion of *definability* depends on having a formal semantics for the language of arithmetic. We have described a set of formulas and sentences in the language of arithmetic. The "intended interpretation" is to read such sentences as making assertions about the natural numbers, and such an assertion can be true or false. Let $\mathfrak{N}$ be the structure with domain $\mathbb{N}$ and the standard interpretation for the symbols in the language of arithmetic. Then $\mathfrak{N} \vDash \varphi$ means "φ is true in the standard interpretation."

Definition 37.11. A relation $R(x_1, \ldots, x_k)$ of natural numbers is *definable* in $\mathfrak{N}$ if and only if there is a formula $\varphi(x_1, \ldots, x_k)$ in the language of arithmetic such that for every $n_1, \ldots, n_k$, $R(n_1, \ldots, n_k)$ if and only if $\mathfrak{N} \vDash \varphi(\overline{n}_1, \ldots, \overline{n}_k)$.

Put differently, a relation is definable in $\mathfrak{N}$ if and only if it is representable in the theory $\mathbf{TA}$, where $\mathbf{TA} = \{\varphi : \mathfrak{N} \vDash \varphi\}$ is the set of true sentences of arithmetic. (If this is not immediately clear to you, you should go back and check the definitions and convince yourself that this is the case.)

Lemma 37.12. *Every computable relation is definable in* $\mathfrak{N}$.

Proof. It is easy to check that the formula representing a relation in $\mathbf{Q}$ defines the same relation in $\mathfrak{N}$. □

Now one can ask, is the converse also true? That is, is every relation definable in $\mathfrak{N}$ computable? The answer is no. For example:

Lemma 37.13. *The halting relation is definable in* $\mathfrak{N}$.

Proof. Recall that the Kleene normal form theorem states that every partial computable function f has an index e such that $f(x) = U(\mu s\ T(e, x, s))$ for all $x \in \mathbb{N}$, where U and T are primitive recursive and therefore total. Thus,

$f(x)$ is defined (i.e., the computation halts) iff there is an s such that $T(e, x, s)$ holds.

Now let H be the halting relation, i.e.,

$$H = \{\langle e, x\rangle : \exists s\, T(e, x, s)\}.$$

Let θ_T define T in $\mathfrak{N}$. Then

$$H = \{\langle e, x\rangle : \mathfrak{N} \vDash \exists s\, \theta_T(\overline{e}, \overline{x}, s)\},$$

so $\exists s\, \theta_T(z, x, s)$ defines H in $\mathfrak{N}$. □

Problem 37.6. Show that $Q(n) \Leftrightarrow n \in \{^{\ulcorner}\varphi^{\#} : \mathbf{Q} \vdash \varphi\}$ is definable in arithmetic.

What about **TA** itself? Is it definable in arithmetic? That is: is the set $\{^{\ulcorner}\varphi^{\#} : \mathfrak{N} \vDash \varphi\}$ definable in arithmetic? Tarski's theorem answers this in the negative.

Theorem 37.14. *The set of true sentences of arithmetic is not definable in arithmetic.* *inc:inp:tar: thm:tarski*

Proof. Suppose $\theta(x)$ defined it, i.e., $\mathfrak{N} \vDash \varphi$ iff $\mathfrak{N} \vDash \theta(^{\ulcorner}\varphi^{\urcorner})$. By the fixed-point lemma, there is a formula φ such that $\mathbf{Q} \vdash \varphi \leftrightarrow \neg\theta(^{\ulcorner}\varphi^{\urcorner})$, and hence $\mathfrak{N} \vDash \varphi \leftrightarrow \neg\theta(^{\ulcorner}\varphi^{\urcorner})$. But then $\mathfrak{N} \vDash \varphi$ if and only if $\mathfrak{N} \vDash \neg\theta(^{\ulcorner}\varphi^{\urcorner})$, which contradicts the fact that $\theta(y)$ is supposed to define the set of true statements of arithmetic. □

Tarski applied this analysis to a more general philosophical notion of truth. Given any language L, Tarski argued that an adequate notion of truth for L would have to satisfy, for each sentence X,

> 'X' is true if and only if X.

Tarski's oft-quoted example, for English, is the sentence

> 'Snow is white' is true if and only if snow is white.

However, for any language strong enough to represent the diagonal function, and any linguistic predicate $T(x)$, we can construct a sentence X satisfying "X if and only if not $T(`X')$." Given that we do not want a truth predicate to declare some sentences to be both true and false, Tarski concluded that one cannot specify a truth predicate for all sentences in a language without, somehow, stepping outside the bounds of the language. In other words, a the truth predicate for a language cannot be defined in the language itself.

Part VIII

Second-order Logic

This is the beginnings of a part on second-order logic.

Chapter 38

Syntax and Semantics

Basic syntax and semantics for SOL covered so far. As a chapter it's too short. Substitution for second-order variables has to be covered to be able to talk about derivation systems for SOL, and there's some subtle issues there.

---

content/second-order-logic/syntax-and-semantics/introduction.tex

38.1 Introduction

sol:syn:int:sec

In first-order logic, we combine the non-logical symbols of a given language, i.e., its constant symbols, function symbols, and predicate symbols, with the logical symbols to express things about first-order structures. This is done using the notion of satisfaction, which relates a structure $\mathfrak{M}$, together with a variable assignment s, and a formula φ: $\mathfrak{M}, s \vDash \varphi$ holds iff what φ expresses when its constant symbols, function symbols, and predicate symbols are interpreted as $\mathfrak{M}$ says, and its free variables are interpreted as s says, is true. The interpretation of the identity predicate $=$ is built into the definition of $\mathfrak{M}, s \vDash \varphi$, as is the interpretation of $\forall$ and $\exists$. The former is always interpreted as the identity relation on the domain $|\mathfrak{M}|$ of the structure, and the quantifiers are always interpreted as ranging over the entire domain. But, crucially, quantification

is only allowed over elements of the domain, and so only object variables are allowed to follow a quantifier.

In second-order logic, both the language and the definition of satisfaction are extended to include free and bound function and predicate variables, and quantification over them. These variables are related to function symbols and predicate symbols the same way that object variables are related to constant symbols. They play the same role in the formation of terms and formulas of second-order logic, and quantification over them is handled in a similar way. In the *standard* semantics, the second-order quantifiers range over all possible objects of the right type (n-place functions from $|\mathfrak{M}|$ to $|\mathfrak{M}|$ for function variables, n-place relations for predicate variables). For instance, while $\forall v_0 \, (P^1_0(v_0) \vee \neg P^1_0(v_0))$ is a formula in both first- and second-order logic, in the latter we can also consider $\forall V^1_0 \, \forall v_0 \, (V^1_0(v_0) \vee \neg V^1_0(v_0))$ and $\exists V^1_0 \, \forall v_0 \, (V^1_0(v_0) \vee \neg V^1_0(v_0))$. Since these contain no free variables, they are sentences of second-order logic. Here, V^1_0 is a second-order 1-place predicate variable. The allowable interpretations of V^1_0 are the same that we can assign to a 1-place predicate symbol like P^1_0, i.e., subsets of $|\mathfrak{M}|$. Quantification over them then amounts to saying that $\forall v_0 \, (V^1_0(v_0) \vee \neg V^1_0(v_0))$ holds for all ways of assigning a subset of $|\mathfrak{M}|$ as the value of V^1_0, or for at least one. Since every set either contains or fails to contain a given object, both are true in any structure.

---

content/second-order-logic/syntax-and-semantics/terms-formulas.tex

38.2 Terms and Formulas

sol:syn:frm:sec

Like in first-order logic, expressions of second-order logic are built up from a basic vocabulary containing *variables*, *constant symbols*, *predicate symbols* and sometimes *function symbols*. From them, together with logical connectives, quantifiers, and punctuation symbols such as parentheses and commas, *terms* and *formulas* are formed. The difference is that in addition to variables for objects, second-order logic also contains variables for relations and functions, and allows quantification over them. So the logical symbols of second-order logic are those of first-order logic, plus:

1. A denumerable set of second-order relation variables of every arity n: V^n_0, V^n_1, V^n_2, ...
2. A denumerable set of second-order function variables: u^n_0, u^n_1, u^n_2, ...

Just as we use x, y, z as meta-variables for first-order variables v_i, we'll use X, Y, Z, etc., as metavariables for V^n_i and u, v, etc., as meta-variables for u^n_i.

explanation

The non-logical symbols of a second-order language are specified the same way a first-order language is: by listing its constant symbols, function symbols, and predicate symbols.

In first-order logic, the identity predicate $=$ is usually included. In first-order logic, the non-logical symbols of a language $\mathcal{L}$ are crucial to allow us to express anything interesting. There are of course sentences that use no non-logical symbols, but with only $=$ it is hard to say anything interesting. In second-order logic, since we have an unlimited supply of relation and function variables, we can say anything we can say in a first-order language even without a special supply of non-logical symbols.

Definition 38.1 (Second-order Terms). The set of *second-order terms* of $\mathcal{L}$, $\mathrm{Trm}^2(\mathcal{L})$, is defined by adding to Definition 15.4 the clause

1. If u is an n-place function variable and $t_1, \ldots, t_n$ are terms, then $u(t_1, \ldots, t_n)$ is a term.

So, a second-order term looks just like a first-order term, except that where a first-order term contains a function symbol f_i^n, a second-order term may contain a function variable u_i^n in its place. explanation

Definition 38.2 (Second-order formula). The set of *second-order formulas* $\mathrm{Frm}^2(\mathcal{L})$ of the language $\mathcal{L}$ is defined by adding to Definition 15.5 the clauses

1. If X is an n-place predicate variable and $t_1, \ldots, t_n$ are second-order terms of $\mathcal{L}$, then $X(t_1, \ldots, t_n)$ is an atomic formula.
2. If φ is a formula and u is a function variable, then $\forall u\, \varphi$ is a formula.
3. If φ is a formula and X is a predicate variable, then $\forall X\, \varphi$ is a formula.
4. If φ is a formula and u is a function variable, then $\exists u\, \varphi$ is a formula.
5. If φ is a formula and X is a predicate variable, then $\exists X\, \varphi$ is a formula.

---

content/second-order-logic/syntax-and-semantics/satisfaction.tex

38.3 Satisfaction

sol:syn:sat:sec

To define the satisfaction relation $\mathfrak{M}, s \vDash \varphi$ for second-order formulas, we have to extend the definitions to cover second-order variables. The notion of a structure is the same for second-order logic as it is for first-order logic. There is only a difference for variable assignments s: these now must not just provide values for the first-order variables, but also for the second-order variables. explanation

Definition 38.3 (Variable Assignment). A *variable assignment* s for a structure $\mathfrak{M}$ is a function which maps each

1. object variable v_i to an element of $|\mathfrak{M}|$, i.e., $s(v_i) \in |\mathfrak{M}|$

2. n-place relation variable V_i^n to an n-place relation on $|\mathfrak{M}|$, i.e., $s(V_i^n) \subseteq |\mathfrak{M}|^n$;

3. n-place function variable u_i^n to an n-place function from $|\mathfrak{M}|$ to $|\mathfrak{M}|$, i.e., $s(u_i^n)\colon |\mathfrak{M}|^n \to |\mathfrak{M}|$;

explanation

A structure assigns a value to each constant symbol and function symbol, and a second-order variable assignment assigns objects and functions to each object and function variable. Together, they let us assign a value to every term.

Definition 38.4 (Value of a Term). If t is a term of the language $\mathcal{L}$, $\mathfrak{M}$ is a structure for $\mathcal{L}$, and s is a variable assignment for $\mathfrak{M}$, the *value* $\mathrm{Val}_s^{\mathfrak{M}}(t)$ is defined as for first-order terms, plus the following clause:

$t \equiv u(t_1, \ldots, t_n)$:

$$\mathrm{Val}_s^{\mathfrak{M}}(t) = s(u)(\mathrm{Val}_s^{\mathfrak{M}}(t_1), \ldots, \mathrm{Val}_s^{\mathfrak{M}}(t_n)).$$

Definition 38.5 (x-Variant). If s is a variable assignment for a structure $\mathfrak{M}$, then any variable assignment s' for $\mathfrak{M}$ which differs from s at most in what it assigns to x is called an *x-variant* of s. If s' is an x-variant of s we write $s' \sim_x s$. (Similarly for second-order variables X or u.)

Definition 38.6. If s is a variable assignment for a structure $\mathfrak{M}$ and $m \in |\mathfrak{M}|$, then the assignment $s[m/x]$ is the variable assignment defined by

$$s[m/y] = \begin{cases} m & \text{if } y \equiv x \\ s(y) & \text{otherwise,} \end{cases}$$

If X is an n-place relation variable and $M \subseteq |\mathfrak{M}|^n$, then $s[M/X]$ is the variable assignment defined by

$$s[M/y] = \begin{cases} M & \text{if } y \equiv X \\ s(y) & \text{otherwise.} \end{cases}$$

If u is an n-place function variable and $f\colon |\mathfrak{M}|^n \to |\mathfrak{M}|$, then $s[f/u]$ is the variable assignment defined by

$$s[f/y] = \begin{cases} f & \text{if } y \equiv u \\ s(y) & \text{otherwise.} \end{cases}$$

In each case, y may be any first- or second-order variable.

Definition 38.7 (Satisfaction). For second-order formulas φ, the definition of satisfaction is like Definition 16.11 with the addition of:

1. $\varphi \equiv X^n(t_1, \ldots, t_n)$: $\mathfrak{M}, s \vDash \varphi$ iff $\langle \mathrm{Val}_s^{\mathfrak{M}}(t_1), \ldots, \mathrm{Val}_s^{\mathfrak{M}}(t_n)\rangle \in s(X^n)$.

2. $\varphi \equiv \forall X\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff for every $M \subseteq |\mathfrak{M}|^n$, $\mathfrak{M}, s[M/X] \vDash \psi$.

3. $\varphi \equiv \exists X\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff for at least one $M \subseteq |\mathfrak{M}|^n$ so that $\mathfrak{M}, s[M/X] \vDash \psi$.

4. $\varphi \equiv \forall u\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff for every $f\colon |\mathfrak{M}|^n \to |\mathfrak{M}|$, $\mathfrak{M}, s[f/u] \vDash \psi$.

5. $\varphi \equiv \exists u\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff for at least one $f\colon |\mathfrak{M}|^n \to |\mathfrak{M}|$ so that $\mathfrak{M}, s[f/u] \vDash \psi$.

Example 38.8. Consider the formula $\forall z\, (X(z) \leftrightarrow \neg Y(z))$. It contains no second-order quantifiers, but does contain the second-order variables X and Y (here understood to be one-place). The corresponding first-order sentence $\forall z\, (P(z) \leftrightarrow \neg R(z))$ says that whatever falls under the interpretation of P does not fall under the interpretation of R and vice versa. In a structure, the interpretation of a predicate symbol P is given by the interpretation $P^{\mathfrak{M}}$. But for second-order variables like X and Y, the interpretation is provided, not by the structure itself, but by a variable assignment. Since the second-order formula is not a sentence (it includes free variables X and Y), it is only satisfied relative to a structure $\mathfrak{M}$ together with a variable assignment s.

$\mathfrak{M}, s \vDash \forall z\, (X(z) \leftrightarrow \neg Y(z))$ whenever the elements of $s(X)$ are not elements of $s(Y)$, and vice versa, i.e., iff $s(Y) = |\mathfrak{M}| \setminus s(X)$. For instance, take $|\mathfrak{M}| = \{1, 2, 3\}$. Since no predicate symbols, function symbols, or constant symbols are involved, the domain of $\mathfrak{M}$ is all that is relevant. Now for $s_1(X) = \{1, 2\}$ and $s_1(Y) = \{3\}$, we have $\mathfrak{M}, s_1 \vDash \forall z\, (X(z) \leftrightarrow \neg Y(z))$.

By contrast, if we have $s_2(X) = \{1, 2\}$ and $s_2(Y) = \{2, 3\}$, $\mathfrak{M}, s_2 \nvDash \forall z\, (X(z) \leftrightarrow \neg Y(z))$. That's because $\mathfrak{M}, s_2[2/z] \vDash X(z)$ (since $2 \in s_2[2/z](X)$) but $\mathfrak{M}, s_2[2/z] \nvDash \neg Y(z)$ (since also $2 \in s_2[2/z](Y)$).

Example 38.9. $\mathfrak{M}, s \vDash \exists Y\, (\exists y\, Y(y) \wedge \forall z\, (X(z) \leftrightarrow \neg Y(z)))$ if there is an $N \subseteq |\mathfrak{M}|$ such that $\mathfrak{M}, s[N/Y] \vDash (\exists y\, Y(y) \wedge \forall z\, (X(z) \leftrightarrow \neg Y(z)))$. And that is the case for any $N \neq \emptyset$ (so that $\mathfrak{M}, s[N/Y] \vDash \exists y\, Y(y)$) and, as in the previous example, $N = |\mathfrak{M}| \setminus s(X)$. In other words, $\mathfrak{M}, s \vDash \exists Y\, (\exists y\, Y(y) \wedge \forall z\, (X(z) \leftrightarrow \neg Y(z)))$ iff $|\mathfrak{M}| \setminus s(X)$ is non-empty, i.e., $s(X) \neq |\mathfrak{M}|$. So, the formula is satisfied, e.g., if $|\mathfrak{M}| = \{1, 2, 3\}$ and $s(X) = \{1, 2\}$, but not if $s(X) = \{1, 2, 3\} = |\mathfrak{M}|$.

Since the formula is not satisfied whenever $s(X) = |\mathfrak{M}|$, the sentence

$$\forall X\, \exists Y\, (\exists y\, Y(y) \wedge \forall z\, (X(z) \leftrightarrow \neg Y(z)))$$

is never satisfied: For any structure $\mathfrak{M}$, the assignment $s(X) = |\mathfrak{M}|$ will make the sentence false. On the other hand, the sentence

$$\exists X\, \exists Y\, (\exists y\, Y(y) \wedge \forall z\, (X(z) \leftrightarrow \neg Y(z)))$$

is satisfied relative to any assignment s, since we can always find $M \subseteq |\mathfrak{M}|$ but $M \neq |\mathfrak{M}|$ (e.g., $M = \emptyset$).

Example 38.10. The second-order sentence $\forall X \forall y\, X(y)$ says that every 1-place relation, i.e., every property, holds of every object. That is clearly never true, since in every $\mathfrak{M}$, for a variable assignment s with $s(X) = \emptyset$, and $s(y) = a \in |\mathfrak{M}|$ we have $\mathfrak{M}, s \nvDash X(y)$. This means that $\varphi \to \forall X \forall y\, X(y)$ is equivalent in second-order logic to $\neg\varphi$, that is: $\mathfrak{M} \vDash \varphi \to \forall X \forall y\, X(y)$ iff $\mathfrak{M} \vDash \neg\varphi$. In other words, in second-order logic we can define $\neg$ using $\forall$ and $\to$.

Problem 38.1. Show that in second-order logic $\forall$ and $\to$ can define the other connectives:

1. Prove that in second-order logic $\varphi \wedge \psi$ is equivalent to $\forall X\, (\varphi \to (\psi \to \forall x\, X(x)) \to \forall x\, X(x))$.

2. Find a second-order formula using only $\forall$ and $\to$ equivalent to $\varphi \vee \psi$.

---

content/second-order-logic/syntax-and-semantics/semantic-notions.tex

38.4 Semantic Notions

explanation

sol:syn:sem:sec

The central logical notions of *validity*, *entailment*, and *satisfiability* are defined the same way for second-order logic as they are for first-order logic, except that the underlying satisfaction relation is now that for second-order formulas. A second-order sentence, of course, is a formula in which all variables, including predicate and function variables, are bound.

Definition 38.11 (Validity). A sentence φ is *valid*, $\vDash \varphi$, iff $\mathfrak{M} \vDash \varphi$ for every structure $\mathfrak{M}$.

Definition 38.12 (Entailment). A set of sentences Γ *entails* a sentence φ, $\Gamma \vDash \varphi$, iff for every structure $\mathfrak{M}$ with $\mathfrak{M} \vDash \Gamma$, $\mathfrak{M} \vDash \varphi$.

Definition 38.13 (Satisfiability). A set of sentences Γ is *satisfiable* if $\mathfrak{M} \vDash \Gamma$ for some structure $\mathfrak{M}$. If Γ is not satisfiable it is called *unsatisfiable*.

---

content/second-order-logic/syntax-and-semantics/expressive-power.tex

38.5 Expressive Power

explanation

sol:syn:exp:sec

Quantification over second-order variables is responsible for an immense increase in the expressive power of the language over that of first-order logic. Second-order existential quantification lets us say that functions or relations with certain properties exist. In first-order logic, the only way to do that is to specify a non-logical symbol (i.e., a function symbol or predicate symbol) for this purpose. Second-order universal quantification lets us say that all subsets of, relations on, or functions from the domain to the domain have a property.

In first-order logic, we can only say that the subsets, relations, or functions assigned to one of the non-logical symbols of the language have a property. And when we say that subsets, relations, functions exist that have a property, or that all of them have it, we can use second-order quantification in specifying this property as well. This lets us define relations not definable in first-order logic, and express properties of the domain not expressible in first-order logic.

Definition 38.14. If $\mathfrak{M}$ is a structure for a language $\mathcal{L}$, a relation $R \subseteq |\mathfrak{M}|^2$ is *definable* in $\mathcal{L}$ if there is some formula $\varphi_R(x, y)$ with only the variables x and y free, such that $R(a, b)$ holds (i.e., $\langle a, b\rangle \in R$) iff $\mathfrak{M}, s \vDash \varphi_R(x, y)$ for $s(x) = a$ and $s(y) = b$.

Example 38.15. In first-order logic we can define the identity relation $\mathrm{Id}_{|\mathfrak{M}|}$ (i.e., $\{\langle a, a\rangle : a \in |\mathfrak{M}|\}$) by the formula $x = y$. In second-order logic, we can define this relation *without* $=$. For if a and b are the same element of $|\mathfrak{M}|$, then they are elements of the same subsets of $|\mathfrak{M}|$ (since sets are determined by their elements). Conversely, if a and b are different, then they are not elements of the same subsets: e.g., $a \in \{a\}$ but $b \notin \{a\}$ if $a \neq b$. So "being elements of the same subsets of $|\mathfrak{M}|$" is a relation that holds of a and b iff $a = b$. It is a relation that can be expressed in second-order logic, since we can quantify over all subsets of $|\mathfrak{M}|$. Hence, the following formula defines $\mathrm{Id}_{|\mathfrak{M}|}$:

$$\forall X\, (X(x) \leftrightarrow X(y))$$

Problem 38.2. Show that $\forall X\, (X(x) \rightarrow X(y))$ (note: $\rightarrow$ not $\leftrightarrow$!) defines $\mathrm{Id}_{|\mathfrak{M}|}$.

Example 38.16. If R is a two-place predicate symbol, $R^{\mathfrak{M}}$ is a two-place relation on $|\mathfrak{M}|$. Perhaps somewhat confusingly, we'll use R as the predicate symbol for R and for the relation $R^{\mathfrak{M}}$ itself. The *transitive closure* R^* of R is the relation that holds between a and b iff for some $c_1, \ldots, c_k$, $R(a, c_1)$, $R(c_1, c_2)$, $\ldots$, $R(c_k, b)$ holds. This includes the case if $k = 0$, i.e., if $R(a, b)$ holds, so does $R^*(a, b)$. This means that $R \subseteq R^*$. In fact, R^* is the smallest relation that includes R and that is transitive. We can say in second-order logic that X is a transitive relation that includes R:

$$\psi_R(X) \equiv \forall x\, \forall y\, (R(x, y) \rightarrow X(x, y)) \wedge \forall x\, \forall y\, \forall z\, ((X(x, y) \wedge X(y, z)) \rightarrow X(x, z)).$$

The first conjunct says that $R \subseteq X$ and the second that X is transitive.

To say that X is the smallest such relation is to say that it is itself included in every relation that includes R and is transitive. So we can define the transitive closure of R by the formula

$$R^*(X) \equiv \psi_R(X) \wedge \forall Y\, (\psi_R(Y) \rightarrow \forall x\, \forall y\, (X(x, y) \rightarrow Y(x, y))).$$

We have $\mathfrak{M}, s \vDash R^*(X)$ iff $s(X) = R^*$. The transitive closure of R cannot be expressed in first-order logic.

38.6 Describing Infinite and Enumerable Domains

sol:syn:siz:sec

A set M is (Dedekind) infinite iff there is an injective function $f\colon M \to M$ which is not surjective, i.e., with $\operatorname{ran}(f) \neq M$. In first-order logic, we can consider a one-place function symbol f and say that the function $f^{\mathfrak{M}}$ assigned to it in a structure $\mathfrak{M}$ is injective and $\operatorname{ran}(f) \neq |\mathfrak{M}|$:

$$\forall x\, \forall y\, (f(x) = f(y) \to x = y) \wedge \exists y\, \forall x\, y \neq f(x).$$

If $\mathfrak{M}$ satisfies this sentence, $f^{\mathfrak{M}} : |\mathfrak{M}| \to |\mathfrak{M}|$ is injective, and so $|\mathfrak{M}|$ must be infinite. If $|\mathfrak{M}|$ is infinite, and hence such a function exists, we can let $f^{\mathfrak{M}}$ be that function and $\mathfrak{M}$ will satisfy the sentence. However, this requires that our language contains the non-logical symbol f which we use for this purpose. In second-order logic, we can simply say that such a function *exists*. This no-longer requires f, and we obtain the sentence in pure second-order logic

$$\mathrm{Inf} \equiv \exists u\, (\forall x\, \forall y\, (u(x) = u(y) \to x = y) \wedge \exists y\, \forall x\, y \neq u(x)).$$

$\mathfrak{M} \vDash \mathrm{Inf}$ iff $|\mathfrak{M}|$ is infinite. We can then define $\mathrm{Fin} \equiv \neg\mathrm{Inf}$; $\mathfrak{M} \vDash \mathrm{Fin}$ iff $|\mathfrak{M}|$ is finite. No single sentence of pure first-order logic can express that the domain is infinite although an infinite set of them can. There is no set of sentences of pure first-order logic that is satisfied in a structure iff its domain is finite.

Proposition 38.17. $\mathfrak{M} \vDash \mathrm{Inf}$ *iff* $|\mathfrak{M}|$ *is infinite.*

Proof. $\mathfrak{M} \vDash \mathrm{Inf}$ iff $\mathfrak{M}, s \vDash \forall x\, \forall y\, (u(x) = u(y) \to x = y) \wedge \exists y\, \forall x\, y \neq u(x)$ for some s. If it does, $s(u)$ is an injective function, and some $y \in |\mathfrak{M}|$ is not in the range of $s(u)$. Conversely, if there is an injective $f\colon |\mathfrak{M}| \to |\mathfrak{M}|$ with $\operatorname{ran}(f) \neq |\mathfrak{M}|$, then $s(u) = f$ is such a variable assignment. □

A set M is enumerable if there is an enumeration

$$m_0, m_1, m_2, \ldots$$

of its elements (without repetitions but possibly finite). Such an enumeration exists iff there is an element $z \in M$ and a function $f\colon M \to M$ such that z, $f(z)$, $f(f(z))$, ..., are all the elements of M. For if the enumeration exists, $z = m_0$ and $f(m_k) = m_{k+1}$ (or $f(m_k) = m_k$ if m_k is the last element of the enumeration) are the requisite element and function. On the other hand, if such a z and f exist, then z, $f(z)$, $f(f(z))$, ..., is an enumeration of M, and M is enumerable. We can express the existence of z and f in second-order logic to produce a sentence true in a structure iff the structure is enumerable:

$$\mathrm{Count} \equiv \exists z\, \exists u\, \forall X\, ((X(z) \wedge \forall x\, (X(x) \to X(u(x)))) \to \forall x\, X(x))$$

Proposition 38.18. $\mathfrak{M} \vDash \mathrm{Count}$ *iff* $|\mathfrak{M}|$ *is enumerable.*

Proof. Suppose $|\mathfrak{M}|$ is enumerable, and let m_0, m_1, ..., be an enumeration. By removing repetitions we can guarantee that no m_k appears twice. Define $f(m_k) = m_{k+1}$ and let $s(z) = m_0$ and $s(u) = f$. We show that

$$\mathfrak{M}, s \vDash \forall X\,((X(z) \wedge \forall x\,(X(x) \to X(u(x)))) \to \forall x\,X(x))$$

Suppose $M \subseteq |\mathfrak{M}|$ is arbitrary. Suppose further that $\mathfrak{M}, s[M/X] \vDash (X(z) \wedge \forall x\,(X(x) \to X(u(x))))$. Then $s[M/X](z) \in M$ and whenever $x \in M$, also $(s[M/X](u))(x) \in M$. In other words, since $s[M/X] \sim_X s$, $m_0 \in M$ and if $x \in M$ then $f(x) \in M$, so $m_0 \in M$, $m_1 = f(m_0) \in M$, $m_2 = f(f(m_0)) \in M$, etc. Thus, $M = |\mathfrak{M}|$, and so $\mathfrak{M}, s[M/X] \vDash \forall x\,X(x)$. Since $M \subseteq |\mathfrak{M}|$ was arbitrary, we are done: $\mathfrak{M} \vDash \text{Count}$.

Now assume that $\mathfrak{M} \vDash \text{Count}$, i.e.,

$$\mathfrak{M}, s \vDash \forall X\,((X(z) \wedge \forall x\,(X(x) \to X(u(x)))) \to \forall x\,X(x))$$

for some s. Let $m = s(z)$ and $f = s(u)$ and consider $M = \{m, f(m), f(f(m)), \dots\}$. M so defined is clearly enumerable. Then

$$\mathfrak{M}, s[M/X] \vDash (X(z) \wedge \forall x\,(X(x) \to X(u(x)))) \to \forall x\,X(x)$$

by assumption. Also, $\mathfrak{M}, s[M/X] \vDash X(z)$ since $M \ni m = s[M/X](z)$, and also $\mathfrak{M}, s[M/X] \vDash \forall x\,(X(x) \to X(u(x)))$ since whenever $x \in M$ also $f(x) \in M$. So, since both antecedent and conditional are satisfied, the consequent must also be: $\mathfrak{M}, s[M/X] \vDash \forall x\,X(x)$. But that means that $M = |\mathfrak{M}|$, and so $|\mathfrak{M}|$ is enumerable since M is, by definition. □

Problem 38.3. The sentence $\text{Inf} \wedge \text{Count}$ is true in all and only denumerable domains. Adjust the definition of Count so that it becomes a different sentence that directly expresses that the domain is denumerable, and prove that it does.

Chapter 39

Metatheory of Second-order Logic

39.1 Introduction

sol:met:int:sec

First-order logic has a number of nice properties. We know it is not decidable, but at least it is axiomatizable. That is, there are proof systems for first-order logic which are sound and complete, i.e., they give rise to a derivability relation $\vdash$ with the property that for any set of sentences Γ and sentence φ, $\Gamma \vDash \varphi$ iff $\Gamma \vdash \varphi$. This means in particular that the validities of first-order logic are computably enumerable. There is a computable function $f\colon \mathbb{N} \to \mathrm{Sent}(\mathcal{L})$ such that the values of f are all and only the valid sentences of $\mathcal{L}$. This is so because derivations can be enumerated, and those that derive a single sentence are then mapped to that sentence. Second-order logic is more expressive than first-order logic, and so it is in general more complicated to capture its validities. In fact, we'll show that second-order logic is not only undecidable, but its validities are not even computably enumerable. This means there can be no sound and complete proof system for second-order logic (although sound, but incomplete proof systems are available and in fact are important objects of research).

First-order logic also has two more properties: it is compact (if every finite subset of a set Γ of sentences is satisfiable, Γ itself is satisfiable) and the Löwenheim–Skolem Theorem holds for it (if Γ has an infinite model it has a denumerable model). Both of these results fail for second-order logic. Again, the reason is that second-order logic can express facts about the size of domains that first-order logic cannot.

---

content/second-order-logic/metatheory/second-order-arithmetic.tex

39.2 Second-order Arithmetic

sol:met:spa:sec

Recall that the theory **PA** of Peano arithmetic includes the eight axioms of **Q**,

$$\begin{aligned}
&\forall x\, x' \neq \mathsf{0}\\
&\forall x \forall y\, (x' = y' \to x = y)\\
&\forall x\, (x = \mathsf{0} \vee \exists y\, x = y')\\
&\forall x\, (x + \mathsf{0}) = x\\
&\forall x \forall y\, (x + y') = (x + y)'\\
&\forall x\, (x \times \mathsf{0}) = \mathsf{0}\\
&\forall x \forall y\, (x \times y') = ((x \times y) + x)\\
&\forall x \forall y\, (x < y \leftrightarrow \exists z\, (z' + x) = y)
\end{aligned}$$

plus all sentences of the form

$$(\varphi(\mathsf{0}) \wedge \forall x\, (\varphi(x) \to \varphi(x'))) \to \forall x\, \varphi(x).$$

The latter is a "schema," i.e., a pattern that generates infinitely many sentences of the language of arithmetic, one for each formula $\varphi(x)$. We call this

schema the (first-order) *axiom schema of induction*. In *second-order* Peano arithmetic $\mathbf{PA^2}$, induction can be stated as a single sentence. $\mathbf{PA^2}$ consists of the first eight axioms above plus the (second-order) *induction axiom*:

$$\forall X\,((X(\mathsf{o}) \wedge \forall x\,(X(x) \to X(x'))) \to \forall x\,X(x)).$$

It says that if a subset X of the domain contains $\mathsf{o}^{\mathfrak{M}}$ and with any $x \in |\mathfrak{M}|$ also contains $\prime^{\mathfrak{M}}(x)$ (i.e., it is "closed under successor") it contains everything in the domain (i.e., $X = |\mathfrak{M}|$).

The induction axiom guarantees that any structure satisfying it contains only those elements of $|\mathfrak{M}|$ the axioms require to be there, i.e., the values of $\overline{n}$ for $n \in \mathbb{N}$. A model of $\mathbf{PA^2}$ contains no non-standard numbers.

sol:met:spa: thm:sol-pa-standard

Theorem 39.1. *If* $\mathfrak{M} \vDash \mathbf{PA^2}$ *then* $|\mathfrak{M}| = \{\mathrm{Val}^{\mathfrak{M}}(\overline{n}) : n \in \mathbb{N}\}$.

Proof. Let $N = \{\mathrm{Val}^{\mathfrak{M}}(\overline{n}) : n \in \mathbb{N}\}$, and suppose $\mathfrak{M} \vDash \mathbf{PA^2}$. Of course, for any $n \in \mathbb{N}$, $\mathrm{Val}^{\mathfrak{M}}(\overline{n}) \in |\mathfrak{M}|$, so $N \subseteq |\mathfrak{M}|$.

Now for inclusion in the other direction. Consider a variable assignment s with $s(X) = N$. By assumption,

$$\begin{gathered}\mathfrak{M} \vDash \forall X\,((X(\mathsf{o}) \wedge \forall x\,(X(x) \to X(x'))) \to \forall x\,X(x)), \text{ thus} \\ \mathfrak{M}, s \vDash (X(\mathsf{o}) \wedge \forall x\,(X(x) \to X(x'))) \to \forall x\,X(x).\end{gathered}$$

Consider the antecedent of this conditional. $\mathrm{Val}^{\mathfrak{M}}(\mathsf{o}) \in N$, and so $\mathfrak{M}, s \vDash X(\mathsf{o})$. The second conjunct, $\forall x\,(X(x) \to X(x'))$ is also satisfied. For suppose $x \in N$. By definition of N, $x = \mathrm{Val}^{\mathfrak{M}}(\overline{n})$ for some n. That gives $\prime^{\mathfrak{M}}(x) = \mathrm{Val}^{\mathfrak{M}}(\overline{n+1}) \in N$. So, $\prime^{\mathfrak{M}}(x) \in N$.

We have that $\mathfrak{M}, s \vDash X(\mathsf{o}) \wedge \forall x\,(X(x) \to X(x'))$. Consequently, $\mathfrak{M}, s \vDash \forall x\,X(x)$. But that means that for every $x \in |\mathfrak{M}|$ we have $x \in s(X) = N$. So, $|\mathfrak{M}| \subseteq N$. □

sol:met:spa: cor:sol-pa-categorical

Corollary 39.2. *Any two models of* $\mathbf{PA^2}$ *are isomorphic.*

Proof. By Theorem 39.1, the domain of any model of $\mathbf{PA^2}$ is exhausted by $\mathrm{Val}^{\mathfrak{M}}(\overline{n})$. Any such model is also a model of $\mathbf{Q}$. By Proposition 26.3, any such model is standard, i.e., isomorphic to $\mathfrak{N}$. □

Above we defined $\mathbf{PA^2}$ as the theory that contains the first eight arithmetical axioms plus the second-order induction axiom. In fact, thanks to the expressive power of second-order logic, only the *first two* of the arithmetical axioms plus induction are needed for second-order Peano arithmetic.

sol:met:spa: prop:sol-pa-definable

Proposition 39.3. *Let* $\mathbf{PA^{2\dagger}}$ *be the second-order theory containing the first two arithmetical axioms (the successor axioms) and the second-order induction axiom. Then* $\leq$, $+$, *and* $\times$ *are definable in* $\mathbf{PA^{2\dagger}}$.

Proof. To show that $\leq$ is definable, we have to find a formula $\varphi_{\leq}(x, y)$ such that $\mathfrak{N} \vDash \varphi_{\leq}(\overline{n}, \overline{m})$ iff $n \leq m$. Consider the formula

$$\psi(x, Y) \equiv Y(x) \wedge \forall y\,(Y(y) \to Y(y'))$$

Clearly, $\psi(\overline{n}, Y)$ is satisfied by a set $Y \subseteq \mathbb{N}$ iff $\{m : n \leq m\} \subseteq Y$, so we can take $\varphi_{\leq}(x, y) \equiv \forall Y\,(\psi(x, Y) \to Y(y))$.

To see that addition is definable observe that $k+l = m$ iff there is a function u such that $u(0) = k$, $u(n') = u(n)'$ for all n, and $m = u(l)$. We can use this equivalence to define addition in $\mathbf{PA}^{2\dagger}$ by the following formula:

$$\varphi_{+}(x, y, z) \equiv \exists u\,(u(\mathsf{o}) = x \wedge \forall w\, u(x') = u(x)' \wedge u(y) = z)$$

It should be clear that $\mathfrak{N} \vDash \varphi_{+}(\overline{k}, \overline{l}, \overline{m})$ iff $k + l = m$. □

Problem 39.1. Complete the proof of Proposition 39.3.

---

content/second-order-logic/metatheory/undecidability-and-axiomatizability.tex

39.3 Second-order Logic is not Axiomatizable

sol:met:nax:sec
sol:met:nax:thm:sol-undecidable

Theorem 39.4. *Second-order logic is undecidable.*

Proof. A first-order sentence is valid in first-order logic iff it is valid in second-order logic, and first-order logic is undecidable. □

sol:met:nax:cor:sol-not-axiomatizable

Theorem 39.5. *There is no sound and complete derivation system for second-order logic.*

Proof. Let φ be a sentence in the language of arithmetic. $\mathfrak{N} \vDash \varphi$ iff $\mathbf{PA}^{\mathbf{2}} \vDash \varphi$. Let P be the conjunction of the nine axioms of $\mathbf{PA}^{\mathbf{2}}$. $\mathbf{PA}^{\mathbf{2}} \vDash \varphi$ iff $\vDash P \to \varphi$, i.e., $\mathfrak{M} \vDash P \to \varphi$. Now consider the sentence $\forall z\, \forall u\, \forall u'\, \forall u''\, \forall L\,(P' \to \varphi')$ resulting by replacing o by z, $'$ by the one-place function variable u, $+$ and $\times$ by the two-place function-variables u' and u'', respectively, and $<$ by the two-place relation variable L and universally quantifying. It is a valid sentence of pure second-order logic iff the original sentence was valid iff $\mathbf{PA}^{\mathbf{2}} \vDash \varphi$ iff $\mathfrak{N} \vDash \varphi$. Thus if there were a sound and complete proof system for second-order logic, we could use it to define a computable enumeration $f\colon \mathbb{N} \to \mathrm{Sent}(\mathcal{L}_A)$ of the sentences true in $\mathfrak{N}$. This function would be representable in $\mathbf{Q}$ by some first-order formula $\psi_f(x, y)$. Then the formula $\exists x\, \psi_f(x, y)$ would define the set of true first-order sentences of $\mathfrak{N}$, contradicting Tarski's Theorem. □

---

content/second-order-logic/metatheory/compactness.tex

39.4 Second-order Logic is not Compact

sol:met:com:sec

Call a set of sentences Γ *finitely satisfiable* if every one of its finite subsets is satisfiable. First-order logic has the property that if a set of sentences Γ is finitely satisfiable, it is satisfiable. This property is called *compactness*. It has an equivalent version involving entailment: if $\Gamma \models \varphi$, then already $\Gamma_0 \models \varphi$ for some finite subset $\Gamma_0 \subseteq \Gamma$. In this version it is an immediate corollary of the completeness theorem: for if $\Gamma \models \varphi$, by completeness $\Gamma \vdash \varphi$. But a derivation can only make use of finitely many sentences of Γ.

explanation

Compactness is not true for second-order logic. There are sets of second-order sentences that are finitely satisfiable but not satisfiable, and that entail some φ without a finite subset entailing φ.

sol:met:com:thm:sol-undecidable

Theorem 39.6. *Second-order logic is not compact.*

Proof. Recall that

$$\mathrm{Inf} \equiv \exists u\,(\forall x\,\forall y\,(u(x) = u(y) \to x = y) \wedge \exists y\,\forall x\,y \neq u(x))$$

is satisfied in a structure iff its domain is infinite. Let $\varphi^{\geq n}$ be a sentence that asserts that the domain has at least n elements, e.g.,

$$\varphi^{\geq n} \equiv \exists x_1 \ldots \exists x_n\,(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge \cdots \wedge x_{n-1} \neq x_n).$$

Consider the set of sentences

$$\Gamma = \{\neg\mathrm{Inf}, \varphi^{\geq 1}, \varphi^{\geq 2}, \varphi^{\geq 3}, \ldots\}.$$

It is finitely satisfiable, since for any finite subset $\Gamma_0 \subseteq \Gamma$ there is some k so that $\varphi^{\geq k} \in \Gamma$ but no $\varphi^{\geq n} \in \Gamma$ for $n > k$. If $|\mathfrak{M}|$ has k elements, $\mathfrak{M} \models \Gamma_0$. But, Γ is not satisfiable: if $\mathfrak{M} \models \neg\mathrm{Inf}$, $|\mathfrak{M}|$ must be finite, say, of size k. Then $\mathfrak{M} \not\models \varphi^{\geq k+1}$. □

Problem 39.2. Give an example of a set Γ and a sentence φ so that $\Gamma \models \varphi$ but for every finite subset $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \not\models \varphi$.

---

content/second-order-logic/metatheory/loewenheim-skolem.tex

39.5 The Löwenheim–Skolem Theorem Fails for Second-order Logic

sol:met:lst:sec

The (Downward) Löwenheim–Skolem Theorem states that every set of sentences with an infinite model has an enumerable model. It, too, is a consequence of the completeness theorem: the proof of completeness generates a model for any consistent set of sentences, and that model is enumerable. There is also an Upward Löwenheim–Skolem Theorem, which guarantees that if a set of sentences has a denumerable model it also has a non-enumerable model. Both theorems fail in second-order logic.

explanation

Theorem 39.7. *The Löwenheim–Skolem Theorem fails for second-order logic: There are sentences with infinite models but no enumerable models.*

sol:met:lst:thm:sol-no-ls

Proof. Recall that

$$\text{Count} \equiv \exists z\, \exists u\, \forall X\, ((X(z) \wedge \forall x\, (X(x) \to X(u(x)))) \to \forall x\, X(x))$$

is true in a structure $\mathfrak{M}$ iff $|\mathfrak{M}|$ is enumerable, so $\neg$Count is true in $\mathfrak{M}$ iff $|\mathfrak{M}|$ is non-enumerable. There are such structures—take any non-enumerable set as the domain, e.g., $\wp(\mathbb{N})$ or $\mathbb{R}$. So $\neg$Count has infinite models but no enumerable models. □

Theorem 39.8. *There are sentences with denumerable but no non-enumerable models.*

Proof. Count $\wedge$ Inf is true in $\mathbb{N}$ but not in any structure $\mathfrak{M}$ with $|\mathfrak{M}|$ non-enumerable. □

Chapter 40

Second-order Logic and Set Theory

This section deals with coding powersets and the continuum in second-order logic. The results are stated but proofs have yet to be filled in. There are no problems yet—and the definitions and results themselves may have problems. Use with caution and report anything that's false or unclear.

content/second-order-logic/sol-and-set-theory/introduction.tex

40.1 Introduction

sol:set:int:sec

Since second-order logic can quantify over subsets of the domain as well as functions, it is to be expected that some amount, at least, of set theory can be carried out in second-order logic. By "carry out," we mean that it is possible to express set theoretic properties and statements in second-order logic, and is possible without any special, non-logical vocabulary for sets (e.g., the membership predicate symbol of set theory). For instance, we can define unions and intersections of sets and the subset relationship, but also compare the sizes of sets, and state results such as Cantor's Theorem.

---

content/second-order-logic/sol-and-set-theory/comparing-sets.tex

40.2 Comparing Sets

sol:set:cmp:sec

Proposition 40.1. *The formula* $\forall x\,(X(x) \to Y(x))$ *defines the subset relation, i.e.,* $\mathfrak{M}, s \vDash \forall x\,(X(x) \to Y(x))$ *iff* $s(X) \subseteq s(Y)$.

Proposition 40.2. *The formula* $\forall x\,(X(x) \leftrightarrow Y(x))$ *defines the identity relation on sets, i.e.,* $\mathfrak{M}, s \vDash \forall x\,(X(x) \leftrightarrow Y(x))$ *iff* $s(X) = s(Y)$.

Proposition 40.3. *The formula* $\exists x\, X(x)$ *defines the property of being non-empty, i.e.,* $\mathfrak{M}, s \vDash \exists x\, X(x)$ *iff* $s(X) \neq \emptyset$.

A set X is no larger than a set Y, $X \preceq Y$, iff there is an injective function $f\colon X \to Y$. Since we can express that a function is injective, and also that its values for arguments in X are in Y, we can also define the relation of being no larger than on subsets of the domain.

Proposition 40.4. *The formula*

$$\exists u\,(\forall x\,(X(x) \to Y(u(x))) \land \forall x\,\forall y\,(u(x) = u(y) \to x = y))$$

defines the relation of being no larger than.

Two sets are the same size, or "equinumerous," $X \approx Y$, iff there is a bijective function $f\colon X \to Y$.

Proposition 40.5. *The formula*

$$\begin{aligned}\exists u\,(&\forall x\,(X(x) \to Y(u(x))) \land \\ &\forall x\,\forall y\,(u(x) = u(y) \to x = y) \land \\ &\forall y\,(Y(y) \to \exists x\,(X(x) \land y = u(x))))\end{aligned}$$

defines the relation of being equinumerous with.

We will abbreviate these formulas, respectively, as $X \subseteq Y$, $X = Y$, $X \neq \emptyset$, $X \preceq Y$, and $X \approx Y$. (This may be slightly confusing, since we use the same notation when we speak informally about sets X and Y—but here the notation is an abbreviation for formulas in second-order logic involving one-place relation variables X and Y.)

Proposition 40.6. *The sentence* $\forall X \, \forall Y \, ((X \preceq Y \wedge Y \preceq X) \to X \approx Y)$ *is valid.*

Proof. The sentence is satisfied in a structure $\mathfrak{M}$ if, for any subsets $X \subseteq |\mathfrak{M}|$ and $Y \subseteq |\mathfrak{M}|$, if $X \preceq Y$ and $Y \preceq X$ then $X \approx Y$. But this holds for *any* sets X and Y—it is the Schröder-Bernstein Theorem. □

---

content/second-order-logic/sol-and-set-theory/cardinalities.tex

40.3 Cardinalities of Sets

explanation

Just as we can express that the domain is finite or infinite, enumerable or non-enumerable, we can define the property of a subset of $|\mathfrak{M}|$ being finite or infinite, enumerable or non-enumerable.

sol:set:crd:sec

Proposition 40.7. *The formula* $\mathrm{Inf}(X) \equiv$

$$\exists u \, (\forall x \, \forall y \, (u(x) = u(y) \to x = y) \wedge \exists y \, (X(y) \wedge \forall x \, (X(x) \to y \neq u(x))))$$

is satisfied with respect to a variable assignment s iff $s(X)$ is infinite.

Proposition 40.8. *The formula* $\mathrm{Count}(X) \equiv$

$$\exists z \, \exists u \, (X(z) \wedge \forall x \, (X(x) \to X(u(x))) \wedge \forall Y \, ((Y(z) \wedge \forall x \, (Y(x) \to Y(u(x)))) \to X = Y))$$

is satisfied with respect to a variable assignment s iff $s(X)$ is enumerable.

We know from Cantor's Theorem that there are non-enumerable sets, and in fact, that there are infinitely many different levels of infinite sizes. Set theory develops an entire arithmetic of sizes of sets, and assigns infinite cardinal numbers to sets. The natural numbers serve as the cardinal numbers measuring the sizes of finite sets. The cardinality of denumerable sets is the first infinite cardinality, called $\aleph_0$ ("aleph-nought" or "aleph-zero"). The next infinite size is $\aleph_1$. It is the smallest size a set can be without being countable (i.e., of size $\aleph_0$). We can define "X has size $\aleph_0$" as $\mathrm{Aleph}_0(X) \leftrightarrow \mathrm{Inf}(X) \wedge \mathrm{Count}(X)$. X has size $\aleph_1$ iff all its subsets are finite or have size $\aleph_0$, but is not itself of size $\aleph_0$. Hence we can express this by the formula $\mathrm{Aleph}_1(X) \equiv \forall Y \, (Y \subseteq$

$X \to (\neg \mathrm{Inf}(Y) \vee \mathrm{Aleph}_0(Y))) \wedge \neg \mathrm{Aleph}_0(X)$. Being of size $\aleph_2$ is defined similarly, etc.

There is one size of special interest, the so-called cardinality of the continuum. It is the size of $\wp(\mathbb{N})$, or, equivalently, the size of $\mathbb{R}$. That a set is the size of the continuum can also be expressed in second-order logic, but requires a bit more work.

---

content/second-order-logic/sol-and-set-theory/power-of-continuum.tex

40.4 The Power of the Continuum

sol:set:pow:sec

In second-order logic we can quantify over subsets of the domain, but not over sets of subsets of the domain. To do this directly, we would need *third-order* logic. For instance, if we wanted to state Cantor's Theorem that there is no injective function from the power set of a set to the set itself, we might try to formulate it as "for every set X, and every set P, if P is the power set of X, then not $P \preceq X$". And to say that P is the power set of X would require formalizing that the elements of P are all and only the subsets of X, so something like $\forall Y\, (P(Y) \leftrightarrow Y \subseteq X)$. The problem lies in $P(Y)$: that is not a formula of second-order logic, since only terms can be arguments to one-place relation variables like P.

explanation

We can, however, *simulate* quantification over sets of sets, if the domain is large enough. The idea is to make use of the fact that two-place relations R relates elements of the domain to elements of the domain. Given such an R, we can collect all the elements to which some x is R-related: $\{y \in |\mathfrak{M}| : R(x, y)\}$ is the set "coded by" x. Conversely, if $Z \subseteq \wp(|\mathfrak{M}|)$ is some collection of subsets of $|\mathfrak{M}|$, and there are at least as many elements of $|\mathfrak{M}|$ as there are sets in Z, then there is also a relation $R \subseteq |\mathfrak{M}|^2$ such that every $Y \in Z$ is coded by some x using R.

Definition 40.9. If $R \subseteq |\mathfrak{M}|^2$, then x *R-codes* $\{y \in |\mathfrak{M}| : R(x, y)\}$.

If an element $x \in |\mathfrak{M}|$ R-codes a set $Z \subseteq |\mathfrak{M}|$, then a set $Y \subseteq |\mathfrak{M}|$ codes a set of sets, namely the sets coded by the elements of Y. So a set Y can R-code $\wp(X)$. It does so iff for every $Z \subseteq X$, some $x \in Y$ R-codes Z, and every $x \in Y$ R-codes a $Z \subseteq X$.

Proposition 40.10. *The formula*

$$\mathrm{Codes}(x, R, Z) \equiv \forall y\, (Z(y) \leftrightarrow R(x, y))$$

expresses that $s(x)$ $s(R)$-codes $s(Z)$. The formula

$$\begin{aligned}\mathrm{Pow}(Y, R, X) \equiv\ & \forall Z\, (Z \subseteq X \to \exists x\, (Y(x) \wedge \mathrm{Codes}(x, R, Z))) \wedge \\ & \forall x\, (Y(x) \to \forall Z\, (\mathrm{Codes}(x, R, Z) \to Z \subseteq X))\end{aligned}$$

expresses that $s(Y)$ $s(R)$-codes the power set of $s(X)$, i.e., the elements of $s(Y)$ $s(R)$-code exactly the subsets of $s(X)$.

explanation

With this trick, we can express statements about the power set by quantifying over the codes of subsets rather than the subsets themselves. For instance, Cantor's Theorem can now be expressed by saying that there is no injective function from the domain of any relation that codes the power set of X to X itself.

Proposition 40.11. *The sentence*

$$\forall X\, \forall Y\, \forall R\, (\mathrm{Pow}(Y, R, X) \to \neg\exists u\, (\forall x\, \forall y\, (u(x) = u(y) \to x = y) \wedge \forall x\, (Y(x) \to X(u(x)))))$$

is valid.

explanation

The power set of a denumerable set is non-enumerable, and so its cardinality is larger than that of any denumerable set (which is $\aleph_0$). The size of $\wp(\mathbb{N})$ is called the "power of the continuum," since it is the same size as the points on the real number line, $\mathbb{R}$. If the domain is large enough to code the power set of a denumerable set, we can express that a set is the size of the continuum by saying that it is equinumerous with any set Y that codes the power set of set X of size $\aleph_0$. (If the domain is not large enough, i.e., it contains no subset equinumerous with $\mathbb{R}$, then there can also be no relation that codes $\wp(X)$.)

Proposition 40.12. *If $\mathbb{R} \preceq |\mathfrak{M}|$, then the formula*

$$\mathrm{Cont}(Y) \equiv \exists X\, \exists R\, ((\mathrm{Aleph}_0(X) \wedge \mathrm{Pow}(Y, R, X)) \wedge \forall x\, \forall y\, ((Y(x) \wedge Y(y) \wedge \forall z\, R(x, z) \leftrightarrow R(y, z)) \to x = y))$$

expresses that $s(Y) \approx \mathbb{R}$.

Proof. $\mathrm{Pow}(Y, R, X)$ expresses that $s(Y)$ $s(R)$-codes the power set of $s(X)$, which $\mathrm{Aleph}_0(X)$ says is countable. So $s(Y)$ is at least as large as the power of the continuum, although it may be larger (if multiple elements of $s(Y)$ code the same subset of X). This is ruled out be the last conjunct, which requires the association between elements of $s(Y)$ and subsets of $s(Z)$ via $s(R)$ to be injective. □

Proposition 40.13. $|\mathfrak{M}| \approx \mathbb{R}$ *iff*

$$\mathfrak{M} \vDash \exists X\, \exists Y\, \exists R\, (\mathrm{Aleph}_0(X) \wedge \mathrm{Pow}(Y, R, X) \wedge \exists u\, (\forall x\, \forall y\, (u(x) = u(y) \to x = y) \wedge \forall y\, (Y(y) \to \exists x\, y = u(x)))).$$

The Continuum Hypothesis is the statement that the size of the continuum is the first non-enumerable cardinality, i.e, that $\wp(\mathbb{N})$ has size $\aleph_1$. explanation

Proposition 40.14. *The Continuum Hypothesis is true iff*

$$\mathrm{CH} \equiv \forall X\,(\mathrm{Aleph}_1(X) \leftrightarrow \mathrm{Cont}(X))$$

is valid.

Note that it isn't true that $\neg$CH is valid iff the Continuum Hypothesis is false. In an enumerable domain, there are no subsets of size $\aleph_1$ and also no subsets of the size of the continuum, so CH is always true in an enumerable domain. However, we can give a different sentence that is valid iff the Continuum Hypothesis is false:

Proposition 40.15. *The Continuum Hypothesis is false iff*

$$\mathrm{NCH} \equiv \forall X\,(\mathrm{Cont}(X) \to \exists Y\,(Y \subseteq X \wedge \neg\mathrm{Count}(Y) \wedge \neg X \approx Y))$$

is valid.

Part IX

The Lambda Calculus

This part deals with the lambda calculus. The introduction chapter is based on Jeremy Avigad's notes; part of it is now redundant and covered in later chapters. The chapters on syntax, Church–Rosser property, and lambda definability were produced by Zesen Qian during his Mitacs summer internship. They still have to be reviewed and revised.

Chapter 41

Introduction

This chapter consists of Jeremy's original concise notes on the lambda calculus. The sections need to be combined, and the material on lambda definability merged with the material in the separate, more detailed chapter on lambda definability.

---

content/lambda-calculus/introduction/overview.tex

41.1 Overview

lam:int:ovr:sec

The lambda calculus was originally designed by Alonzo Church in the early 1930s as a basis for constructive logic, and *not* as a model of the computable functions. But it was soon shown to be equivalent to other definitions of computability, such as the Turing computable functions and the partial recursive functions. The fact that this initially came as a small surprise makes the characterization all the more interesting.

Lambda notation is a convenient way of referring to a function directly by a symbolic expression which defines it, instead of defining a name for it.

Instead of saying "let f be the function defined by $f(x) = x + 3$," one can say, "let f be the function $\lambda x.\,(x + 3)$." In other words, $\lambda x.\,(x + 3)$ is just a *name* for the function that adds three to its argument. In this expression, x is a dummy variable, or a placeholder: the same function can just as well be denoted by $\lambda y.\,(y+3)$. The notation works even with other parameters around. For example, suppose $g(x, y)$ is a function of two variables, and k is a natural number. Then $\lambda x.\,g(x, k)$ is the function which maps any x to $g(x, k)$.

This way of defining a function from a symbolic expression is known as *lambda abstraction*. The flip side of lambda abstraction is *application*: assuming one has a function f (say, defined on the natural numbers), one can *apply* it to any value, like 2. In conventional notation, of course, we write $f(2)$ for the result.

What happens when you combine lambda abstraction with application? Then the resulting expression can be simplified, by "plugging" the applicand in for the abstracted variable. For example,

$$(\lambda x.\,(x + 3))(2)$$

can be simplified to $2 + 3$.

Up to this point, we have done nothing but introduce new notations for conventional notions. The lambda calculus, however, represents a more radical departure from the set-theoretic viewpoint. In this framework:

1. Everything denotes a function.
2. Functions can be defined using lambda abstraction.
3. Anything can be applied to anything else.

For example, if F is a term in the lambda calculus, $F(F)$ is always assumed to be meaningful. This liberal framework is known as the *untyped* lambda calculus, where "untyped" means "no restriction on what can be applied to what."

digression

There is also a *typed* lambda calculus, which is an important variation on the untyped version. Although in many ways the typed lambda calculus is similar to the untyped one, it is much easier to reconcile with a classical set-theoretic framework, and has some very different properties.

Research on the lambda calculus has proved to be central in theoretical computer science, and in the design of programming languages. LISP, designed by John McCarthy in the 1950s, is an early example of a language that was influenced by these ideas.

---

content/lambda-calculus/introduction/syntax.tex

41.2 The Syntax of the Lambda Calculus

lam:int:syn:sec

One starts with a sequence of variables x, y, z, ... and some constant symbols a, b, c, ... The set of terms is defined inductively, as follows:

1. Each variable is a term.

2. Each constant is a term.

3. If M and N are terms, so is (MN).

4. If M is a term and x is a variable, then $(\lambda x.\, M)$ is a term.

Terms of the form (MN) are called *applications* and those of the form $(\lambda x.\, M)$ *abstractions*.

The system without any constants at all is called the *pure* lambda calculus. We'll mainly be working in the pure λ-calculus, so all lowercase letters will stand for variables. We use uppercase letters (M, N, etc.) to stand for terms of the λ-calculus.

We will follow a few notational conventions:

Convention 1.

1. When parentheses are left out, application takes place from left to right. For example, if M, N, P, and Q are terms, then $MNPQ$ abbreviates $(((MN)P)Q)$.

2. Again, when parentheses are left out, lambda abstraction is to be given the widest scope possible. From example, $\lambda x.\, MNP$ is read $(\lambda x.\, ((MN)P))$.

3. A lambda can be used to abstract multiple variables. For example, $\lambda xyz.\, M$ is short for $\lambda x.\, \lambda y.\, \lambda z.\, M$.

For example,

$$\lambda xy.\, xxyx\lambda z.\, xz$$

abbreviates

$$\lambda x.\, \lambda y.\, ((((xx)y)x)(\lambda z.\, (xz))).$$

You should memorize these conventions. They will drive you crazy at first, but you will get used to them, and after a while they will drive you less crazy than having to deal with a morass of parentheses.

Two terms that differ only in the names of the bound variables are called α-equivalent; for example, $\lambda x.\, x$ and $\lambda y.\, y$. It will be convenient to think of these as being the "same" term; in other words, when we say that M and N are the same, we also mean "up to renamings of the bound variables." Variables that are in the scope of a λ are called "bound", while others are called "free." There are no free variables in the previous example; but in

$$(\lambda z.\, yz)x$$

y and x are free, and z is bound.

41.3 Reduction of Lambda Terms

lam:int:red:sec

What can one do with lambda terms? Simplify them. If M and N are any lambda terms and x is any variable, we can use $M[N/x]$ to denote the result of substituting N for x in M, after renaming any bound variables of M that would interfere with the free variables of N after the substitution. For example,

$$(\lambda w.\, xxw)[yyz/x] = \lambda w.\, (yyz)(yyz)w.$$

Alternative notations for substitution are $[N/x]M$, $[x/N]M$, and also $M[x/N]$ digression Beware!

Intuitively, $(\lambda x.\, M)N$ and $M[N/x]$ have the same meaning; the act of replacing the first term by the second is called *β-contraction*. $(\lambda x.\, M)N$ is called a *redex* and $M[N/x]$ its *contractum*. Generally, if it is possible to change a term P to P' by β-contraction of some subterm, we say that P *β-reduces to P' in one step*, and write $P \to P'$. If from P we can obtain P' with some number of one-step reductions (possibly none), then P *β-reduces* to P'; in symbols, $P \twoheadrightarrow P'$. A term that cannot be β-reduced any further is called *β-irreducible*, or *β-normal*. We will say "reduces" instead of "β-reduces," etc., when the context is clear.

Let us consider some examples.

1. We have

$$\begin{aligned}(\lambda x.\, xxy)\lambda z.\, z &\to (\lambda z.\, z)(\lambda z.\, z)y \\ &\to (\lambda z.\, z)y \\ &\to y.\end{aligned}$$

2. "Simplifying" a term can make it more complex:

$$\begin{aligned}(\lambda x.\, xxy)(\lambda x.\, xxy) &\to (\lambda x.\, xxy)(\lambda x.\, xxy)y \\ &\to (\lambda x.\, xxy)(\lambda x.\, xxy)yy \\ &\to \ldots\end{aligned}$$

3. It can also leave a term unchanged:

$$(\lambda x.\, xx)(\lambda x.\, xx) \to (\lambda x.\, xx)(\lambda x.\, xx).$$

4. Also, some terms can be reduced in more than one way; for example,

$$(\lambda x.\, (\lambda y.\, yx)z)v \to (\lambda y.\, yv)z$$

by contracting the outermost application; and

$$(\lambda x.\, (\lambda y.\, yx)z)v \to (\lambda x.\, zx)v$$

by contracting the innermost one. Note, in this case, however, that both terms further reduce to the same term, zv.

The final outcome in the last example is not a coincidence, but rather illustrates a deep and important property of the lambda calculus, known as the "Church–Rosser property."

---

content/lambda-calculus/introduction/church-rosser.tex

41.4 The Church–Rosser Property

lam:int:cr:sec
lam:int:cr:thm:church-rosser

Theorem 41.1. *Let M, N_1, and N_2 be terms, such that $M \twoheadrightarrow N_1$ and $M \twoheadrightarrow N_2$. Then there is a term P such that $N_1 \twoheadrightarrow P$ and $N_2 \twoheadrightarrow P$.*

Corollary 41.2. *Suppose M can be reduced to normal form. Then this normal form is unique.*

Proof. If $M \twoheadrightarrow N_1$ and $M \twoheadrightarrow N_2$, by the previous theorem there is a term P such that N_1 and N_2 both reduce to P. If N_1 and N_2 are both in normal form, this can only happen if $N_1 \equiv P \equiv N_2$. □

Finally, we will say that two terms M and N are *β-equivalent*, or just *equivalent*, if they reduce to a common term; in other words, if there is some P such that $M \twoheadrightarrow P$ and $N \twoheadrightarrow P$. This is written $M \overset{\beta}{=} N$. Using Theorem 41.1, you can check that $\overset{\beta}{=}$ is an equivalence relation, with the additional property that for every M and N, if $M \twoheadrightarrow N$ or $N \twoheadrightarrow M$, then $M \overset{\beta}{=} N$. (In fact, one can show that $\overset{\beta}{=}$ is the *smallest* equivalence relation having this property.)

---

content/lambda-calculus/introduction/currying.tex

41.5 Currying

lam:rep:cur:sec

A λ-abstract $\lambda x.\, M$ represents a function of one argument, which is quite a limitation when we want to define function accepting multiple arguments. One way to do this would be by extending the λ-calculus to allow the formation of pairs, triples, etc., in which case, say, a three-place function $\lambda x.\, M$ would expect its argument to be a triple. However, it is more convenient to do this by *Currying*.

Let's consider an example. We'll pretend for a moment that we have a $+$ operation in the λ-calculus. The addition function is 2-place, i.e., it takes two arguments. But a λ-abstract only gives us functions of one argument: the syntax does not allow expressions like $\lambda(x, y).\, (x+y)$. However, we can consider the one-place function $f_x(y)$ given by $\lambda y.\, (x + y)$, which adds x to its single argument y. Actually, this is not a single function, but a family of different functions "add x," one for each number x. Now we can define another one-place function g as $\lambda x.\, f_x$. Applied to argument x, $g(x)$ returns the function f_x—so its values are other functions. Now if we apply g to x, and then the result to y

we get: $(g(x))y = f_x(y) = x + y$. In this way, the one-place function g can do the same job as the two-place addition function. "Currying" simply refers to this trick for turning two-place functions into one place functions (whose values are one-place functions).

Here is an example properly in the syntax of the λ-calculus. How do we represent the function $f(x, y) = x$? If we want to define a function that accepts two arguments and returns the first, we can write $\lambda x.\, \lambda y.\, x$, which literally is a function that accepts an argument x and returns the function $\lambda y.\, x$. The function $\lambda y.\, x$ accepts another argument y, but drops it, and always returns x. Let's see what happens when we apply $\lambda x.\, \lambda y.\, x$ to two arguments:

$$\begin{aligned}(\lambda x.\, \lambda y.\, x)MN &\xrightarrow{\beta}(\lambda y.\, M)N\\ &\xrightarrow{\beta}M\end{aligned}$$

In general, to write a function with parameters $x_1, \ldots, x_n$ defined by some term N, we can write $\lambda x_1.\, \lambda x_2.\, \ldots \lambda x_n.\, N$. If we apply n arguments to it we get:

$$\begin{aligned}(\lambda x_1.\, \lambda x_2.\, \ldots \lambda x_n.\, N)M_1 \ldots M_n &\xrightarrow{\beta}\\ &\xrightarrow{\beta} ((\lambda x_2.\, \ldots \lambda x_n.\, N)[M_1/x_1])M_2 \ldots M_n\\ &\equiv (\lambda x_2.\, \ldots \lambda x_n.\, N[M_1/x_1])M_2 \ldots M_n\\ &\;\;\vdots\\ &\xrightarrow{\beta} P[M_1/x_1] \ldots [M_n/x_n]\end{aligned}$$

The last line literally means substituting M_i for x_i in the body of the function definition, which is exactly what we want when applying multiple arguments to a function.

---

content/lambda-calculus/introduction/lambda-definability.tex

41.6 λ-Definable Arithmetical Functions

lam:int:rep:sec

How can the lambda calculus serve as a model of computation? At first, it is not even clear how to make sense of this statement. To talk about computability on the natural numbers, we need to find a suitable representation for such numbers. Here is one that works surprisingly well.

Definition 41.3. For each natural number n, define the *Church numeral* $\overline{n}$ to be the lambda term $\lambda x.\, \lambda y.\, (x(x(x(\ldots x(y)))))$, where there are n x's in all.

The terms $\overline{n}$ are "iterators": on input f, $\overline{n}$ returns the function mapping y to $f^n(y)$. Note that each numeral is normal. We can now say what it means for a lambda term to "compute" a function on the natural numbers.

Definition 41.4. Let $f(x_0, \ldots, x_{k-1})$ be an n-ary partial function from $\mathbb{N}$ to $\mathbb{N}$. We say a λ-term F *λ-defines* f iff for every sequence of natural numbers $n_0, \ldots, n_{k-1}$,

$$F\,\overline{n_0}\,\overline{n_1}\ldots\overline{n_{k-1}} \twoheadrightarrow \overline{f(n_0, n_1, \ldots, n_{k-1})}$$

if $f(n_0, \ldots, n_{k-1})$ is defined, and $F, \overline{n_0}\,\overline{n_1}\ldots\overline{n_{k-1}}$ has no normal form otherwise.

Theorem 41.5. *A function f is a partial computable function if and only if it is λ-defined by a lambda term.* lam:int:rep: thm:lambda-def

explanation This theorem is somewhat striking. As a model of computation, the lambda calculus is a rather simple calculus; the only operations are lambda abstraction and application! From these meager resources, however, it is possible to implement any computational procedure.

---

content/lambda-calculus/introduction/lambda-computable.tex

41.7 λ-Definable Functions are Computable

lam:int:cmp: sec

Theorem 41.6. *If a partial function f is λ-defined by a lambda term, it is computable.* lam:int:cmp: thm:lambda-computable

Proof. Suppose a function f is λ-defined by a lambda term X. Let us describe an informal procedure to compute f. On input $m_0, \ldots, m_{n-1}$, write down the term $X\overline{m}_0 \ldots \overline{m}_{n-1}$. Build a tree, first writing down all the one-step reductions of the original term; below that, write all the one-step reductions of those (i.e., the two-step reductions of the original term); and keep going. If you ever reach a numeral, return that as the answer; otherwise, the function is undefined.

An appeal to Church's thesis tells us that this function is computable. A better way to prove the theorem would be to give a recursive description of this search procedure. For example, one could define a sequence primitive recursive functions and relations, "IsASubterm," "Substitute," "ReducesToInOneStep," "ReductionSequence," "Numeral," etc. The partial recursive procedure for computing $f(m_0, \ldots, m_{n-1})$ is then to search for a sequence of one-step reductions starting with $X\overline{m_0}\ldots\overline{m_{n-1}}$ and ending with a numeral, and return the number corresponding to that numeral. The details are long and tedious but otherwise routine. □

---

content/lambda-calculus/introduction/computable-lambda.tex

41.8 Computable Functions are λ-Definable

lam:int:lrp: sec

Theorem 41.7. *Every computable partial function is λ-definable.* lam:int:lrp: thm:computable-lambda

Proof. We need to show that every partial computable function f is λ-defined by a lambda term F. By Kleene's normal form theorem, it suffices to show that every primitive recursive function is λ-defined by a lambda term, and then that the functions λ-definable are closed under suitable compositions and unbounded search. To show that every primitive recursive function is λ-defined by a lambda term, it suffices to show that the initial functions are λ-definable, and that the partial functions that are λ-definable are closed under composition, primitive recursion, and unbounded search. □

We will use a more conventional notation to make the rest of the proof more readable. For example, we will write $M(x, y, z)$ instead of $Mxyz$. While this is suggestive, you should remember that terms in the untyped lambda calculus do not have associated arities; so, for the same term M, it makes just as much sense to write $M(x, y)$ and $M(x, y, z, w)$. But using this notation indicates that we are treating M as a function of three variables, and helps make the intentions behind the definitions clearer. In a similar way, we will say "define M by $M(x, y, z) = \ldots$" instead of "define M by $M = \lambda x.\, \lambda y.\, \lambda z.\, \ldots$."

---

content/lambda-calculus/introduction/basic-pr-lambda.tex

41.9 The Basic Primitive Recursive Functions are λ-Definable

lam:int:bas:sec

Lemma 41.8. *The functions* zero, succ, *and* P_i^n *are λ-definable.*

Proof. zero is just $\lambda x.\, \lambda y.\, y$.

The successor function succ, is defined by $\mathrm{Succ}(u) = \lambda x.\, \lambda y.\, x(uxy)$. You should think about why this works; for each numeral $\overline{n}$, thought of as an iterator, and each function f, $\mathrm{Succ}(\overline{n}, f)$ is a function that, on input y, applies f n times starting with y, and then applies it once more.

There is nothing to say about projections: $\mathrm{Proj}_i^n(x_0, \ldots, x_{n-1}) = x_i$. In other words, by our conventions, Proj_i^n is the lambda term $\lambda x_0.\, \ldots \lambda x_{n-1}.\, x_i$.□

---

content/lambda-calculus/introduction/composition.tex

41.10 The λ-Definable Functions are Closed under Composition

lam:int:com:sec

Lemma 41.9. *The λ-definable functions are closed under composition.*

Proof. Suppose f is defined by composition from h, g_0, …, g_{k-1}. Assuming h, g_0, …, g_{k-1} are λ-defined by H, G_0, …, G_{k-1}, respectively, we need to find a term F that λ-defines f. But we can simply define F by

$$F(x_0, \ldots, x_{l-1}) = H(G_0(x_0, \ldots, x_{l-1}), \ldots, G_{k-1}(x_0, \ldots, x_{l-1})).$$

In other words, the language of the lambda calculus is well suited to represent composition. □

---

content/lambda-calculus/introduction/primitive-recursion.tex

41.11 λ-Definable Functions are Closed under Primitive Recursion

lam:int:pr:sec

When it comes to primitive recursion, we finally need to do some work. We will have to proceed in stages. As before, on the assumption that we already have terms G and H that λ-define functions g and h, respectively, we want a term H that λ-defines the function f defined by

$$f(0, \vec{z}) = g(\vec{z})$$
$$f(x+1, \vec{z}) = h(z, f(x, \vec{z}), \vec{z}).$$

So, in general, given lambda terms G' and H', it suffices to find a term F such that

$$F(\overline{0}, \vec{z}) \equiv G(\vec{z})$$
$$F(\overline{n+1}, \vec{z}) \equiv H(\overline{n}, F(\overline{n}, \vec{z}), \vec{z})$$

for every natural number n; the fact that G' and H' λ-define g and h means that whenever we plug in numerals $\overline{\vec{m}}$ for $\vec{z}$, $F(\overline{n+1}, \overline{\vec{m}})$ will normalize to the right answer.

But for this, it suffices to find a term F satisfying

$$F(\overline{0}) \equiv G$$
$$F(\overline{n+1}) \equiv H(\overline{n}, F(\overline{n}))$$

for every natural number n, where

$$G = \lambda\vec{z}.\, G'(\vec{z}) \text{ and}$$
$$H(u, v) = \lambda\vec{z}.\, H'(u, v(u, \vec{z}), \vec{z}).$$

In other words, with lambda trickery, we can avoid having to worry about the extra parameters $\vec{z}$—they just get absorbed in the lambda notation.

Before we define the term F, we need a mechanism for handling ordered pairs. This is provided by the next lemma.

Lemma 41.10. *There is a lambda term D such that for each pair of lambda terms M and N, $D(M,N)(\overline{0}) \twoheadrightarrow M$ and $D(M,N)(\overline{1}) \twoheadrightarrow N$.*

Proof. First, define the lambda term K by

$$K(y) = \lambda x.\, y.$$

In other words, K is the term $\lambda y.\, \lambda x.\, y$. Looking at it differently, for every M, $K(M)$ is a constant function that returns M on any input.

Now define $D(x,y,z)$ by $D(x,y,z) = z(K(y))x$. Then we have

$$\begin{aligned} D(M,N,\overline{0}) &\twoheadrightarrow \overline{0}(K(N))M \twoheadrightarrow M \text{ and} \\ D(M,N,\overline{1}) &\twoheadrightarrow \overline{1}(K(N))M \twoheadrightarrow K(N)M \twoheadrightarrow N, \end{aligned}$$

as required. □

The idea is that $D(M,N)$ represents the pair $\langle M,N\rangle$, and if P is assumed to represent such a pair, $P(\overline{0})$ and $P(\overline{1})$ represent the left and right projections, $(P)_0$ and $(P)_1$. We will use the latter notations.

Lemma 41.11. *The λ-definable functions are closed under primitive recursion.*

Proof. We need to show that given any terms, G and H, we can find a term F such that

$$\begin{aligned} F(\overline{0}) &\equiv G \\ F(\overline{n+1}) &\equiv H(\overline{n}, F(\overline{n})) \end{aligned}$$

for every natural number n. The idea is roughly to compute sequences of *pairs*

$$\langle \overline{0}, F(\overline{0})\rangle, \langle \overline{1}, F(\overline{1})\rangle, \ldots,$$

using numerals as iterators. Notice that the first pair is just $\langle \overline{0}, G\rangle$. Given a pair $\langle \overline{n}, F(\overline{n})\rangle$, the next pair, $\langle \overline{n+1}, F(\overline{n+1})\rangle$ is supposed to be equivalent to $\langle \overline{n+1}, H(\overline{n}, F(\overline{n}))\rangle$. We will design a lambda term T that makes this one-step transition.

The details are as follows. Define $T(u)$ by

$$T(u) = \langle S((u)_0), H((u)_0, (u)_1)\rangle.$$

Now it is easy to verify that for any number n,

$$T(\langle \overline{n}, M\rangle) \twoheadrightarrow \langle \overline{n+1}, H(\overline{n}, M)\rangle.$$

As suggested above, given G and H, define $F(u)$ by

$$F(u) = (u(T, \langle \overline{0}, G\rangle))_1.$$

In other words, on input $\overline{n}$, F iterates T n times on $\langle \overline{0}, G\rangle$, and then returns the second component. To start with, we have

1. $\overline{0}(T, \langle \overline{0}, G\rangle) \equiv \langle \overline{0}, G\rangle$
2. $F(\overline{0}) \equiv G$

By induction on n, we can show that for each natural number one has the following:

1. $\overline{n+1}(T, \langle \overline{0}, G\rangle) \equiv \langle \overline{n+1}, F(\overline{n+1})\rangle$
2. $F(\overline{n+1}) \equiv H(\overline{n}, F(\overline{n}))$

For the second clause, we have

$$\begin{aligned} F(\overline{n+1}) &\twoheadrightarrow (\overline{n+1}(T, \langle \overline{0}, G\rangle))_1 \\ &\equiv (T(\overline{n}(T, \langle \overline{0}, G\rangle)))_1 \\ &\equiv (T(\langle \overline{n}, F(\overline{n})\rangle))_1 \\ &\equiv (\langle \overline{n+1}, H(\overline{n}, F(\overline{n}))\rangle)_1 \\ &\equiv H(\overline{n}, F(\overline{n})). \end{aligned}$$

Here we have used the induction hypothesis on the second-to-last line. For the first clause, we have

$$\begin{aligned} \overline{n+1}(T, \langle \overline{0}, G\rangle) &\equiv T(\overline{n}(T, \langle \overline{0}, G\rangle)) \\ &\equiv T(\langle \overline{n}, F(\overline{n})\rangle) \\ &\equiv \langle \overline{n+1}, H(\overline{n}, F(\overline{n}))\rangle \\ &\equiv \langle \overline{n+1}, F(\overline{n+1})\rangle. \end{aligned}$$

Here we have used the second clause in the last line. So we have shown $F(\overline{0}) \equiv G$ and, for every n, $F(\overline{n+1}) \equiv H(\overline{n}, F(\overline{n}))$, which is exactly what we needed.□

---

content/lambda-calculus/introduction/fixed-point-combinator.tex

41.12 Fixed-Point Combinators

lam:int:fix:sec

Suppose you have a lambda term g, and you want another term k with the property that k is β-equivalent to gk. Define terms

$$\mathrm{diag}(x) = xx$$

and

$$l(x) = g(\mathrm{diag}(x))$$

using our notational conventions; in other words, l is the term $\lambda x.\, g(xx)$. Let k be the term ll. Then we have

$$\begin{aligned} k &= (\lambda x.\, g(xx))(\lambda x.\, g(xx)) \\ &\twoheadrightarrow g((\lambda x.\, g(xx))(\lambda x.\, g(xx))) \\ &= gk. \end{aligned}$$

If one takes

$$Y = \lambda g.\,((\lambda x.\,g(xx))(\lambda x.\,g(xx)))$$

then Yg and $g(Yg)$ reduce to a common term; so $Yg \equiv_\beta g(Yg)$. This is known as "Curry's combinator." If instead one takes

$$Y = (\lambda xg.\,g(xxg))(\lambda xg.\,g(xxg))$$

then in fact Yg reduces to $g(Yg)$, which is a stronger statement. This latter version of Y is known as "Turing's combinator."

---

content/lambda-calculus/introduction/minimization.tex

41.13 The λ-Definable Functions are Closed under Minimization

lam:int:min:sec

Lemma 41.12. *Suppose $f(x, y)$ is λ-definable. Let g be defined by*

$$g(x) \simeq \mu y\ f(x, y).$$

Then g is λ-definable.

Proof. The idea is roughly as follows. Given x, we will use the fixed-point lambda term Y to define a function $h_x(n)$ which searches for a y starting at n; then $g(x)$ is just $h_x(0)$. The function h_x can be expressed as the solution of a fixed-point equation:

$$h_x(n) \simeq \begin{cases} n & \text{if } f(x, n) = 0 \\ h_x(n+1) & \text{otherwise.} \end{cases}$$

Here are the details. Since f is primitive recursive, it is λ-defined by some term F. Remember that we also have a lambda term D, such that $D(M, N, \bar{0}) \twoheadrightarrow M$ and $D(M, N, \bar{1}) \twoheadrightarrow N$. Fixing x for the moment, to λ-define h_x we want to find a term H (depending on x) satisfying

$$H(\overline{n}) \equiv D(\overline{n}, H(S(\overline{n})), F(x, \overline{n})).$$

We can do this using the fixed-point term Y. First, let U be the term

$$\lambda h.\,\lambda z.\,D(z, (h(Sz)), F(x, z)),$$

and then let H be the term YU. Notice that the only free variable in H is x. Let us show that H satisfies the equation above.

By the definition of Y, we have

$$H = YU \equiv U(YU) = U(H).$$

In particular, for each natural number n, we have

$$\begin{aligned} H(\overline{n}) &\equiv U(H, \overline{n}) \\ &\twoheadrightarrow D(\overline{n}, H(S(\overline{n})), F(x, \overline{n})), \end{aligned}$$

as required. Notice that if you substitute a numeral $\overline{m}$ for x in the last line, the expression reduces to $\overline{n}$ if $F(\overline{m}, \overline{n})$ reduces to $\overline{0}$, and it reduces to $H(S(\overline{n}))$ if $F(\overline{m}, \overline{n})$ reduces to any other numeral.

To finish off the proof, let G be $\lambda x.\, H(\overline{0})$. Then G λ-defines g; in other words, for every m, $G(\overline{m})$ reduces to $\overline{g(m)}$, if $g(m)$ is defined, and has no normal form otherwise. □

Chapter 42

Syntax

---

content/lambda-calculus/syntax/terms.tex

42.1 Terms

lam:syn:trm:sec

The terms of the lambda calculus are built up inductively from an infinite supply of variables v_0, v_1, ..., the symbol "λ", and parentheses. We will use x, y, z, ... to designate variables, and M, N, P, ... to desginate terms.

lam:syn:trm:defn:term

Definition 42.1 (Terms). The set of *terms* of the lambda calculus is defined inductively by:

1. If x is a variable, then x is a term. lam:syn:trm:defn:term-var
2. If x is a variable and M is a term, then $(\lambda x.\, M)$ is a term. lam:syn:trm:defn:term-abs
3. If both M and N are terms, then (MN) is a term. lam:syn:trm:defn:term-app

If a term $(\lambda x.\, M)$ is formed according to (2) we say it is the result of an *abstraction*, and the x in λx is called a *parameter*. A term (MN) formed according to (3) is the result of an *application*.

The terms defined above are fully parenthesized. This can get rather cumbersome, as the term $(\lambda x.\, ((\lambda x.\, x)(\lambda x.\, (xx))))$ demnostrates. We will introduce

conventions for avoiding parentheses. However, the official definition makes it easy to determine how a term is constructed according to Definition 42.1. For example, the last step of forming the term $(\lambda x.\,((\lambda x.\,x)(\lambda x.\,(xx))))$ must be abstraction where the parameter is x. It results by abstraction from the term $((\lambda x.\,x)(\lambda x.\,(xx)))$, which is an application of two terms. Each of these two terms is the result of an abstraction, and so on.

Problem 42.1. Describe the formation of $(\lambda g.\,(\lambda x.\,(g(xx)))(\lambda x.\,(g(xx))))$.

---

content/lambda-calculus/syntax/unique-readability.tex

42.2 Unique Readability

lam:syn:unq:sec

We may wonder if for each term there is a unique way of forming it, and there is. For each lambda term there is only one way to construct and interpret it. In the following discussion, a *formation* is the procedure of constructing a term using the formation rules (one or several times) of Definition 42.1.

lam:syn:unq:lem:term-start

Lemma 42.2. *A term starts with either a variable or a parenthesis.*

Proof. Something counts as a term only if it is constructed according to Definition 42.1. If it is the result of (1), it must be a variable. If it is the result of (2) or (3), it starts with a parenthesis. □

lam:syn:unq:lem:app-start

Lemma 42.3. *The result of an application starts with either two parentheses or a parenthesis and a variable.*

Proof. If M is the result of an application, it is of the form (PQ), so it begins with a parenthesis. Since P is a term, by Lemma 42.2, it begins either with a parenthesis or a variable. □

lam:syn:unq:lem:initial

Lemma 42.4. *No proper initial part of a term is itself a term.*

Problem 42.2. Prove Lemma 42.4 by induction on the length of terms.

lam:syn:unq:prop:unq

Proposition 42.5 (Unique Readability). *There is a unique formation for each term. In other words, if a term M is formed by a formation, then it is the only formation that can form this term.*

Proof. We prove this by induction on the formation of terms.

1. M is of the form x, where x is some variable. Since the results of abstractions and applications always start with parentheses, they cannot have been used to construct M; Thus, the formation of M must be a single step of Definition 42.1(1).

2. M is of the form $(\lambda x.\, N)$, where x is some variable and N is a term. It could not have been constructed according to Definition 42.1(1), because it is not a single variable. It is not the result of an application, by Lemma 42.3. Thus M can only be the result of an abstraction on N. By inductive hypothesis we know that formation of N is itself unique.

3. M is of the form (PQ), where P and Q are terms. Since it starts with a parentheses, it cannot also be constructed by Definition 42.1(1). By Lemma 42.2, P cannot begin with λ, so (PQ) cannot be the result of an abstraction. Now suppose there were another way of constructing M by application, e.g., it is also of the form $(P'Q')$. Then P is a proper initial segment of P' (or vice versa), and this is impossible by Lemma 42.4. So P and Q are uniquely determined, and by inductive hypothesis we know that formations of P and Q is unique. □

A more readable paraphrase of the above proposition is as follows:

Proposition 42.6. *A term M can only be one of the following forms:*

1. *x, where x is a variable uniquely determined by M.*
2. *$(\lambda x.\, N)$, where x is a variable and N is another term, both of which is uniquely determined by M.*
3. *(PQ), where P and Q are two terms uniquely determined by M.*

---

content/lambda-calculus/syntax/abbreviated-syntax.tex

42.3 Abbreviated Syntax

lam:syn:abb:sec

Terms as defined in Definition 42.1 are sometimes cumbersome to write, so it is useful to introduce a more concise syntax. We must of course be careful to make sure that the terms in the concise notation also are uniquely readable. One widely used version called *abbreviated terms* is as follows.

1. When parentheses are left out, application takes place from left to right. For example, if M, N, P, and Q are terms, then $MNPQ$ abbreviates $(((MN)P)Q)$.
2. Again, when parentheses are left out, lambda abstraction is given the widest scope possible. From example, $\lambda x.\, MNP$ is read as $(\lambda x.\, MNP)$.
3. A lambda can be used to abstract multiple variables. For example, $\lambda xyz.\, M$ is short for $\lambda x.\, \lambda y.\, \lambda z.\, M$.

For example,

$$\lambda xy.\, xxyx\lambda z.\, xz$$

abbreviates

$$(\lambda x.\, (\lambda y.\, ((((xx)y)x)(\lambda z.\, (xz))))).$$

Problem 42.3. Expand the abbreviated term $\lambda g.\,(\lambda x.\,g(xx))\lambda x.\,g(xx)$.

---

content/lambda-calculus/syntax/free-variables.tex

42.4 Free Variables

lam:syn:fv:sec

Lambda calculus is about functions, and lambda abstraction is how functions arise. Intuitively, $\lambda x.\,M$ is the function with values given by M when the argument to the function is assigned to x. But not every occurrence of x in M is relevant: if M contains another abstract $\lambda x.\,N$ then the occurrences of x in N are relevant to $\lambda x.\,N$ but not to $\lambda x.\,M$. So, a lambda abstract λx inside $\lambda x.\,M$ *binds* those occurrences of x in M that are not already bound by another lambda abstract—the *free* occurrences of x in M.

Definition 42.7 (Scope). If $\lambda x.\,M$ occurs inside a term N, then the corresponding occurrence of N is the *scope* of the λx.

Definition 42.8 (Free and bound occurrence). An occurrence of variable x in a term M is *free* if it is not in the scope of a λx, and *bound* otherwise. An occurrence of a variable x in $\lambda x.\,M$ is bound by the initial λx iff the occurrence of x in M is free.

Example 42.9. In $\lambda x.\,xy$, both x and y are in the scope of λx, so x is bound by λx. Since y is not in the scope of any λy, it is free. In $\lambda x.\,xx$, both occurrences of x are bound by λx, since both are free in xx. In $((\lambda x.\,xx)x)$, the last occurrence of x is free, since it is not in the scope of a λx. In $\lambda x.\,(\lambda x.\,x)x$, the scope of the first λx is $(\lambda x.\,x)x$ and the scope of the second λx is the second-to-last occurrence of x. In $(\lambda x.\,x)x$, the last occurrence of x is free, and the second-to-last is bound. Thus, the second-to-last occurrence of x in $\lambda x.\,(\lambda x.\,x)x$ is bound by the second λx, and the last occurrence by the first λx.

For a term P, we can check all variable occurrences in it and get a set of free variables. This set is denoted by $\mathrm{FV}(P)$ with a natural definition as follows:

lam:syn:fv:def:fv

Definition 42.10 (Free variables of a term). The set of *free variables* of a term is defined inductively by:

lam:syn:fv:def:fv1

1. $\mathrm{FV}(x) = \{x\}$

lam:syn:fv:def:fv2

2. $\mathrm{FV}(\lambda x.\,N) = \mathrm{FV}(N) \setminus \{x\}$

lam:syn:fv:def:fv3

3. $\mathrm{FV}(PQ) = \mathrm{FV}(P) \cup \mathrm{FV}(Q)$

Problem 42.4.

1. Identify the scopes of λg and the two λx in this term: $\lambda g.\,(\lambda x.\,g(xx))\lambda x.\,g(xx)$.
2. In $\lambda g.\,(\lambda x.\,g(xx))\lambda x.\,g(xx)$, are all occurrences of variables bound? By which abstractions are they bound respectively?

3. Give $\mathrm{FV}(\lambda x.\,(\lambda y.\,(\lambda z.\,xy)z)y)$

explanation A free variable is like a reference to the outside world (the *environment*), and a term containing free variables can be seen as a partially specified term, since its behaviour depends on how we set up the environment. For example, in the term $\lambda x.\,fx$, which accepts an argument x and returns f of that argument, the variable f is free. This value of the term is dependent on the environment it is in, in particular the value of f in that environment.

If we apply abstraction to this term, we get $\lambda f.\,\lambda x.\,fx$. This term is no longer dependent on the environment variable f, because it now designates a function that accepts two arguments and returns the result of applying the first to the second. Changing f in the environment won't have any effect on the behavior of this term, as the term will only use whatever is passed as an argument, and not the value of f in the environment.

Definition 42.11 (Closed term, combinator). A term with no free variables is called a *closed term*, or a *combinator*.

Lemma 42.12. lam:syn:fv: lem:fv

1. *If* $y \neq x$, *then* $y \in \mathrm{FV}(\lambda x.\,N)$ *iff* $y \in \mathrm{FV}(N)$. lam:syn:fv: lem:fv-abs
2. $y \in \mathrm{FV}(PQ)$ *iff* $y \in \mathrm{FV}(P)$ *or* $y \in \mathrm{FV}(Q)$. lam:syn:fv: lem:fv-app

Proof. Exercise. □

Problem 42.5. Prove Lemma 42.12.

---

content/lambda-calculus/syntax/substitution.tex

42.5 Substitution

explanation Free variables are references to environment variables, thus it makes sense to actually use a specific value in the place of a free variable. For example, we may want to replace f in $\lambda x.\,fx$ with a specific term, like the identity function $\lambda y.\,y$. This results in $\lambda x.\,(\lambda y.\,y)x$. The process of replacing free variables with lambda terms is called substitution. lam:syn:sub: sec

Definition 42.13 (Substitution). The *substitution* of a term N for a variable x in a term M, $M[N/x]$, is defined inductively by: lam:syn:sub: defn:substitution

1. $x[N/x] = N$. lam:syn:sub: defn:substitution-1
2. $y[N/x] = y$ if $x \neq y$. lam:syn:sub: defn:substitution-2
3. $PQ[N/x] = (P[N/x])(Q[N/x])$. lam:syn:sub: def:substitution-3

4. $(\lambda y.\, P)[N/x] = \lambda y.\, P[N/x]$, if $x \neq y$ and $y \notin \mathrm{FV}(N)$, otherwise undefined.

lam:syn:sub: defn:substitution-4

In Definition 42.13(4), we require $x \neq y$ because we don't want to replace *bound* occurrences of the variable x in M by N. For example, if we compute the substitution $\lambda x.\, x[y/x]$, the result should not be $\lambda x.\, y$ but simply $\lambda x.\, x$. explanation

When substituting N for x in $\lambda y.\, P$, we also require that $y \notin \mathrm{FV}(N)$. For example, we cannot substitute y for x in $\lambda y.\, x$, i.e., $\lambda y.\, x[y/x]$, because it would result in $\lambda y.\, y$, a term that stands for the function that accepts an argument and returns it directly. But the term $\lambda y.\, x$ stands for a function that always returns the term x (or whatever x refers to). So the result we actually want is a function that accepts an argument, drop it, and returns the environment variable y. To do this properly, we would first have to "rename" the bound variable y.

Problem 42.6. What is the result of the following substitutions?

1. $\lambda y.\, x(\lambda w.\, vwx)[(uv)/x]$
2. $\lambda y.\, x(\lambda x.\, x)[(\lambda y.\, xy)/x]$
3. $y(\lambda v.\, xv)[(\lambda y.\, vy)/x]$

lam:syn:sub: thm:notinfv **Theorem 42.14.** *If $x \notin \mathrm{FV}(M)$, then $\mathrm{FV}(M[N/x]) = \mathrm{FV}(M)$, if the left-hand side is defined.*

Proof. By induction on the formation of M.

1. M is a variable: exercise.
2. M is of the form (PQ): exercise.
3. M is of the form $\lambda y.\, P$, and since $\lambda y.\, P[N/x]$ is defined, it has to be $\lambda y.\, P[N/x]$. Then $P[N/x]$ has to be defined; also, $x \neq y$ and $x \notin \mathrm{FV}(Q)$. Then:

$$\begin{aligned}
\mathrm{FV}(\lambda y.\, P[N/x]) = & \\
&= \mathrm{FV}(\lambda y.\, P[N/x]) && \text{by (4)} \\
&= \mathrm{FV}(P[N/x]) \setminus \{y\} && \text{by Definition 42.10(2)} \\
&= \mathrm{FV}(P) \setminus \{y\} && \text{by inductive hypothesis} \\
&= \mathrm{FV}(\lambda y.\, P) && \text{by Definition 42.10(2)}
\end{aligned}$$

□

Problem 42.7. Complete the proof of Theorem 42.14.

lam:syn:sub: thm:infv **Theorem 42.15.** *If $x \in \mathrm{FV}(M))$, then $\mathrm{FV}(M[N/x]) = (\mathrm{FV}(M) \setminus \{x\}) \cup \mathrm{FV}(N)$, provided the left hand is defined.*

Proof. By induction on the formation of M.

1. M is a variable: exercise.
2. M is of the form PQ: Since $(PQ)[N/y]$ is defined, it has to be $(P[N/x])(Q[N/x])$ with both substitution defined. Also, since $x \in \mathrm{FV}(PQ)$, either $x \in \mathrm{FV}(P)$ or $x \in \mathrm{FV}(Q)$ or both. The rest is left as an exercise.
3. M is of the form $\lambda y.\, P$. Since $\lambda y.\, P[N/x]$ is defined, it has to be $\lambda y.\, P[N/x]$, with $P[N/x]$ defined, $x \neq y$ and $y \notin \mathrm{FV}(N)$; also, since $y \in \mathrm{FV}(\lambda x.\, P)$, we have $y \in \mathrm{FV}(P)$ too. Now:

$$\begin{aligned}
&\mathrm{FV}((\lambda y.\, P)[N/x]) = \\
&\quad = \mathrm{FV}(\lambda y.\, P[N/x]) \\
&\quad = \mathrm{FV}(P[N/x]) \setminus \{y\} \\
&\quad = ((\mathrm{FV}(P) \setminus \{y\}) \cup (\mathrm{FV}(N) \setminus \{x\}) && \text{by inductive hypothesis} \\
&\quad = (\mathrm{FV}(P) \setminus \{x, y\}) \cup \mathrm{FV}(N) && x \notin \mathrm{FV}(N) \\
&\quad = (\mathrm{FV}(\lambda y.\, P) \setminus \{x\}) \cup \mathrm{FV}(N)
\end{aligned}$$

□

Problem 42.8. Complete the proof of Theorem 42.15.

Theorem 42.16. *$x \notin \mathrm{FV}(M[N/x])$, if the right-hand side is defined and $x \notin \mathrm{FV}(N)$.* *lam:syn:sub:thm:clr*

Proof. Exercise. □

Problem 42.9. Prove Theorem 42.16.

Theorem 42.17. *If $M[y/x]$ is defined and $y \notin \mathrm{FV}(M)$, then $M[y/x][x/y] = M$.* *lam:syn:sub:thm:inv*

Proof. By induction on the formation of M.

1. M is a variable z: Exercise.
2. M is of the form (PQ). Then:

$$\begin{aligned}
(PQ)[y/x][x/y] &= ((P[y/x])(Q[y/x]))[x/y] \\
&= (P[y/x][x/y])(Q[y/x][x/y]) \\
&= (PQ) \text{ by inductive hypothesis}
\end{aligned}$$

3. M is of the form $\lambda z.\, N$. Because $\lambda z.\, N[y/x]$ is defined, we know that $z \neq y$. So:

$$\begin{aligned}
&(\lambda z.\, N)[y/x][x/y] \\
&\qquad = (\lambda z.\, N[y/x])[x/y] \\
&\qquad = \lambda z.\, N[y/x][x/y] \\
&\qquad = \lambda z.\, N \text{ by inductive hypothesis}
\end{aligned}$$

□

Problem 42.10. Complete the proof of Theorem 42.17.

---

content/lambda-calculus/syntax/alpha.tex

42.6 α-Conversion

lam:syn:alp:sec

What is the relation between $\lambda x.\, x$ and $\lambda y.\, y$? They both represent the identity function. They are, of course, syntactically different terms. They differ only in the name of the bound variable, and one is the result of "renaming" the bound variable in the other. This is called *α-conversion*.

Definition 42.18 (Change of bound variable, $\xrightarrow{\alpha}$). If a term M contains an occurrence of $\lambda x.\, N$, $y \notin \mathrm{FV}(N)$, and $N[y/x]$ is defined, then replacing this occurrence by

$$\lambda y.\, N[y/x]$$

resulting in M' is called a *change of bound variable*, written as $M \xrightarrow{\alpha} M'$.

Definition 42.19 (Compatibility of relation). A relation R on terms is said to be *compatible* if it satisfies following conditions:

1. If RNN' then $R\lambda x.\, N\lambda x.\, N'$
2. If RPP' then $R(PQ)(P'Q)$
3. If RQQ' then $R(PQ)(PQ')$

Thus let's rephrase the definition:

Definition 42.20 (Change of bound variable, $\xrightarrow{\alpha}$). *Change of bound variable* ($\xrightarrow{\alpha}$) is the smallest compatible relation on terms satisfying following condition:

$$\lambda x.\, N \xrightarrow{\alpha} \lambda y.\, N[y/x] \qquad \text{if } x \neq y,\ y \notin \mathrm{FV}(N) \text{ and } N[y/x] \text{ is defined}$$

"Smallest" here means the relation contains only pairs that are required by compatibility and the additional condition, and nothing else. Thus this relation can also be defined as follows:

lam:syn:alp:defn:aconvone

Definition 42.21 (Change of bound variable, $\xrightarrow{\alpha}$). *Change of bound variable* ($\xrightarrow{\alpha}$) is inductively defined as follows:

lam:syn:alp:defn:aconvone1
1. If $N \xrightarrow{\alpha} N'$ then $\lambda x.\, N \xrightarrow{\alpha} \lambda x.\, N'$

lam:syn:alp:defn:aconvone2
2. If $P \xrightarrow{\alpha} P'$ then $(PQ) \xrightarrow{\alpha} (P'Q)$

lam:syn:alp:defn:aconvone3
3. If $Q \xrightarrow{\alpha} Q'$ then $(PQ) \xrightarrow{\alpha} (PQ')$

4. If $x \neq y$, $y \notin \mathrm{FV}(N)$ and $N[y/x]$ is defined, then $\lambda x.\, N \xrightarrow{\alpha} \lambda y.\, N[y/x]$. lam:syn:alp: defn:aconvone4

The definitions are equivalent, but we leave the proof as an exercise. From now on we will use the inductive definition.

Definition 42.22 (α-conversion, $\xrightarrow[\alpha]{}\!\!\!\!\!\rightarrow$). *$\alpha$-conversion* ($\xrightarrow[\alpha]{}\!\!\!\!\!\rightarrow$) is the smallest reflexitive and transitive relation on terms containing $\xrightarrow{\alpha}$.

As above, "smallest" means the relation only contains pairs required by transitivity, and $\xrightarrow{\alpha}$, which leads to the following equivalent definition:

Definition 42.23 (α-conversion, $\xrightarrow[\alpha]{}\!\!\!\!\!\rightarrow$). *$\alpha$-conversion* ($\xrightarrow[\alpha]{}\!\!\!\!\!\rightarrow$) is inductively defined as follows: lam:syn:alp: defn:aconv

1. If $P \xrightarrow[\alpha]{}\!\!\!\!\!\rightarrow Q$ and $Q \xrightarrow[\alpha]{}\!\!\!\!\!\rightarrow R$, then $P \xrightarrow[\alpha]{}\!\!\!\!\!\rightarrow R$. lam:syn:alp: defn:aconv1
2. If $P \xrightarrow{\alpha} Q$, then $P \xrightarrow[\alpha]{}\!\!\!\!\!\rightarrow Q$. lam:syn:alp: defn:aconv2
3. $P \xrightarrow[\alpha]{}\!\!\!\!\!\rightarrow P$. lam:syn:alp: defn:aconv3

Example 42.24. $\lambda x.\, fx$ α-converts to $\lambda y.\, fy$, and conversely. Informally speaking, they are both functions that accept an argument and return f of that argument, refering to the environment variable f.

$\lambda x.\, fx$ does not α-convert to $\lambda x.\, gx$. Informally speaking, they refer to the environment variables f and g respectively, and this makes them different functions: they behave differently in environments where f and g are different.

Problem 42.11. Are the following pairs of terms α-convertible?

1. $\lambda x.\, \lambda y.\, x$ and $\lambda y.\, \lambda x.\, y$
2. $\lambda x.\, \lambda y.\, x$ and $\lambda c.\, \lambda b.\, a$
3. $\lambda x.\, \lambda y.\, x$ and $\lambda c.\, \lambda b.\, a$

Lemma 42.25. *If $P \xrightarrow{\alpha} Q$ then $\mathrm{FV}(P) = \mathrm{FV}(Q)$.* lam:syn:alp: lem:fv-one

Proof. By induction on the derivation of $P \xrightarrow{\alpha} Q$.

1. If the last rule is (4), then P is of the form $\lambda x.\, N$ and Q of the form $\lambda y.\, N[y/x]$, with $x \neq y$, $y \notin \mathrm{FV}(N)$ and $N[y/x]$ defined. We distinguish cases according to whether $x \in \mathrm{FV}(N)$:

   a) If $x \in FV(N)$, then:

$$\begin{aligned}
\mathrm{FV}(\lambda y.\, N[y/x]) &= \mathrm{FV}(N[y/x]) \setminus \{y\} \\
&= ((\mathrm{FV}(N) \setminus \{x\}) \cup \{y\}) \setminus \{y\} \quad \text{by Theorem 42.15} \\
&= \mathrm{FV}(N) \setminus \{x\} \\
&= \mathrm{FV}(\lambda x.\, N)
\end{aligned}$$

b) If $x \notin FV(N)$, then:

$$\begin{aligned} \mathrm{FV}(\lambda y.\, N[y/x]) &= FVN[y/x] \setminus \{y\} \\ &= \mathrm{FV}(N) \setminus \{x\} \qquad \text{by Theorem 42.14} \\ &= \mathrm{FV}(\lambda x.\, N). \end{aligned}$$

2. The other three cases are left as exercises. □

Problem 42.12. Complete the proof of Lemma 42.25.

lam:syn:alp: lem:inv **Lemma 42.26.** *If $P \xrightarrow{\alpha} Q$ then $Q \xrightarrow{\alpha} P$.*

Proof. Induction on the derivation of $P \xrightarrow{\alpha} Q$.

1. If the last rule is (4), then P is of the form $\lambda x.\, N$ and Q of the form $\lambda y.\, N[y/x]$, where $x \neq y$, $y \notin \mathrm{FV}(N)$ and $N[y/x]$ defined. First, we have $y \notin \mathrm{FV}(N[y/x])$ by Theorem 42.16. By Theorem 42.17 we have that $N[y/x][x/y]$ is not only defined, but also equal to N. Then by (4), we have $\lambda y.\, N[y/x] \xrightarrow{\alpha} \lambda x.\, N[y/x][x/y] = \lambda x.\, N$. □

Problem 42.13. Complete the proof of Lemma 42.26

Theorem 42.27. *α-Conversion is an equivalence relation on terms, i.e., it is reflexive, symmetric, and transitive.*

Proof.
1. For each term M, M can be changed to M by *zero* changes of bound variables.
2. If P is α-converts to Q by a series of changes of bound variables, then from Q we can just inverse these changes (by Lemma 42.26) in opposite order to obtain P.
3. If P α-converts to Q by a series of changes of bound variables, and Q to R by another series, then we can change P to R by first applying the first series and then the second series. □

From now on we say that M and N are *α-equivalent*, $M \overset{\alpha}{=} N$, iff M α-converts to N (which, as we've just shown, is the case iff N α-converts to M).

lam:syn:alp: thm:fv **Theorem 42.28.** *If $M \overset{\alpha}{=} N$, then $\mathrm{FV}(M) = \mathrm{FV}(N)$.*

Proof. Immediate from Lemma 42.25. □

lam:syn:alp: lem:sub:R **Lemma 42.29.** *If $R \overset{\alpha}{=} R'$ and $M[R/y]$ is defined, then $M[R'/y]$ is defined and α-equivalent to $M[R/y]$.*

Proof. Exercise. □

Problem 42.14. Prove Lemma 42.29.

Recall that in section 42.5, substitution is undefined in some cases; however, using α-conversion on terms, we can make substitution always defined by renaming bound variables. The result preserves α-equivalence, as shown in this theorem:

Theorem 42.30. *For any M, R, and y, there exists M' such that $M \stackrel{\alpha}{=} M'$ and $M'[R/y]$ is defined. Moreover, if there is another pair $M'' \stackrel{\alpha}{=} M$ and R'' where $M''[R''/y]$ is defined and $R'' \stackrel{\alpha}{=} R$, then $M'[R/y] \stackrel{\alpha}{=} M''[R''/y]$.* *lam:syn:alp: thm:sub*

Proof. By induction on the formation of M:

1. M is a variable z: Exercise.

2. Suppose M is of the form $\lambda x.\, N$. Select a variable z other than x and y and such that $z \notin \mathrm{FV}(N)$ and $z \notin \mathrm{FV}(R)$. By inductive hypothesis, we there is N' such that $N' \stackrel{\alpha}{=} N$ and $N'[z/x]$ is defined. Then $\lambda x.\, N \stackrel{\alpha}{=} \lambda x.\, N'$ too, by Definition 42.21(1). Now $\lambda x.\, N' \stackrel{\alpha}{=} \lambda z.\, N'[z/x]$ by Definition 42.21(4). We can do this because $z \neq x$, $z \notin FV(N')$ and $N'[z/x]$ is defined. Finally, $\lambda z.\, N'[z/x][R/y]$ is defined, because $z \neq y$ and $z \notin FV(R)$.

   Moreover, if there is another N'' and R'' satisfying the same conditions,

   $$\begin{aligned}(\lambda z.\, N''[z/x])[R''/y] = & \\ &= \lambda z.\, N''[z/x][R''/y] \\ &= \lambda z.\, N''[z/x][R/y] && \text{by Lemma 42.29} \\ &= \lambda z.\, N'[z/x][R/y] && \text{by inductive hypothesis} \\ &= (\lambda z.\, N'[z/x])[R/y]\end{aligned}$$

3. M is of the form (PQ): Exercise. □

Problem 42.15. Complete the proof of Theorem 42.30.

Corollary 42.31. *For any M, R, and y, there exists a pair of M' and R' such that $M' \stackrel{\alpha}{=} M$, $R \stackrel{\alpha}{=} R'$ and $M'[R'/y]$ is defined. Moreover, if there is another pair $M'' \stackrel{\alpha}{=} M$ and R'' with $M'[R'/y]$ defined, then $M'[R'/y] \stackrel{\alpha}{=} M''[R''/y]$.* *lam:syn:alp: cor:sub*

Proof. Immediate from Theorem 42.30. □

---

content/lambda-calculus/syntax/de-bruijn.tex

42.7 The De Bruijn Index

lam:syn:deb:sec

α-Equivalence is very natural, as terms that are α-equivalent "mean the same." In fact, it is possible to give a syntax for lambda terms which does not distinguish terms that can be α-converted to each other. The best known replaces variables by their *De Bruijn index*.

When we write $\lambda x.\, M$, we explicitly state that x is the parameter of the function, so that we can use x in M to refer to this parameter. In the de Bruijn index, however, parameters have no name and reference to them in the function body is denoted by a number denoting the levels of abstraction between them. For example, consider the example of $\lambda x.\, \lambda y.\, yx$: the outer abstraction is on binds the variable x; the inner abstraction binds the variable is y; the sub-term yx lies in the scope of the inner abstraction: there is no abstraction between y and its abstract λy, but one abstract between x and its abstract λx. Thus we write $0\,1$ for yx, and $\lambda.\, \lambda.\, 01$ for the entire term.

Definition 42.32. De Bruijn terms are inductively defines as follows:

1. n, where n is any natural number.
2. PQ, where P and Q are both De Bruijn terms.
3. $\lambda.\, N$, where N is a De Bruijn term.

A formalized translation from ordinary lambda terms to De Bruijn indexed terms is as follows:

Definition 42.33.

$$\begin{aligned} F_\Gamma(x) &= \Gamma(x) \\ F_\Gamma(PQ) &= F_\Gamma(P) F_\Gamma(Q) \\ F_\Gamma(\lambda x.\, N) &= \lambda.\, F_{x,\Gamma}(N) \end{aligned}$$

where Γ is a list of variables indexed from zero, and $\Gamma(x)$ denotes the position of the variable x in Γ. For example, if Γ is x, y, z, then $\Gamma(x)$ is 0 and $\Gamma(z)$ is 2.

x, Γ denotes the list resulted from pushing x to the head of Γ; for instance, continuing the Γ in last example, w, Γ is w, x, y, z.

Recovering a standard lambda term from a de Bruijn term is done as follows:

Definition 42.34.

$$\begin{aligned} G_\Gamma(n) &= \Gamma[n] \\ G_\Gamma(PQ) &= G_\Gamma(P) G_\Gamma(Q) \\ G_\Gamma(\lambda.\, N) &= \lambda x.\, G_{x,\Gamma}(N) \end{aligned}$$

where Γ is again a list of variables indexed from zero, and $\Gamma[n]$ denotes the variable in position n. For example, if Γ is x, y, z, then $\Gamma[1]$ is y.

The variable x in last equation is chosen to be any variable that not in Γ.

Here we give some results without proving them:

Proposition 42.35. *If $M \xrightarrow{\alpha} M'$, and Γ is any list containing* $\mathrm{FV}(M)$*, then* $F_\Gamma(M) \equiv F_\Gamma(M')$.

---

content/lambda-calculus/syntax/term-revisited.tex

42.8 Terms as α-Equivalence Classes

lam:syn:tr:sec

From now on, we will consider terms up to α-equivalence. That means when we write a term, we mean its α-equivalance class it is in. For example, we write $\lambda a.\, \lambda b.\, ac$ for the set of all terms α-equivalent to it, such as $\lambda a.\, \lambda b.\, ac$, $\lambda b.\, \lambda a.\, bc$, etc.

Also, while in previous sections letters such as N, Q are used to denote a term, from now on we use them to denote a class, and it is these classes instead of terms that will be our subjects of study in what follows. Letters such as x, y continues to denote a variable.

We also adopt the notation $\underline{M}$ to denote an arbitrary element of the class M, and $\underline{M}_0, \underline{M}_1$, *etc.* if we need more than one.

We reuse the notations from terms to simplify our wording. We have following definition on classes:

Definition 42.36.

1. $\lambda x.\, N$ is defined as the class containing $\lambda x.\, \underline{N}$.
2. PQ is defined to be the class containing $\underline{P}\,\underline{Q}$.

It is not hard to see that they are well defined, because α-conversion is compatible.

Definition 42.37. The *free variables* of an α-equivalence class M, or $FV(M)$, is defined to be $FV(\underline{M})$.

lam:syn:tr:def:fv

This is well defined since $FV(\underline{M}_0) = FV(\underline{M}_1)$, as shown in Theorem 42.28.

We also reuse the notation for substition into classes:

Definition 42.38. The *substitution* of R for y in M, or $M[R/y]$, is defined to be $\underline{M}_{\underline{R}/y}$, for any $\underline{M}$ and $\underline{R}$ making the substition defined.

lam:syn:tr:def:sub

This is also well defined as shown in Corollary 42.31.

Note how this definition significantly simplifies our reasoning. For example:

$$\begin{aligned} \lambda x.\, x[y/x] &= && (42.1) \\ &= \lambda z.\, z[y/x] && (42.2) \\ &= \lambda z.\, z[y/x] && (42.3) \\ &= \lambda z.\, z && (42.4) \end{aligned}$$

lam:syn:tr:eq:1

lam:syn:tr:eq:2

eq. (42.1) is undefined if we still regard it as substitution on terms; but as mentioned earlier, we now consider it a substitution on classes, which is why eq. (42.2) can happen: we can replace $\lambda x.\, x$ with $\lambda z.\, z$ because they belong to the same class.

For the same reason, from now on we will assume that the representatives we choose always satisfy the conditions needed for substitution. For example, when we see $\lambda x.\, N[R/y]$, we will assume the representative $\lambda x.\, N$ is chosen so that $x \neq y$ and $x \notin FV(R)$.

Since it is a bit strange to call $\lambda x.\, x$ a "class", let's call them Λ-terms (or simply "terms" in the rest of the part) from now on, to distinguish them from λ-terms that we are familiar with.

> We cannot say goodbye to terms yet: the whole definition of Λ-terms is based on λ-terms, and we haven't provided a method to define functions on Λ-terms, which means all such functions have to be first defined on λ-terms, and then "projected" to Λ-terms, as we did for substitutions. However we assume the reader can intuitively understand how we can define functions on Λ-terms.

---

content/lambda-calculus/syntax/beta.tex

42.9 β-reduction

lam:int:bet:sec

When we see $(\lambda m.\, (\lambda y.\, y)m)$, it is natural to conjecture that it has some connection with $\lambda m.\, m$, namely the second term should be the result of "simplifying" the first. The notion of *β-reduction* captures this intuition formally.

lam:int:bet:defn:betacontr

Definition 42.39 (β-contraction, $\xrightarrow{\beta}$). *The β-contraction* ($\xrightarrow{\beta}$) is the smallest compatible relation on terms satisfying the following condition:

$$(\lambda x.\, N)Q \xrightarrow{\beta} N[Q/x]$$

We say P is *β-contracted* to Q if $P \xrightarrow{\beta} Q$. A term of the form $(\lambda x.\, N)Q$ is called a *redex*.

lam:int:bet:prob:def

Problem 42.16. Spell out the equivalent inductive definitions of β-contraction as we did for change of bound variable in Definition 42.21.

lam:int:bet:defn:betared

Definition 42.40 (β-reduction, $\xrightarrow{\beta}\!\!\!\!\rightarrow$). *β-reduction* ($\xrightarrow{\beta}\!\!\!\!\rightarrow$) is the smallest reflexive, transitive relation on terms containing $\xrightarrow{\beta}$. We say P is β-reduced to Q if $P \xrightarrow{\beta}\!\!\!\!\rightarrow Q$.

We will write $\to$ instead of $\xrightarrow{\beta}$, and $\twoheadrightarrow$ instead of $\xrightarrow{\beta}\!\!\!\!\!\!\twoheadrightarrow$ when context is clear.

Informally speaking, $M \xrightarrow{\beta}\!\!\!\!\!\!\twoheadrightarrow N$ if and only if M can be changed to N by zero or several steps of β-contraction.

Definition 42.41 (β-normal). A term that cannot be β-contracted any further is said to be *β-normal*.

If $M \xrightarrow{\beta}\!\!\!\!\!\!\twoheadrightarrow N$ and N is β-normal, then we say N is a *normal form* of M. One may ask if the normal form of a term is unique, and the answer is yes, as we will see later.

Let us consider some examples.

1. We have
$$\begin{aligned}(\lambda x.\, xxy)\lambda z.\, z &\to (\lambda z.\, z)(\lambda z.\, z)y \\ &\to (\lambda z.\, z)y \\ &\to y\end{aligned}$$

2. "Simplifying" a term can actually make it more complex:
$$\begin{aligned}(\lambda x.\, xxy)(\lambda x.\, xxy) &\to (\lambda x.\, xxy)(\lambda x.\, xxy)y \\ &\to (\lambda x.\, xxy)(\lambda x.\, xxy)yy \\ &\to \dots\end{aligned}$$

3. It can also leave a term unchanged:
$$(\lambda x.\, xx)(\lambda x.\, xx) \to (\lambda x.\, xx)(\lambda x.\, xx)$$

4. Also, some terms can be reduced in more than one way; for example,
$$(\lambda x.\, (\lambda y.\, yx)z)v \to (\lambda y.\, yv)z$$
by contracting the outermost application; and
$$(\lambda x.\, (\lambda y.\, yx)z)v \to (\lambda x.\, zx)v$$
by contracting the innermost one. Note, in this case, however, that both terms further reduce to the same term, zv.

The final outcome in the last example is not a coincidence, but rather illustrates a deep and important property of the lambda calculus, known as the Church–Rosser property.

digression

In general, there is more than one way to β-reduce a term, thus many reduction strategies have been invented, among which the most common is the *natural strategy*. The natural strategy always contracts the *left-most* redex, where the position of a redex is defined as its starting point in the term. The natural strategy has the useful property that a term can be reduced to a normal form by some strategy iff it can be reduced to normal form using the natural strategy. In what follows we will use the natural stratuegy unless otherwise specified.

Definition 42.42 (β-equivalence, $=$). *β-Equivalence* $(=)$ is the relation inductively defined as follows:

1. $M = M$.
2. If $M = N$, then $N = M$.
3. If $M = N$, $N = O$, then $M = O$.
4. If $M = N$, then $PM = PN$.
5. If $M = N$, then $MQ = NQ$.
6. If $M = N$, then $\lambda x.\, M = \lambda x.\, N$.
7. $(\lambda x.\, N)Q = N[Q/x]$.

The first three rules make the relation an equivalence relation; the next three make it compatible; the last ensures that it contains β-contraction.

Informally speaking, two terms are β-equivalent if and only if one of them can be changed to the other in zero or more steps of β-contraction, or "inverse" of β-contraction. The inverse of β-contraction is defined so that M inverse-β-contracts to N iff N β-contracts to M.

Besides the above rules, we will extend the relation with more rules, and denote the extended equivalence relation as $\overset{X}{=}$, where X is the extending rule.

---

content/lambda-calculus/syntax/eta.tex

42.10 η-conversion

lam:syn:eta:sec

There is another relation on λ terms. In section 42.4 we used the example $\lambda x.\, (fx)$, which accepts an argument and applies f to it. In other words, it is the same function as f: $\lambda x.\, (fx)N$ and fN both reduce to fN. We use η-reduction (and η-extension) to capture this idea.

lam:syn:eta:defn:beredone

Definition 42.43 (η-contraction, $\xrightarrow{\eta}$). *η-contraction* $(\xrightarrow{\eta})$ is the smallest compatible relation on terms satisfying the following condition:

$$\lambda x.\, Mx \xrightarrow{\eta} M \text{ provided } x \notin FV(M)$$

lam:syn:eta:defn:bered

Definition 42.44 ($\beta\eta$-reduction, $\xrightarrow{\beta\eta}\!\!\!\!\rightarrow$). *$\beta\eta$-reduction* $(\xrightarrow{\beta\eta}\!\!\!\!\rightarrow)$ is the smallest reflexive, transitive relation on terms containing $\xrightarrow{\beta}$ and $\xrightarrow{\eta}$, i.e., the rules of reflexivity and transitive plus the following two rules:

lam:syn:eta:defn:bered3

1. If $M \xrightarrow{\beta} N$ then $M \xrightarrow{\beta\eta}\!\!\!\!\rightarrow N$.

lam:syn:eta:defn:bered4

2. If $M \xrightarrow{\eta} N$ then $M \xrightarrow{\beta\eta}\!\!\!\!\rightarrow N$.

Definition 42.45. We extend the equivalence relation $=$ with the η-conversion rule:

$$\lambda x.\, fx = f$$

and denote the extended relation as $\overset{\eta}{=}$.

η-equivalence is important because it is related to extensionality of lambda terms:

Definition 42.46 (Extensionality). We extend the equivalence relation $=$ with the (ext) rule:

$$\text{If } Mx = Nx \text{ then } M = N, \text{ provided } x \notin FV(MN).$$

and denote the extended relation as $\overset{ext}{=}$.

Roughly speaking, the rule states that two terms, viewed as functions, should be considered equal if they behave the same for the same argument.

We now prove that the η rule provides exactly the extensionality, and nothing else.

Theorem 42.47. *$M \overset{ext}{=} N$ if and only if $M \overset{\eta}{=} N$.*

Proof. First we prove that $\overset{\eta}{=}$ is closed under the extensionality rule. That is, *ext* rule doesn't add anything to $\overset{\eta}{=}$. We then have $\overset{\eta}{=}$ contains $\overset{ext}{=}$, and if $M \overset{ext}{=} N$, then $M \overset{\eta}{=} N$.

To prove $\overset{\eta}{=}$ is closed under *ext*, note that for any $M = N$ derived by the *ext* rule, we have $Mx \overset{\eta}{=} Nx$ as premise. Then we have $\lambda x.\, Mx \overset{\eta}{=} \lambda x.\, Nx$ by a rule of $=$, applying η on both side gives us $M \overset{\eta}{=} N$.

Similarly we prove that the η rule is contained in $\overset{ext}{=}$. For any $\lambda x.\, Mx$ and M with $x \notin FV(M)$, we have that $(\lambda x.\, Mx)x \overset{ext}{=} Mx$, giving us $\lambda x.\, Mx \overset{ext}{=} M$ by the *ext* rule. □

Chapter 43

The Church–Rosser Property

---

content/lambda-calculus/church-rosser/definitions-and-properties.tex

43.1 Definition and Properties

lam:cr:dap:sec In this chapter we introduce the concept of Church–Rosser property and some common properties of this property.

Definition 43.1 (Church–Rosser property, CR). A relation $\xrightarrow{X}$ on terms is said to satisfy the *Church–Rosser property* iff, whenever $M \xrightarrow{X} P$ and $M \xrightarrow{X} Q$, then there exists some N such that $P \xrightarrow{X} N$ and $Q \xrightarrow{X} N$.

We can view the lambda calculus as a model of computation in which terms in normal form are "values" and a reducibility relation on terms are the "calculation rules." The Church–Rosser property states is that when there is more than one way to proceed with a calculation, there is still only a single value of the expression.

To take an example from elementary algebra, there's more than one way to calculate $4 \times (1 + 2) + 3$. It can either be reduced to $4 \times 3 + 3$ (if we first reduce $1 + 2$ to 3) or to $4 \times 1 + 4 \times 2 + 3$ (if we first reduce $4 \times (1 + 2)$ using distributivity). Both of these, however, can be further reduced to $12 + 3$.

If we take $\xrightarrow{X}$ to be β-reduction, we easily see that a consequence of the Church–Rosser property is that if a term has a normal form, then it is unique. For suppose M can be reduced to P and Q, both of which are normal forms. By the Church–Rosser property, there exists some N such that both P and Q reduce to it. Since by assumption P and Q are normal forms, the reduction of P and Q to N can only be the trivial reduction, i.e., P, Q, and N are identical. This justifies our speaking of *the* normal form of a term.

In viewing the lambda calculus as a model of computation, then, the normal form of a term can be thought of as the "final result" of the computation starting with that term. The above corollary means there's only one, if any, final result of a computation, just like there is only one result of computing $4 \times (1 + 2) + 3$, namely 15.

lam:cr:dap:thm:str **Theorem 43.2.** *If a relation $\xrightarrow{X}$ satisfies the Church–Rosser property, and $\xrightarrow{X}\!\!\!\!\!\rightarrow$ is the smallest transitive relation containing $\xrightarrow{X}$, then $\xrightarrow{X}\!\!\!\!\!\rightarrow$ satisfies the Church–Rosser property too.*

Proof. Suppose

$$M \xrightarrow{X} P_1 \xrightarrow{X} \dots \xrightarrow{X} P_m \text{ and}$$
$$M \xrightarrow{X} Q_1 \xrightarrow{X} \dots \xrightarrow{X} Q_n.$$

We will prove the theorem by constructing a grid N of terms of height is $m+1$ and width $n + 1$. We use $N_{i,j}$ to denote the term in the i-th row and j-th column.

We construct N in such a way that $N_{i,j} \xrightarrow{X} N_{i+1,j}$ and $N_{i,j} \xrightarrow{X} N_{i,j+1}$. It is defined as follows:

$$\begin{aligned} N_{0,0} &= M \\ N_{i,0} &= P_i && \text{if } 1 \leq i \leq m \\ N_{0,j} &= Q_j && \text{if } 1 \leq j \leq n \end{aligned}$$

and otherwise:

$$N_{i,j} = R$$

where R is a term such that $N_{i-1,j} \xrightarrow{X} R$ and $N_{i,j-1} \xrightarrow{X} R$. By the Church–Rosser property of $\xrightarrow{X}$, such a term always exists.

Now we have $N_{m,0} \xrightarrow{X} \ldots \xrightarrow{X} N_{m,n}$ and $N_{0,n} \xrightarrow{X} \ldots \xrightarrow{X} N_{m,n}$. Note $N_{m,0}$ is P and $N_{0,n}$ is Q. By definition of $\xrightarrow{X}\!\!\!\!\twoheadrightarrow$ the theorem follows. □

---

content/lambda-calculus/church-rosser/parallel-beta-reduction.tex

43.2 Parallel β-reduction

We introduce the notion of *parallel β-reduction*, and prove the it has the Church–Rosser property. lam:cr:pb:sec

Definition 43.3 (parallel β-reduction, $\overset{\beta}{\Longrightarrow}$). Parallel reduction ($\overset{\beta}{\Longrightarrow}$) of terms is inductively defined as follows: lam:cr:pb:defn:bredpar

1. $x \overset{\beta}{\Longrightarrow} x$. lam:cr:pb:defn:bredpar1
2. If $N \xrightarrow{\beta} N'$ then $\lambda x.\, N \overset{\beta}{\Longrightarrow} \lambda x.\, N'$. lam:cr:pb:defn:bredpar2
3. If $P \overset{\beta}{\Longrightarrow} P'$ and $Q \overset{\beta}{\Longrightarrow} Q'$ then $PQ \overset{\beta}{\Longrightarrow} P'Q'$. lam:cr:pb:defn:bredpar3
4. If $N \overset{\beta}{\Longrightarrow} N'$ and $Q \overset{\beta}{\Longrightarrow} Q'$ then $(\lambda x.\, N)Q \overset{\beta}{\Longrightarrow} N'[Q'/x]$. lam:cr:pb:defn:bredpar4

Parallel β-reduction allows us to reduce any number of redices in a term in one step. It is different from β-reduction in the sense that we can only contract redices that occur in the original term, but not redices arising from parallel β-reduction. For example, the term $(\lambda f.\, fx)(\lambda y.\, y)$ can only be parallel β-reduced to itself or to $(\lambda y.\, y)x$, but not further to x, although it β-reduces to x, because this redex arises only after one step of parallel β-reduction. A second parallel β-reduction step yields x, though.

Theorem 43.4. $M \overset{\beta}{\Longrightarrow} M$. *lam:cr:pb:thm:refl*

Proof. Exercise. □

Problem 43.1. Prove Theorem 43.4.

lam:cr:pb: defn:bcd

Definition 43.5 (β-complete development). The *β-complete development* $M^{*\beta}$ of M is defined inductively as follows:

lam:cr:pb: defn:bcd1
$$x^{*\beta} = x \tag{43.1}$$

lam:cr:pb: defn:bcd2
$$(\lambda x.\, N)^{*\beta} = \lambda x.\, N^{*\beta} \tag{43.2}$$

lam:cr:pb: defn:bcd3
$$(PQ)^{*\beta} = P^{*\beta}Q^{*\beta} \qquad \text{if } P \text{ is not a } \lambda\text{-abstract} \tag{43.3}$$

lam:cr:pb: defn:bcd4
$$((\lambda x.\, N)Q)^{*\beta} = N^{*\beta}[Q^{*\beta}/x] \tag{43.4}$$

The β-complete development of a term, as its name suggests, is a "complete parallel reduction." While for parallel β-reduction we still can choose to not contract a redex, for complete development we have no choice but to contract all of them. Thus the complete development of $(\lambda f.\, fx)(\lambda y.\, y)$ is $(\lambda y.\, y)x$, not itself.

> This definition has the problem that we haven't introduced how to define functions on (λ-)terms recursively. Will fix in future.

lam:cr:pb: lem:comp

Lemma 43.6. *If $M \overset{\beta}{\Longrightarrow} M'$ and $R \overset{\beta}{\Longrightarrow} R'$, then $M[R/y] \overset{\beta}{\Longrightarrow} M'[R'/y]$.*

Proof. By induction on the derivation of $M \overset{\beta}{\Longrightarrow} M'$.

1. The last step is (1): Exercise.
2. The last step is (2): Then M is $\lambda x.\, N$ and M' is $\lambda x.\, N'$, where $N \overset{\beta}{\Longrightarrow} N'$. We want to prove that $(\lambda x.\, N)[R/y] \overset{\beta}{\Longrightarrow} (\lambda x.\, N')[R'/y]$, i.e., $\lambda x.\, N[R/y] \overset{\beta}{\Longrightarrow} \lambda x.\, N'[R/y]$. This follows immediately by (2) and the induction hypothesis.
3. The last step is (3): Exercise.
4. The last step is (4): M is $(\lambda x.\, N)Q$ and M' is $N'[Q'/x]$. We want to prove that $((\lambda x.\, N)Q)[R/y] \overset{\beta}{\Longrightarrow} N'[Q'/x][R'/y]$, i.e., $(\lambda x.\, N[R/y])Q[R/y] \overset{\beta}{\Longrightarrow} N'[R'/y][Q'[R'/y]/x]$. This follows by (4) and the induction hypothesis. □

Problem 43.2. Complete the proof of Lemma 43.6.

lam:cr:pb: lem:cont

Lemma 43.7. *If $M \overset{\beta}{\Longrightarrow} M'$ then $M' \overset{\beta}{\Longrightarrow} M^{*\beta}$.*

Proof. By induction on the derivation of $M \overset{\beta}{\Longrightarrow} M'$.

1. The last rule is (1): Exercise.

2. The last rule is (2): M is $\lambda x.\, N$ and M' is $\lambda x.\, N'$ with $N \overset{\beta}{\Longrightarrow} N'$. We want to show that $\lambda x.\, N' \overset{\beta}{\Longrightarrow} (\lambda x.\, N)^{*\beta}$, i.e., $\lambda x.\, N' \overset{\beta}{\Longrightarrow} \lambda x.\, N^{*\beta}$ by eq. (43.2). It follows by (2) and the induction hypothesis.

3. The last rule is (3):M is PQ and M' is $P'Q'$ for some P, Q, P' and Q', with $P \overset{\beta}{\Longrightarrow} P'$ and $Q \overset{\beta}{\Longrightarrow} Q'$. By induction hypothesis, we have $P' \overset{\beta}{\Longrightarrow} P^{*\beta}$ and $Q' \overset{\beta}{\Longrightarrow} Q^{*\beta}$.

   a) If P is $\lambda x.\, N$ for some x and N, then P' must be $\lambda x.\, N'$ for some N' with $N \overset{\beta}{\Longrightarrow} N'$. By induction hypothesis we have $N' \overset{\beta}{\Longrightarrow} N^{*\beta}$ and $Q' \overset{\beta}{\Longrightarrow} Q^{*\beta}$. Then $(\lambda x.\, N')Q' \overset{\beta}{\Longrightarrow} N^{*\beta}[Q^{*\beta}/x]$ by (4).

   b) If P is not a λ-abstract, then $P'Q' \overset{\beta}{\Longrightarrow} P^{*\beta}Q^{*\beta}$ by (3), and the right-hand side is $PQ^{*\beta}$ by eq. (43.3).

4. The last rule is (4): M is $(\lambda x.\, N)Q$ and M' is $N'[Q'/x]$ for some x, N, Q, N', and Q', with $N \overset{\beta}{\Longrightarrow} N'$ and $Q \overset{\beta}{\Longrightarrow} Q'$. By induction hypothesis we know $N' \overset{\beta}{\Longrightarrow} N^{*\beta}$ and $Q' \overset{\beta}{\Longrightarrow} Q^{*\beta}$. By Lemma 43.6 we have $N'[Q'/x] \overset{\beta}{\Longrightarrow} N^{*\beta}[Q^{*\beta}/x]$, the right-hand side of which is exactly $((\lambda x.\, N)Q)^{*\beta}$. □

Problem 43.3. Complete the proof of Lemma 43.7.

Theorem 43.8. *$\overset{\beta}{\Longrightarrow}$ has the Church–Rosser property.*

lam:cr:pb: thm:cr

Proof. Immediate from Lemma 43.7. □

---

content/lambda-calculus/church-rosser/beta-reduction.tex

43.3 β-reduction

Lemma 43.9. *If $M \overset{\beta}{\to} M'$, then $M \overset{\beta}{\Longrightarrow} M'$.*

lam:cr:b: sec lam:cr:b: lem:one-par

Proof. If $M \overset{\beta}{\to} M'$, then M is $(\lambda x.\, N)Q$, M' is $N[Q/x]$, for some x, N, and Q. Since $N \overset{\beta}{\Longrightarrow} N$ and $Q \overset{\beta}{\Longrightarrow} Q$ by Theorem 43.4, we immediately have $(\lambda x.\, N)Q \overset{\beta}{\Longrightarrow} N[Q/x]$ by Definition 43.3(4). □

Lemma 43.10. *If $M \overset{\beta}{\Longrightarrow} M'$, then $M \overset{\beta}{\twoheadrightarrow} M'$.*

lam:cr:b: lem:par-red

Proof. By induction on the derivation of $M \overset{\beta}{\Longrightarrow} M'$.

1. The last rule is (1): Then M and M' are just x, and $x \xrightarrow{\beta}\!\!\!\!\rightarrow x$.
2. The last rule is (2): M is $\lambda x.\, N$ and M' is $\lambda x.\, N'$ for some x, N, N', where $N \stackrel{\beta}{\Longrightarrow} N'$. By induction hypothesis we have $N \xrightarrow{\beta}\!\!\!\!\rightarrow N'$. Then $\lambda x.\, N \xrightarrow{\beta}\!\!\!\!\rightarrow \lambda x.\, N'$ (by the same series of $\xrightarrow{\beta}$ contractions as $N \xrightarrow{\beta}\!\!\!\!\rightarrow N'$).
3. The last rule is (3): M is PQ and M' is $P'Q'$ for some P, Q, P', Q', where $P \stackrel{\beta}{\Longrightarrow} P'$ and $Q \stackrel{\beta}{\Longrightarrow} Q'$. By induction hypothesis we have $P \xrightarrow{\beta}\!\!\!\!\rightarrow P'$ and $Q \xrightarrow{\beta}\!\!\!\!\rightarrow Q'$. So $PQ \xrightarrow{\beta}\!\!\!\!\rightarrow P'Q'$ by the reduction sequence $P \xrightarrow{\beta}\!\!\!\!\rightarrow P'$ followed by the reduction $Q \xrightarrow{\beta}\!\!\!\!\rightarrow Q'$.
4. The last rule is (4): M is $(\lambda x.\, N)Q$ and M' is $N'[Q'/x]$ for some x, N, M', Q, Q', where $N \stackrel{\beta}{\Longrightarrow} N'$ and $Q \stackrel{\beta}{\Longrightarrow} Q'$. By induction hypothesis we get $Q \xrightarrow{\beta}\!\!\!\!\rightarrow Q'$ and $N \xrightarrow{\beta}\!\!\!\!\rightarrow N'$. So $(\lambda x.\, N)Q \xrightarrow{\beta}\!\!\!\!\rightarrow N'[Q'/x]$ by $N \xrightarrow{\beta}\!\!\!\!\rightarrow N'$ followed by $Q \xrightarrow{\beta}\!\!\!\!\rightarrow Q'$ and finally contraction of $(\lambda x.\, N')Q'$ to $N'[Q'/x]$. □

lam:cr:b:lem:str **Lemma 43.11.** *$\xrightarrow{\beta}\!\!\!\!\rightarrow$ is the smallest transitive relation containing $\stackrel{\beta}{\Longrightarrow}$.*

Proof. Let $\xrightarrow{X}\!\!\!\!\rightarrow$ be the smallest transitive relation containing $\stackrel{\beta}{\Longrightarrow}$.

$\xrightarrow{\beta}\!\!\!\!\rightarrow \subseteq \xrightarrow{X}\!\!\!\!\rightarrow$: Suppose $M \xrightarrow{\beta}\!\!\!\!\rightarrow M'$, i.e., $M \equiv M_1 \xrightarrow{\beta} \ldots \xrightarrow{\beta} M_k \equiv M'$. By Lemma 43.9, $M \equiv M_1 \stackrel{\beta}{\Longrightarrow} \ldots \stackrel{\beta}{\Longrightarrow} M_k \equiv M'$. Since is $\xrightarrow{X}\!\!\!\!\rightarrow$ contains $\stackrel{\beta}{\Longrightarrow}$ and is transitive, $M \xrightarrow{X}\!\!\!\!\rightarrow M'$.

$\xrightarrow{X}\!\!\!\!\rightarrow \subseteq \xrightarrow{\beta}\!\!\!\!\rightarrow$: Suppose $M \xrightarrow{X}\!\!\!\!\rightarrow M'$, i.e., $M \equiv M_1 \stackrel{\beta}{\Longrightarrow} \ldots \stackrel{\beta}{\Longrightarrow} M_k \equiv M'$. By Lemma 43.10, $M \equiv M_1 \xrightarrow{\beta}\!\!\!\!\rightarrow \ldots \xrightarrow{\beta}\!\!\!\!\rightarrow M_k \equiv M'$. Since $\xrightarrow{\beta}\!\!\!\!\rightarrow$ is transitive, $M \xrightarrow{\beta}\!\!\!\!\rightarrow M'$. □

lam:cr:b:thm:cr **Theorem 43.12.** *$\xrightarrow{\beta}\!\!\!\!\rightarrow$ satisfies the Church–Rosser property.*

Proof. Immediate from Theorem 43.2, Theorem 43.8, and Lemma 43.11. □

---

content/lambda-calculus/church-rosser/parallel-beta-eta-reduction.tex

43.4 Parallel $\beta\eta$-reduction

lam:cr:pbe:sec In this section we prove the Church-Rosser property for parallel $\beta\eta$-reduction, the parallel reduction notion corresponding to $\beta\eta$-reduction.

lam:cr:pbe:defn:beredpar **Definition 43.13 (Parallel $\beta\eta$-reduction, $\stackrel{\beta\eta}{\Longrightarrow}$).** *Parallel $\beta\eta$-reduction* ($\stackrel{\beta\eta}{\Longrightarrow}$) on terms is inductively defined as follows:

lam:cr:pbe:defn:beredpar1 1. $x \stackrel{\beta\eta}{\Longrightarrow} x$.

2. If $N \xrightarrow{\beta} N'$ then $\lambda x.\, N \xRightarrow{\beta\eta} \lambda x.\, N'$. lam:cr:pbe: defn:beredpar2
3. If $P \xRightarrow{\beta\eta} P'$ and $Q \xRightarrow{\beta\eta} Q'$ then $PQ \xRightarrow{\beta\eta} P'Q'$. lam:cr:pbe: defn:beredpar3
4. If $N \xRightarrow{\beta\eta} N'$ and $Q \xRightarrow{\beta\eta} Q'$ then $(\lambda x.\, N)Q \xRightarrow{\beta\eta} N'[Q'/x]$. lam:cr:pbe: defn:beredpar4
5. If $N \xRightarrow{\beta\eta} N'$ then $\lambda x.\, Nx \xRightarrow{\beta\eta} N'$, provided $x \notin FV(N)$. lam:cr:pbe: defn:beredpar5

Theorem 43.14. $M \xRightarrow{\beta\eta} M$. *lam:cr:pbe: thm:refl*

Proof. Exercise. □

Problem 43.4. Prove Theorem 43.14.

Definition 43.15 ($\beta\eta$-complete development). The *$\beta\eta$-complete development* $M^{*\beta\eta}$ of M is defined as follows: lam:cr:pbe: defn:becd

$$x^{*\beta\eta} = x \tag{43.5}$$ lam:cr:pbe: defn:becd1

$$(\lambda x.\, N)^{*\beta\eta} = \lambda x.\, N^{*\beta\eta} \tag{43.6}$$ lam:cr:pbe: defn:becd2

$$(PQ)^{*\beta\eta} = P^{*\beta\eta}Q^{*\beta\eta} \quad \text{if } P \text{ is not a } \lambda\text{-abstract} \tag{43.7}$$ lam:cr:pbe: defn:becd3

$$((\lambda x.\, N)Q)^{*\beta\eta} = N^{*\beta\eta}[Q^{*\beta\eta}/x] \tag{43.8}$$ lam:cr:pbe: defn:becd4

$$(\lambda x.\, Nx)^{*\beta\eta} = N^{*\beta\eta} \quad \text{if } x \notin FV(N) \tag{43.9}$$ lam:cr:pbe: defn:becd5

Lemma 43.16. *If $M \xRightarrow{\beta\eta} M'$ and $R \xRightarrow{\beta\eta} R'$, then $M[R/y] \xRightarrow{\beta\eta} M'[R'/y]$.* *lam:cr:pbe: lem:comp*

Proof. By induction on the derivation of $M \xRightarrow{\beta\eta} M'$.

The first four cases are exactly like those in Lemma 43.6. If the last rule is (5), then M is $\lambda x.\, Nx$, M' is N' for some x and N' where $x \notin FV(N)$, and $N \xRightarrow{\beta\eta} N'$. We want to show that $(\lambda x.\, Nx)[R/y] \xRightarrow{\beta\eta} N'[R'/y]$, i.e., $\lambda x.\, N[R/y]x \xRightarrow{\beta\eta} N'[R'/y]$. It follows by Definition 43.13(5) and the induction hypothesis. □

Lemma 43.17. *If $M \xRightarrow{\beta\eta} M'$ then $M' \xRightarrow{\beta\eta} M^{*\beta\eta}$.* *lam:cr:pbe: lem:cont*

Proof. By induction on the derivation of $M \xRightarrow{\beta\eta} M'$.

The first four cases are like those in Lemma 43.7. If the last rule is (5), then M is $\lambda x.\, Nx$ and M' is N' for some x, N, N' where $x \notin FV(N)$ and $N \xRightarrow{\beta\eta} N'$. We want to show that $N' \xRightarrow{\beta\eta} (\lambda x.\, Nx)^{*\beta\eta}$, i.e., $N' \xRightarrow{\beta\eta} N^{*\beta\eta}$, which is immediate by induction hypothesis. □

Theorem 43.18. $\xRightarrow{\beta\eta}$ *has the Church-Rosser property.* *lam:cr:pbe: thm:cr*

Proof. Immediate from Lemma 43.17. □

---

content/lambda-calculus/church-rosser/beta-eta-reduction.tex

43.5 $\beta\eta$-reduction

lam:cr:be:sec The Church–Rosser property holds for $\beta\eta$-reduction ($\xrightarrow{\beta\eta}\!\!\!\!\!\rightarrow$).

lam:cr:be:lem:one-par **Lemma 43.19.** *If $M \xrightarrow{\beta\eta} M'$, then $M \xRightarrow{\beta\eta} M'$.*

Proof. By induction on the derivation of $M \xrightarrow{\beta\eta} M'$. If $M \xrightarrow{\beta} M'$ by η-conversion (i.e., Definition 42.43), we use Theorem 43.14. The other cases are as in Lemma 43.9. □

lam:cr:be:lem:par-red **Lemma 43.20.** *If $M \xRightarrow{\beta\eta} M'$, then $M \xrightarrow{\beta\eta}\!\!\!\!\!\rightarrow M'$.*

Proof. Induction on the derivation of $M \xRightarrow{\beta\eta} M'$.

If the last rule is (5), then M is $\lambda x.\, Nx$ and M' is N' for some x, N, N' where $x \notin FV(N)$ and $N \xRightarrow{\beta\eta} N'$. Thus we can first reduce $\lambda x.\, Nx$ to N by η-conversion, followed by the series of $\xrightarrow{\beta\eta}$ steps that show that $N \xrightarrow{\beta\eta}\!\!\!\!\!\rightarrow N'$, which holds by induction hypothesis. □

lam:cr:be:lem:str **Lemma 43.21.** *$\xrightarrow{\beta\eta}\!\!\!\!\!\rightarrow$ is the smallest transitive relation containing $\xRightarrow{\beta\eta}$.*

Proof. As in Lemma 43.11 □

lam:cr:be:thm:cr **Theorem 43.22.** *$\xrightarrow{\beta\eta}\!\!\!\!\!\rightarrow$ satisfies Church–Rosser property.*

Proof. By Theorem 43.2, Theorem 43.18 and Lemma 43.21. □

Chapter 44

Lambda Definability

This chapter is experimental. It needs more explanation, and the material should be structured better into definitions and propositions with proofs, and more examples.

---

content/lambda-calculus/lambda-definability/introduction.tex

44.1 Introduction

lam:ldf:int:sec

At first glance, the lambda calculus is just a very abstract calculus of expressions that represent functions and applications of them to others. Nothing in the syntax of the lambda calculus suggests that these are functions of particular kinds of objects, in particular, the syntax includes no mention of natural numbers. Its basic operations—application and lambda abstractions—are operations that apply to any function, not just functions on natural numbers.

Nevertheless, with some ingenuity, it is possible to define arithmetical functions, i.e., functions on the natural numbers, in the lambda calculus. To do this, we define, for each natural number $n \in \mathbb{N}$, a special λ-term $\overline{n}$, the *Church numeral* for n. (Church numerals are named for Alonzo Church.)

Definition 44.1. If $n \in \mathbb{N}$, the corresponding *Church numeral* $\overline{n}$ represents n:

$$\overline{n} \equiv \lambda f x.\, f^n(x)$$

Here, $f^n(x)$ stands for the result of applying f to x n times. For example, $\overline{0}$ is $\lambda f x.\, x$, and $\overline{3}$ is $\lambda f x.\, f(f(f\, x))$.

The Church numeral $\overline{n}$ is encoded as a lambda term which represents a function accepting two arguments f and x, and returns $f^n(x)$. Church numerals are evidently in normal form.

A representation of natural numbers in the lambda calculus is only useful, of course, if we can compute with them. Computing with Church numerals in the lambda calculus means applying a λ-term F to such a Church numeral, and reducing the combined term $F\,\overline{n}$ to a normal form. If it always reduces to a normal form, and the normal form is always a Church numeral $\overline{m}$, we can think of the output of the computation as being the number m. We can then think of F as defining a function $f\colon \mathbb{N} \to \mathbb{N}$, namely the function such that $f(n) = m$ iff $F\,\overline{n} \twoheadrightarrow \overline{m}$. Because of the Church–Rosser property, normal forms are unique if they exist. So if $F\,\overline{n} \twoheadrightarrow \overline{m}$, there can be no other term in normal form, in particular no other Church numeral, that $F\,\overline{n}$ reduces to.

Conversely, given a function $f\colon \mathbb{N} \to \mathbb{N}$, we can ask if there is a term F that defines f in this way. In that case we say that F *λ-defines* f, and that f is λ-definable. We can generalize this to many-place and partial functions.

Definition 44.2. Suppose $f\colon \mathbb{N}^k \to \mathbb{N}$. We say that a lambda term F *λ-defines* f if for all $n_0, \ldots, n_{k-1}$,

$$F\,\overline{n_0}\,\overline{n_1}\ldots\overline{n_{k-1}} \twoheadrightarrow \overline{f(n_0, n_1, \ldots, n_{k-1})}$$

if $f(n_0, \ldots, n_{k-1})$ is defined, and $F\,\overline{n_0}\,\overline{n_1}\ldots\overline{n_{k-1}}$ has no normal form otherwise.

A very simple example are the constant functions. The term $C_k \equiv \lambda x.\,\overline{k}$ λ-defines the function $c_k\colon \mathbb{N} \to \mathbb{N}$ such that $c(n) = k$. For $C_k\,\overline{n} \equiv (\lambda x.\,\overline{k})\overline{n} \to \overline{k}$ for any n. The identity function is λ-defined by $\lambda x.\,x$. More complex functions are of course harder to define, and often require a lot of ingenuity. So it is perhaps surprising that every computable function is λ-definable. The converse is also true: if a function is λ-definable, it is computable.

---

content/lambda-calculus/lambda-definability/arithmetical-functions.tex

44.2 λ-Definable Arithmetical Functions

lam:rep:arf:sec
lam:rep:arf:prop:succ-ld

Proposition 44.3. *The successor function* succ *is λ-definable.*

Proof. A term that λ-defines the successor function is

$$\text{Succ} \equiv \lambda a.\,\lambda f x.\,f(a f x).$$

Given our conventions, this is short for

$$\text{Succ} \equiv \lambda a.\,\lambda f.\,\lambda x.\,(f((a f)x)).$$

Succ is a function that accepts as argument a number a, and evaluates to another function, $\lambda f x.\,f(a f x)$. That function is not itself a Church numeral. However, if the argument a is a Church numeral, it reduces to one. Consider:

$$(\lambda a.\,\lambda f x.\,f(a f x))\,\overline{n} \to \lambda f x.\,f(\overline{n} f x).$$

The embedded term $\overline{n}fx$ is a redex, since $\overline{n}$ is $\lambda fx.\, f^n x$. So $\overline{n}fx \to f^n x$ and so, for the entire term we have

$$\operatorname{Succ}\overline{n} \twoheadrightarrow \lambda fx.\, f(f^n(x)),$$

i.e., $\overline{n+1}$. □

Example 44.4. Let's look at what happens when we apply Succ to $\overline{0}$, i.e., $\lambda fx.\, x$. We'll spell the terms out in full:

$$\begin{aligned}\operatorname{Succ}\overline{0} &\equiv (\lambda a.\, \lambda f.\, \lambda x.\, (f((af)x)))(\lambda f.\, \lambda x.\, x)\\ &\to \lambda f.\, \lambda x.\, (f(((\lambda f.\, \lambda x.\, x)f)x))\\ &\to \lambda f.\, \lambda x.\, (f((\lambda x.\, x)x))\\ &\to \lambda f.\, \lambda x.\, (fx) \equiv \overline{1}\end{aligned}$$

Problem 44.1. The term

$$\operatorname{Succ}' \equiv \lambda n.\, \lambda fx.\, nf(fx)$$

λ-defines the successor function. Explain why.

Proposition 44.5. *The addition function* add *is λ-definable.*

lam:rep:arf: prop:add-ld

Proof. Addition is λ-defined by the terms

$$\operatorname{Add} \equiv \lambda ab.\, \lambda fx.\, af(bfx)$$

or, alternatively,

$$\operatorname{Add}' \equiv \lambda ab.\, a \operatorname{Succ} b.$$

The first addition works as follows: Add first accept two numbers a and b. The result is a function that accepts f and x and returns $af(bfx)$. If a and b are Church numerals $\overline{n}$ and $\overline{m}$, this reduces to $f^{n+m}(x)$, which is identical to $f^n(f^m(x))$. Or, slowly:

$$\begin{aligned}(\lambda ab.\, \lambda fx.\, af(bfx))\overline{n}\,\overline{m} &\to \lambda fx.\, \overline{n}\, f(\overline{m}\, fx)\\ &\to \lambda fx.\, \overline{n}\, f(f^m x)\\ &\to \lambda fx.\, f^n(f^m x) \equiv \overline{n+m}.\end{aligned}$$

The second representation of addition Add′ works differently: Applied to two Church numerals $\overline{n}$ and $\overline{m}$,

$$\operatorname{Add}'\overline{n}\,\overline{m} \to \overline{n} \operatorname{Succ} \overline{m}.$$

But $\overline{n}fx$ always reduces to $f^n(x)$. So,

$$\overline{n}\,\mathrm{Succ}\,\overline{m} \twoheadrightarrow \mathrm{Succ}^n(\overline{m}).$$

And since Succ λ-defines the successor function, and the successor function applied n times to m gives $n+m$, this in turn reduces to $\overline{n+m}$. □

lam:rep:arf: prop:mult-ld

Proposition 44.6. *Multiplication is λ-definable by the term*

$$\mathrm{Mult} \equiv \lambda ab.\,\lambda fx.\,a(bf)x$$

Proof. To see how this works, suppose we apply Mult to Church numerals $\overline{n}$ and $\overline{m}$: $\mathrm{Mult}\,\overline{n}\,\overline{m}$ reduces to $\lambda fx.\,\overline{n}(\overline{m}\,f)x$. The term $\overline{m}f$ defines a function which applies f to its argument m times. Consequently, $\overline{n}(\overline{m}f)x$ applies the function "apply f m times" itself n times to x. In other words, we apply f to x, $n \cdot m$ times. But the resulting normal term is just the Church numeral $\overline{nm}$. □

> We can actually simplify this term further by η-reduction:
>
> $$\mathrm{Mult} \equiv \lambda ab.\,\lambda f.\,a(bf).$$
>
> But then we first have to explain η-reduction.

Problem 44.2. Multiplication can be λ-defined by the term

$$\mathrm{Mult}' \equiv \lambda ab.\,a(\mathrm{Add}\,a)\overline{0}.$$

Explain why this works.

The definition of exponentiation as a λ-term is surprisingly simple:

$$\mathrm{Exp} \equiv \lambda be.\,eb.$$

The first argument b is the base and the second e is the exponent. Intuitively, ef is f^e by our encoding of numbers. If you find it hard to understand, we can still define exponentiation also by iterated multiplication:

$$\mathrm{Exp}' \equiv \lambda be.\,e(\mathrm{Mult}\,b)\overline{1}.$$

Predecessor and subtraction on Church numeral is not as simple as we might think: it requires encoding of pairs.

44.3 Pairs and Predecessor

lam:ldf:pai:sec

Definition 44.7. The pair of M and N (written $\langle M, N\rangle$) is defined as follows:

$$\langle M, N\rangle \equiv \lambda f.\, fMN.$$

Intuitively it is a function that accepts a function, and applies that function to the two elements of the pair. Following this idea we have this constructor, which takes two terms and returns the pair containing them:

$$\text{Pair} \equiv \lambda mn.\, \lambda f.\, fmn$$

Given a pair, we also want to recover its elements. For this we need two access functions, which accept a pair as argument and return the first or second elements in it:

$$\text{Fst} \equiv \lambda p.\, p(\lambda mn.\, m)$$
$$\text{Snd} \equiv \lambda p.\, p(\lambda mn.\, n)$$

Problem 44.3. Explain why the access functions Fst and Snd work.

Now with pairs we can λ-define the predecessor function:

$$\text{Pred} \equiv \lambda n.\, \text{Fst}(n(\lambda p.\, \langle \text{Snd}\, p, \text{Succ}(\text{Snd}\, p)\rangle)\langle \overline{0}, \overline{0}\rangle)$$

Remember that $\overline{n}\, fx$ reduces to $f^n(x)$; in this case f is a function that accepts a pair p and returns a new pair containing the second component of p and the successor of the second component; x is the pair $\langle 0, 0\rangle$. Thus, the result is $\langle 0, 0\rangle$ for $n = 0$, and $\langle \overline{n-1}, \overline{n}\rangle$ otherwise. Pred then returns the first component of the result.

Subtraction can be defined as Pred applied to a, b times:

$$\text{Sub} \equiv \lambda ab.\, b\text{Pred}\, a.$$

---

content/lambda-calculus/lambda-definability/truth-values.tex

44.4 Truth Values and Relations

We can encode truth values in the pure lambda calculus as follows:

lam:ldf:tvr:sec

$$\text{true} \equiv \lambda x.\, \lambda y.\, x$$
$$\text{false} \equiv \lambda x.\, \lambda y.\, y$$

Truth values are represented as *selectors*, i.e., functions that accept two arguments and returning one of them. The truth value true selects its first argument, and false its second. For example, true MN always reduces to M, while false MN always reduces to N.

Definition 44.8. We call a relation $R \subseteq \mathbb{N}^n$ λ-definable if there is a term R such that

$$R\,\overline{n_1}\ldots\overline{n_k} \xrightarrow{\beta\,*} \text{true}$$

whenever $R(n_1, \ldots, n_k)$ and

$$R\,\overline{n_1}\ldots\overline{n_k} \xrightarrow{\beta\,*} \text{false}$$

otherwise.

For instance, the relation IsZero $= \{0\}$ which holds of 0 and 0 only, is λ-definable by

$$\text{IsZero} \equiv \lambda n.\, n(\lambda x.\, \text{false})\,\text{true}.$$

How does it work? Since Church numerals are defined as iterators (functions which apply their first argument n times to the second), we set the initial value to be true, and for every step of iteration, we return false regardless of the result of the last iteration. This step will be applied to the initial value n times, and the result will be true if and only if the step is not applied at all, i.e., when $n = 0$.

On the basis of this representation of truth values, we can further define some truth functions. Here are two, the representations of negation and conjunction:

$$\text{Not} \equiv \lambda x.\, x\,\text{false}\,\text{true}$$
$$\text{And} \equiv \lambda x.\, \lambda y.\, xy\,\text{false}$$

The function "Not" accepts one argument, and returns true if the argument is false, and false if the argument is true. The function "And" accepts two truth values as arguments, and should return true iff both arguments are true. Truth values are represented as selectors (described above), so when x is a truth value and is applied to two arguments, the result will be the first argument if x is true and the second argument otherwise. Now And takes its two arguments x and y, and in return passes y and false to its first argument x. Assuming x is a truth value, the result will evaluate to y if x is true, and to false if x is false, which is just what is desired.

Note that we assume here that only truth values are used as arguments to And. If it is passed other terms, the result (i.e., the normal form, if it exists) may well not be a truth value.

Problem 44.4. Define the functions Or and Xor representing the truth functions of inclusive and exclusive disjunction using the encoding of truth values as λ-terms.

---

content/lambda-calculus/lambda-definability/primitive-recursive-functions.tex

44.5 Primitive Recursive Functions are λ-Definable

Recall that the primitive recursive functions are those that can be defined from the basic functions zero, succ, and P_i^n by composition and primitive recursion. lam:ldf:prf:sec

Lemma 44.9. *The basic primitive recursive functions* zero, succ, *and projections P_i^n are λ-definable.* lam:ldf:prf:lem:basic

Proof. They are λ-defined by the following terms:

$$\begin{aligned}\text{Zero} &\equiv \lambda a.\, \lambda f x.\, x\\ \text{Succ} &\equiv \lambda a.\, \lambda f x.\, f(a f x)\\ \text{Proj}_i^n &\equiv \lambda x_0 \dots x_{n-1}.\, x_i\end{aligned}$$

□

Lemma 44.10. *Suppose the k-ary function f, and n-ary functions $g_0, \dots, g_{k-1}$, are λ-definable by terms F, G_0, ..., G_k, and h is defined from them by composition. Then H is λ-definable.* lam:ldf:prf:lem:comp

Proof. h can be λ-defined by the term

$$H \equiv \lambda x_0 \dots x_{n-1}.\, F\,(G_0 x_0 \dots x_{n-1}) \dots (G_{k-1} x_0 \dots x_{n-1})$$

We leave verification of this fact as an exercise. □

Problem 44.5. Complete the proof of Lemma 44.10 by showing that $H\overline{n_0} \dots \overline{n_{n-1}} \twoheadrightarrow \overline{h(n_0, \dots, n_{n-1})}$.

Note that Lemma 44.10 did not require that f and $g_0, \dots, g_{k-1}$ are primitive recursive; it is only required that they are total and λ-definable.

Lemma 44.11. *Suppose f is an n-ary function and g is an $n+2$-ary function, they are λ-definable by terms F and G, and the function h is defined from f and g by primitive recursion. Then h is also λ-definable.* lam:ldf:prf:lem:prim

Proof. Recall that h is defined by

$$\begin{aligned}h(x_1, \dots, x_n, 0) &= f(x_1, \dots, x_n)\\ h(x_1, \dots, x_n, y+1) &= h(x_1, \dots, x_n, y, h(x_1, \dots, x_n, y)).\end{aligned}$$

Informally speaking, the primitive recursive definition iterates the application of the function h y times and applies it to $f(x_1, \dots, x_n)$. This is reminiscent of the definition of Church numerals, which is also defined as a iterator.

For simplicity, we give the definition and proof for a single additional argument x. The function h is λ-defined by:

$$H \equiv \lambda x.\, \lambda y.\, \text{Snd}(y D \langle \overline{0}, F x \rangle)$$

where

$$D \equiv \lambda p.\, \langle \mathrm{Succ}(\mathrm{Fst}\, p), (G x (\mathrm{Fst}\, p)(\mathrm{Snd}\, p))\rangle$$

The iteration state we maintain is a pair, the first of which is the current y and the second is the corresponding value of h. For every step of iteration we create a pair of new values of y and h; after the iteration is done we return the second part of the pair and that's the final h value. We now prove this is indeed a representation of primitive recursion.

We want to prove that for any n and m, $H\,\overline{n}\,\overline{m} \twoheadrightarrow \overline{h(n,m)}$. To do this we first show that if $D_n \equiv D[\overline{n}/x]$, then $D_n^m\langle \overline{0}, F\,\overline{n}\rangle \twoheadrightarrow \langle \overline{m}, \overline{h(n,m)}\rangle$ We proceed by induction on m.

If $m = 0$, we want $D_n^0\langle \overline{0}, F\,\overline{n}\rangle \twoheadrightarrow \langle \overline{0}, \overline{h(n,0)}\rangle$. But $D_n^0\langle \overline{0}, F\,\overline{n}\rangle$ just is $\langle \overline{0}, F\,\overline{n}\rangle$. Since F λ-defines f, this reduces to $\langle \overline{0}, \overline{f(n)}\rangle$, and since $f(n) = h(n,0)$, this is $\langle \overline{0}, \overline{h(n,0)}\rangle$

Now suppose that $D_n^m\langle \overline{0}, F\,\overline{n}\rangle \twoheadrightarrow \langle \overline{m}, \overline{h(n,m)}\rangle$. We want to show that $D_n^{m+1}\langle \overline{0}, F\,\overline{n}\rangle \twoheadrightarrow \langle \overline{m+1}, \overline{h(n,m+1)}\rangle$.

$$\begin{aligned}
D_n^{m+1}\langle \overline{0}, F\,\overline{n}\rangle &\equiv D_n(D_n^m\langle \overline{0}, F\,\overline{n}\rangle)\\
&\twoheadrightarrow D_n\,\langle \overline{m}, \overline{h(n,m)}\rangle \qquad \text{(by IH)}\\
&\equiv (\lambda p.\, \langle \mathrm{Succ}(\mathrm{Fst}\, p), (G\,\overline{n}(\mathrm{Fst}\, p)(\mathrm{Snd}\, p))\rangle)\langle \overline{m}, \overline{h(n,m)}\rangle\\
&\to \langle \mathrm{Succ}(\mathrm{Fst}\,\langle \overline{m}, \overline{h(n,m)}\rangle),\\
&\qquad (G\,\overline{n}(\mathrm{Fst}\,\langle \overline{m}, \overline{h(n,m)}\rangle)(\mathrm{Snd}\,\langle \overline{m}, \overline{h(n,m)}\rangle))\rangle\\
&\twoheadrightarrow \langle \mathrm{Succ}\,\overline{m}, (G\,\overline{n}\,\overline{m}\,\overline{h(n,m)})\rangle\\
&\twoheadrightarrow \langle \overline{m+1}, \overline{g(n,m,h(n,m))}\rangle
\end{aligned}$$

Since $g(n,m,h(n,m)) = h(n,m+1)$, we are done.

Finally, consider

$$\begin{aligned}
H\,\overline{n}\,\overline{m} &\equiv \lambda x.\, \lambda y.\, \mathrm{Snd}(y(\lambda p.\langle \mathrm{Succ}(\mathrm{Fst}\, p), (G\, x\, (\mathrm{Fst}\, p)\, (\mathrm{Snd}\, p))\rangle)\langle \overline{0}, F x\rangle)\\
&\quad\ \overline{n}\,\overline{m}\\
&\twoheadrightarrow \mathrm{Snd}(\overline{m}\ \underbrace{(\lambda p.\langle \mathrm{Succ}(\mathrm{Fst}\, p), (G\,\overline{n}\,(\mathrm{Fst}\, p)(\mathrm{Snd}\, p))\rangle)}_{D_n}\langle \overline{0}, F\overline{n}\rangle)\\
&\equiv \mathrm{Snd}(\overline{m}\, D_n\, \langle \overline{0}, F\overline{n}\rangle)\\
&\twoheadrightarrow \mathrm{Snd}\,(D_n^m\langle \overline{0}, F\overline{n}\rangle)\\
&\twoheadrightarrow \mathrm{Snd}\,\langle \overline{m}, \overline{h(n,m)}\rangle\\
&\twoheadrightarrow \overline{h(n,m)}.
\end{aligned}$$

□

Proposition 44.12. *Every primitive recursive function is λ-definable.*

Proof. By Lemma 44.9, all basic functions are λ-definable, and by Lemma 44.10 and Lemma 44.11, the λ-definable functions are closed under composition and primitive recursion. □

content/lambda-calculus/lambda-definability/fixpoints.tex

44.6 Fixpoints

lam:ldf:fp:sec

Suppose we wanted to define the factorial function by recursion as a term Fac with the following property:

$$\text{Fac} \equiv \lambda n.\, \text{IsZero}\, n\, \overline{1}(\text{Mult}\, n(\text{Fac}(\text{Pred}\, n)))$$

That is, the factorial of n is 1 if $n = 0$, and n times the factorial of $n - 1$ otherwise. Of course, we cannot define the term Fac this way since Fac itself occurs in the right-hand side. Such recursive definitions involving self-reference are not part of the lambda calculus. Defining a term, e.g., by

$$\text{Mult} \equiv \lambda ab.\, a(\text{Add}\, a)0$$

only involves previously defined terms in the right-hand side, such as Add. We can always remove Add by replacing it with its defining term. This would give the term Mult as a pure lambda term; if Add itself involved defined terms (as, e.g., Add′ does), we could continue this process and finally arrive at a pure lambda term.

However this is not true in the case of recursive definitions like the one of Fac above. If we replace the occurrence of Fac on the right-hand side with the definition of Fac itself, we get:

$$\begin{aligned}\text{Fac} \equiv \lambda n.\, &\text{IsZero}\, n\, \overline{1} \\ &(\text{Mult}\, n((\lambda n.\, \text{IsZero}\, n\, \overline{1}\, (\text{Mult}\, n\, (\text{Fac}(\text{Pred}\, n))))(\text{Pred}\, n)))\end{aligned}$$

and we still haven't gotten rid of Fac on the right-hand side. Clearly, if we repeat this process, the definition keeps growing longer and the process never results in a pure lambda term. Thus this way of defining factorial (or more generally recursive functions) is not feasible.

The recursive definition does tell us something, though: If f were a term representing the factorial function, then the term

$$\text{Fac}' \equiv \lambda g.\, \lambda n.\, \text{IsZero}\, n\, \overline{1}\, (\text{Mult}\, n\, (g(\text{Pred}n)))$$

applied to the term f, i.e., $\text{Fac}'\, f$, also represents the factorial function. That is, if we regard Fac′ as a function accepting a function and returning a function, the value of $\text{Fac}'\, f$ is just f, provided f is the factorial. A function f with the property that $\text{Fac}'\, f \overset{\beta}{=} f$ is called a *fixpoint* of Fac′. So, the factorial is a fixpoint of Fac′.

There are terms in the lambda calculus that compute the fixpoints of a given term, and these terms can then be used to turn a term like Fac′ into the definition of the factorial.

lam:ldf:fp:
defn:Turing-Y

Definition 44.13. The *Y-combinator* is the term:

$$Y \equiv (\lambda ux.\, x(uux))(\lambda ux.\, x(uux)).$$

Theorem 44.14. *Y has the property that $Yg \twoheadrightarrow g(Yg)$ for any term g. Thus, Yg is always a fixpoint of g.*

Proof. Let's abbreviate $(\lambda ux.\, x(uux))$ by U, so that $Y \equiv UU$. Then

$$\begin{aligned} Yg &\equiv (\lambda ux.\, x(uux))U\, g \\ &\twoheadrightarrow (\lambda x.\, x(UUx))g \\ &\twoheadrightarrow g(UUg) \equiv g(Yg). \end{aligned}$$

Since $g(Yg)$ and Yg both reduce to $g(Yg)$, $g(Yg) \overset{\beta}{=} Yg$, so Yg is a fixpoint of g. □

Of course, since Yg is a redex, the reduction can continue indefinitely:

$$\begin{aligned} Yg &\twoheadrightarrow g(Yg) \\ &\twoheadrightarrow g(g(Yg)) \\ &\twoheadrightarrow g(g(g(Yg))) \\ &\dots \end{aligned}$$

So we can think of Yg as g applied to itself infinitely many times. If we apply g to it one additional time, we—so to speak—aren't doing anything extra; g applied to g applied infinitely many times to Yg is still g applied to Yg infinitely many times.

Note that the above sequence of β-reduction steps starting with Yg is infinite. So if we apply Yg to some term, i.e., consider $(Yg)N$, that term will also reduce to infinitely many different terms, namely $(g(Yg))N$, $(g(g(Yg)))N$, ... It is nevertheless possible that some *other* sequence of reduction steps does terminate in a normal form.

Take the factorial for instance. Define Fac as $Y\,\mathrm{Fac}'$ (i.e., a fixpoint of Fac'). Then:

$$\begin{aligned} \mathrm{Fac}\,\overline{3} &\twoheadrightarrow Y\,\mathrm{Fac}'\,\overline{3} \\ &\twoheadrightarrow \mathrm{Fac}'(Y\,\mathrm{Fac}')\,\overline{3} \\ &\equiv (\lambda x.\, \lambda n.\, \mathrm{IsZero}\, n\, \overline{1}\, (\mathrm{Mult}\, n\, (x(\mathrm{Pred}\, n))))\,\mathrm{Fac}\,\overline{3} \\ &\twoheadrightarrow \mathrm{IsZero}\,\overline{3}\,\overline{1}\,(\mathrm{Mult}\,\overline{3}\,(\mathrm{Fac}(\mathrm{Pred}\,\overline{3}))) \\ &\twoheadrightarrow \mathrm{Mult}\,\overline{3}\,(\mathrm{Fac}\,\overline{2}). \end{aligned}$$

Similarly,

$$\begin{aligned} \mathrm{Fac}\,\overline{2} &\twoheadrightarrow \mathrm{Mult}\,\overline{2}\,(\mathrm{Fac}\,\overline{1}) \\ \mathrm{Fac}\,\overline{1} &\twoheadrightarrow \mathrm{Mult}\,\overline{1}\,(\mathrm{Fac}\,\overline{0}) \end{aligned}$$

but

$$\begin{aligned}\mathrm{Fac}\,\overline{0} &\twoheadrightarrow \mathrm{Fac}'(Y\,\mathrm{Fac}')\,\overline{0}\\ &\equiv (\lambda x.\,\lambda n.\,\mathrm{IsZero}\,n\,\overline{1}\,(\mathrm{Mult}\,n\,(x(\mathrm{Pred}\,n))))\,\mathrm{Fac}\,\overline{0}\\ &\twoheadrightarrow \mathrm{IsZero}\,\overline{0}\,\overline{1}\,(\mathrm{Mult}\,\overline{0}\,(\mathrm{Fac}(\mathrm{Pred}\,\overline{0}))).\\ &\twoheadrightarrow \overline{1}.\end{aligned}$$

So together

$$\mathrm{Fac}\,\overline{3} \twoheadrightarrow \mathrm{Mult}\,\overline{3}\,(\mathrm{Mult}\,\overline{2}\,(\mathrm{Mult}\,\overline{1}\,\overline{1})).$$

What goes for Fac′ goes for any recursive definition. Suppose we have a recursive equation

$$g\,x_1\ldots x_n \stackrel{\beta}{=} N$$

where N may contain g and x_1, …, x_n. Then there is always a term $G \equiv (Y\lambda g.\,\lambda x_1\ldots x_n.\,N)$ such that

$$G\,x_1\ldots x_n \stackrel{\beta}{=} N[G/g].$$

For by the fixpoint theorem,

$$\begin{aligned}G \equiv (Y\lambda g.\,\lambda x_1\ldots x_n.\,N) &\twoheadrightarrow \lambda g.\,\lambda x_1\ldots x_n.\,N(Y\lambda g.\,\lambda x_1\ldots x_n.\,N)\\ &\equiv (\lambda g.\,\lambda x_1\ldots x_n.\,N)\,G\end{aligned}$$

and consequently

$$\begin{aligned}G\,x_1\ldots x_n &\twoheadrightarrow (\lambda g.\,\lambda x_1\ldots x_n.\,N)\,G\,x_1\ldots x_n\\ &\twoheadrightarrow (\lambda x_1\ldots x_n.\,N[G/g])\,x_1\ldots x_n\\ &\twoheadrightarrow N[G/g].\end{aligned}$$

The Y combinator of Definition 44.13 is due to Alan Turing. Alonzo Church had proposed a different version which we'll call Y_C:

$$Y_C \equiv \lambda g.\,(\lambda x.\,g(xx))(\lambda x.\,g(xx)).$$

Church's combinator is a bit weaker than Turing's in that $Yg \stackrel{\beta}{=} g(Yg)$ but not $Yg \stackrel{\beta}{\twoheadrightarrow} g(Yg)$. Let V be the term $\lambda x.\,g(xx)$, so that $Y_C \equiv \lambda g.\,VV$. Then

$$\begin{aligned}VV \equiv (\lambda x.\,g(xx))V &\twoheadrightarrow g(VV) \text{ and thus}\\ Y_Cg \equiv (\lambda g.\,VV)g &\twoheadrightarrow VV \twoheadrightarrow g(VV), \text{ but also}\\ g(Y_Cg) \equiv g((\lambda g.\,VV)g) &\twoheadrightarrow g(VV).\end{aligned}$$

In other words, Y_Cg and $g(Y_Cg)$ reduce to a common term $g(VV)$; so $Y_Cg \stackrel{\beta}{=} g(Y_Cg)$. This is often enough for applications.

---

content/lambda-calculus/lambda-definability/minimization.tex

44.7 Minimization

lam:ldf:min:sec

The general recursive functions are those that can be obtained from the basic functions zero, succ, P_i^n by composition, primitive recursion, and regular minimization. To show that all general recursive functions are λ-definable we have to show that any function defined by regular minimization from a λ-definable function is itself λ-definable.

lam:ldf:min:lem:min

Lemma 44.15. *If $f(x_1, \ldots, x_k, y)$ is regular and λ-definable, then g defined by*

$$g(x_1, \ldots, x_k) = \mu y\ f(x_1, \ldots, x_k, y) = 0$$

is also λ-definable.

Proof. Suppose the lambda term F λ-defines the regular function $f(\vec{x}, y)$. To λ-define h we use a search function and a fixpoint combinator:

$$\begin{aligned} \text{Search} &\equiv \lambda g.\, \lambda f\, \vec{x}\, y.\, \text{IsZero}(f\, \vec{x}\, y)\, y\, (g\, \vec{x}(\text{Succ}\, y) \\ H &\equiv \lambda \vec{x}.\, (Y\ \text{Search}) F\, \vec{x}\, \overline{0}, \end{aligned}$$

where Y is any fixpoint combinator. Informally speaking, Search is a self-referencing function: starting with y, test whether $f\, \vec{x}\, y$ is zero: if so, return y, otherwise call itself with $\text{Succ}\, y$. Thus $(Y\ \text{Search}) F \overline{n_1} \ldots \overline{n_k}\, \overline{0}$ returns the least m for which $f(n_1, \ldots, n_k, m) = 0$.

Specifically, observe that

$$(Y\ \text{Search}) F \overline{n_1} \ldots \overline{n_k}\, \overline{m} \twoheadrightarrow \overline{m}$$

if $f(n_1, \ldots, n_k, m) = 0$, or

$$\twoheadrightarrow (Y\ \text{Search}) F\, \overline{n_1} \ldots \overline{n_k}\, \overline{m+1}$$

otherwise. Since f is regular, $f(n_1, \ldots, n_k, y) = 0$ for some y, and so

$$(Y\ \text{Search}) F \overline{n_1} \ldots \overline{n_k}\, \overline{0} \twoheadrightarrow \overline{h(n_1, \ldots, n_k)}.$$ □

Proposition 44.16. *Every general recursive function is λ-definable.*

Proof. By Lemma 44.9, all basic functions are λ-definable, and by Lemma 44.10, Lemma 44.11, and Lemma 44.15, the λ-definable functions are closed under composition, primitive recursion, and regular minimization. □

---

content/lambda-calculus/lambda-definability/partial-recursive-functions.tex

44.8 Partial Recursive Functions are λ-Definable

lam:ldf:par:sec

Partial recursive functions are those obtained from the basic functions by composition, primitive recursion, and unbounded minimization. They differ from general recursive function in that the functions used in unbounded search are not required to be regular. Not requiring regularity means that functions defined by minimization may sometimes not be defined.

At first glance it might seem that the same methods used to show that the (total) general recursive functions are all λ-definable can be used to prove that all partial recursive functions are λ-definable. For instance, the composition of f with g is λ-defined by $\lambda x.\, F(Gx)$ if f and g are λ-defined by terms F and G, respectively. However, when the functions are partial, this is problematic. When $g(x)$ is undefined, meaning Gx has no normal form. In most cases this means that $F(Gx)$ has no normal forms either, which is what we want. But consider when F is $\lambda x.\, \lambda y.\, y$, in which case $F(Gx)$ does have a normal form $(\lambda y.\, y)$.

This problem is not insurmountable, and there are ways to λ-define all partial recursive functions in such a way that undefined values are represented by terms without a normal form. These ways are, however, somewhat more complicated and less intuitive than the approach we have taken for general recursive functions. We record the theorem here without proof:

Theorem 44.17. *All partial recursive functions are λ-definable.*

---

content/lambda-calculus/lambda-definability/lambda-definable-recursive.tex

44.9 λ-Definable Functions are Recursive

lam:dfl:ldr:sec

Not only are all partial recursive functions λ-definable, the converse is true, too. That is, all λ-definable functions are partial recursive.

Theorem 44.18. *If a partial function f is λ-definable, it is partial recursive.*

lam:dfl:ldr:thm:lambda-computable

Proof. We only sketch the proof. First, we arithmetize λ-terms, i.e., systematially assign Gödel numbers to λ-terms, using the usual power-of-primes coding of sequences. Then we define a partial recursive function normalize(t) operating on the Gödel number t of a lambda term as argument, and which returns the Gödel number of the normal form if it has one, or is undefined otherwise. Then define two partial recursive functions toChurch and fromChurch that maps natural numbers to and from the Gödel numbers of the corresponding Church numeral.

Using these recursive functions, we can define the function f as a partial recursive function. There is a λ-term F that λ-defines f. To compute $f(n_1, \ldots, n_k)$, first obtain the Gödel numbers of the corresponding Church numerals using toChurch(n_i), append these to ${}^{\ulcorner}F^{\urcorner}$ to obtain the Gödel number of the term $F\overline{n_1}\ldots\overline{n_k}$. Now use normalize on this Gödel number. If $f(n_1, \ldots, n_k)$

is defined, $F\overline{n_1}\ldots\overline{n_k}$ has a normal form (which must be a Church numeral), and otherwise it has no normal form (and so

$$\text{normalize}({}^{\#}F\overline{n_1}\ldots\overline{n_k}{}^{\#})$$

is undefined). Finally, use fromChurch on the Gödel number of the normalized term. □

Part X

Many-valued Logic

This part contains draft material on propositional many-valued logics.

Chapter 45

Syntax and Semantics

---

content/many-valued-logic/syntax-and-semantics/introduction.tex

45.1 Introduction

mvl:syn:int:sec

In classical logic, we deal with formulas that are built from propositional variables using the propositional connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. When we define a semantics for classical logic, we do so using the two truth values $\mathbb{T}$ and $\mathbb{F}$. We interpret propositional variables in a valuation $\mathfrak{v}$, which assigns these truth values $\mathbb{T}$, $\mathbb{F}$ to the propositional variables. Any valuation then determines a truth value $\overline{\mathfrak{v}}(\varphi)$ for any formula φ, and A formula is satisfied in a valuation $\mathfrak{v}$, $\mathfrak{v} \vDash \varphi$, iff $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$.

Many-valued logics are generalizations of classical two-valued logic by allowing more truth values than just $\mathbb{T}$ and $\mathbb{F}$. So in many-valued logic, a valuation $\mathfrak{v}$ is a function assigning to every propositional variable p one of a range of possible truth values. We'll generally call the set of allowed truth values V. Classical logic is a many-valued logic where $V = \{\mathbb{T}, \mathbb{F}\}$, and the truth value $\overline{\mathfrak{v}}(\varphi)$ is computed using the familiar characteristic truth tables for the connectives.

Once we add additional truth values, we have more than one natural option for how to compute $\overline{\mathfrak{v}}(\varphi)$ for the connectives we read as "and," "or," "not," and "if—then." So a many-valued logic is determined not just by the set of truth values, but also by the *truth functions* we decide to use for each connective. Once these are selected for a many-valued logic $\mathbf{L}$, however, the truth value

$\overline{\mathfrak{v}}_{\mathbf{L}}(\varphi)$ is uniquely determined by the valuation, just like in classical logic. Many-valued logics, like classical logic, are *truth functional*.

With this semantic building blocks in hand, we can go on to define the analogs of the semantic concepts of tautology, entailment, and satisfiability. In classical logic, a formula is a tautology if its truth value $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$ for any $\mathfrak{v}$. In many-valued logic, we have to generalize this a bit as well. First of all, there is no requirement that the set of truth values V contains $\mathbb{T}$. For instance, some many-valued logics use numbers, such as all rational numbers between 0 and 1 as their set of truth values. In such a case, 1 usually plays the rule of $\mathbb{T}$. In other logics, not just one but several truth values do. So, we require that every many-valued logic have a set V^+ of *designated values*. We can then say that a formula is satisfied in a valuation $\mathfrak{v}$, $\mathfrak{v} \vDash_{\mathbf{L}} \varphi$, iff $\overline{\mathfrak{v}}_{\mathbf{L}}(\varphi) \in V^+$. A formula φ is a tautology of the logic, $\vDash_{\mathbf{L}} \varphi$, iff $\overline{\mathfrak{v}}(\varphi) \in V^+$ for any $\mathfrak{v}$. And, finally, we say that φ is entailed by a set of formulas, $\Gamma \vDash_{\mathbf{L}} \varphi$, if every valuation that satisfies all the formulas in Γ also satisfies φ.

---

content/many-valued-logic/syntax-and-semantics/connectives.tex

45.2 Languages and Connectives

mvl:syn:con:sec

Classical propositional logic, and many other logics, use a set supply of *propositional constants* and *connectives*. For instance, we use the following as primitives:

1. The propositional constant for falsity $\bot$.
2. The propositional constant for truth $\top$.
3. The logical connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\to$ (conditional), $\leftrightarrow$ (biconditional)

The same connectives are used in many-valued logics as well. However, it is often useful to include different versions of, say, conjunction, in the same logic, and that would require different symbols to keep the versions separate. Some many-valued logics also include connectives that have no equivalent in classical logic. So, we'll be a bit more general than usual.

Definition 45.1. A *propositional language* consists of a set $\mathcal{L}$ of *connectives*. Each connective $\star$ has an *arity*; a connective of arity n is said to be *n-place*. Connectives of arity 0 are also called *constants*; connectives of arity 1 are called *unary*, and connectives of arity 2, *binary*.

Example 45.2. The standard language of propositional logic $\mathcal{L}_0$ consists of the following connectives (with associated arities): $\bot$ (0) $\neg$ (1), $\wedge$ (2), $\vee$ (2), $\to$ (2). Most logics we consider will use this language. Some logics by tradition an convention use different symbols for some connectives. For instance, in product logic, the conjunction symbol is often $\odot$ instead of $\wedge$. Sometimes

it is convenient to add a new operator, e.g., the determinateness operator $\triangle$ (1-place).

---

content/many-valued-logic/syntax-and-semantics/formulas.tex

45.3 Formulas

mvl:syn:fml:sec
mvl:syn:fml:defn:formulas

Definition 45.3 (Formula). The set $\mathrm{Frm}(\mathcal{L})$ of *formulas* of a propositional language $\mathcal{L}$ is defined inductively as follows:

1. Every propositional variable p_i is an atomic formula.
2. Every 0-place connective (propositional constant) of $\mathcal{L}$ is an atomic formula.
3. If $\star$ is an n-place connective of $\mathcal{L}$, and φ_1, …, φ_n are formulas, then $\star(\varphi_1, \ldots, \varphi_n)$ is a formula.
4. Nothing else is a formula.

If $\star$ is 1-place, then $\star(\varphi_1)$ will often be written simply as $\star\varphi_1$. If $\star$ is 2-place $\star(\varphi_1, \varphi_2)$ will often be written as $(\varphi_1 \star \varphi_2)$.

As usual, we will often silently leave out the outermost parentheses.

Example 45.4. In the standard language $\mathcal{L}_0$, $p_1 \to (p_1 \wedge \neg p_2)$ is a formula. In the language of product logic, it would be written instead as $p_1 \to (p_1 \odot \neg p_2)$. If we add the 1-place $\triangle$ to the language, we would also have formulas such as $\triangle(p_1 \wedge p_2) \to (\triangle p_1 \wedge \triangle p_2)$.

---

content/many-valued-logic/syntax-and-semantics/matrices.tex

45.4 Matrices

mvl:syn:mat:sec

A many-valued logic is defined by its language, its set of truth values V, a subset of designated truth values, and truth functions for its connective. Together, these elements are called a *matrix*.

mvl:syn:mat:defn:matrix

Definition 45.5 (Matrix). A *matrix* for the logic **L** consists of:

1. a set of connectives making up a language $\mathcal{L}$;
2. a set $V \neq \emptyset$ of truth values;
3. a set $V^+ \subseteq V$ of designated truth values;

$\widetilde{\neg}$	
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$

$\widetilde{\wedge}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

$\widetilde{\vee}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$

$\widetilde{\rightarrow}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$

Figure 45.1: Truth functions for classical logic **C**.

mvl:syn:mat: fig:tf-CL

4. for each n-place connective $\star$ in $\mathcal{L}$, a truth function $\widetilde{\star} : V^n \to V$. If $n = 0$, then $\widetilde{\star}$ is just an element of V.

Example 45.6. The matrix for classical logic **C** consists of:

1. The standard propositional language $\mathcal{L}_0$ with $\bot$, $\neg$, $\wedge$, $\vee$, $\rightarrow$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{F}\}$.
3. $\mathbb{T}$ is the only designated value, i.e., $V^+ = \{\mathbb{T}\}$.
4. For $\bot$, we have $\widetilde{\bot} = \mathbb{F}$. The other truth functions are given by the usual truth tables (see Figure 45.1).

---

content/many-valued-logic/syntax-and-semantics/valuations-sat.tex

45.5 Valuations and Satisfaction

mvl:syn:val: sec

Definition 45.7 (Valuations). Let V be a set of truth values. A *valuation* for $\mathcal{L}$ into V is a function $\mathfrak{v}$ assigning an element of V to the propositional variables of the language, i.e., $\mathfrak{v} \colon \mathrm{At}_0 \to V$.

mvl:syn:val: defn:pValue

Definition 45.8. Given a valuation $\mathfrak{v}$ into the set of truth values V of a many-valued logic **L**, define the evaluation function $\overline{\mathfrak{v}} \colon \mathrm{Frm}(\mathcal{L}) \to V$ inductively by:

1. $\overline{\mathfrak{v}}(p_n) = \mathfrak{v}(p_n)$;
2. If $\star$ is a 0-place connective, then $\overline{\mathfrak{v}}(\star) = \widetilde{\star}_{\mathbf{L}}$;
3. If $\star$ is an n-place connective, then

$$\overline{\mathfrak{v}}(\star(\varphi_1, \ldots, \varphi_n)) = \widetilde{\star}_{\mathbf{L}}(\overline{\mathfrak{v}}(\varphi_1), \ldots, \overline{\mathfrak{v}}(\varphi_n)).$$

mvl:syn:val: defn:satisfaction

Definition 45.9 (Satisfaction). The formula φ is *satisfied* by a valuation $\mathfrak{v}$, $\mathfrak{v} \vDash_{\mathbf{L}} \varphi$, iff $\overline{\mathfrak{v}}_{\mathbf{L}}(\varphi) \in V^+$, where V^+ is the set of designated truth values of **L**.

We write $\mathfrak{v} \nvDash_{\mathbf{L}} \varphi$ to mean "not $\mathfrak{v} \vDash_{\mathbf{L}} \varphi$." If Γ is a set of formulas, $\mathfrak{v} \vDash_{\mathbf{L}} \Gamma$ iff $\mathfrak{v} \vDash_{\mathbf{L}} \varphi$ for every $\varphi \in \Gamma$.

---

content/many-valued-logic/syntax-and-semantics/semantic-notions.tex

45.6 Semantic Notions

Suppose a many-valued logic **L** is given by a matrix. Then we can define the usual semantic notions for **L**. mvl:syn:sem:sec

Definition 45.10. 1. A formula φ is *satisfiable* if for some $\mathfrak{v}$, $\mathfrak{v} \vDash \varphi$; it is *unsatisfiable* if for no $\mathfrak{v}$, $\mathfrak{v} \vDash \varphi$;

2. A formula φ is a *tautology* if $\mathfrak{v} \vDash \varphi$ for all valuations v;

3. If Γ is a set of formulas, $\Gamma \vDash \varphi$ ("Γ entails φ") if and only if $\mathfrak{v} \vDash \varphi$ for every valuation $\mathfrak{v}$ for which $\mathfrak{v} \vDash \Gamma$.

4. If Γ is a set of formulas, Γ is *satisfiable* if there is a valuation $\mathfrak{v}$ for which $\mathfrak{v} \vDash \Gamma$, and Γ is *unsatisfiable* otherwise.

We have some of the same facts for these notions as we do for the case of classical logic:

Proposition 45.11. mvl:syn:sem:prop:semanticalfacts

1. φ is a tautology if and only if $\emptyset \vDash \varphi$;

2. If Γ is satisfiable then every finite subset of Γ is also satisfiable;

3. Monotonicity: if $\Gamma \subseteq \Delta$ and $\Gamma \vDash \varphi$ then also $\Delta \vDash \varphi$; mvl:syn:sem:def:monotonicity

4. Transitivity: if $\Gamma \vDash \varphi$ and $\Delta \cup \{\varphi\} \vDash \psi$ then $\Gamma \cup \Delta \vDash \psi$; mvl:syn:sem:def:Cut

Proof. Exercise. □

Problem 45.1. Prove Proposition 45.11

In classical logic we can connect entailment and the conditional. For instance, we have the validity of *modus ponens*: If $\Gamma \vDash \varphi$ and $\Gamma \vDash \varphi \to \psi$ then $\Gamma \vDash \psi$. Another important relationship between $\vDash$ and $\to$ in classical logic is the semantic deduction theorem: $\Gamma \vDash \varphi \to \psi$ if and only if $\Gamma \cup \{\varphi\} \vDash \psi$. These results *do not* always hold in many-valued logics. Whether they do depends on the truth function $\widetilde{\to}$.

---

content/many-valued-logic/syntax-and-semantics/sublogics.tex

45.7 Many-valued logics as sublogics of C

The usual many-valued logics are all defined using matrices in which the value of a truth-function for arguments in $\{\mathbb{T}, \mathbb{F}\}$ agrees with the classical truth functions. Specifically, in these logics, if $x \in \{\mathbb{T}, \mathbb{F}\}$, then $\widetilde{\neg}_{\mathbf{L}}(x) = \widetilde{\neg}_{\mathbf{C}}(x)$, and for $\star$ any one of $\wedge$, $\vee$, $\to$, if $x, y \in \{\mathbb{T}, \mathbb{F}\}$, then $\widetilde{\star}_{\mathbf{L}}(x, y) = \widetilde{\star}_{\mathbf{C}}(x, y)$. In other words, the truth functions for $\neg$, $\wedge$, $\vee$, $\to$ restricted to $\{\mathbb{T}, \mathbb{F}\}$ are exactly the classical truth functions. mvl:syn:sub:sec

mvl:syn:sub: prop:mvl-cl **Proposition 45.12.** *Suppose that a many-valued logic* $\mathbf{L}$ *contains the connectives* $\neg$, $\wedge$, $\vee$, $\to$ *in its language,* $\mathbb{T}, \mathbb{F} \in V$, *and its truth functions satisfy:*

mvl:syn:sub: prop:not 1. $\widetilde{\neg}_{\mathbf{L}}(x) = \widetilde{\neg}_{\mathbf{C}}(x)$ *if* $x = \mathbb{T}$ *or* $x = \mathbb{F}$;

mvl:syn:sub: prop:land 2. $\widetilde{\wedge}_{\mathbf{L}}(x, y) = \widetilde{\wedge}_{\mathbf{C}}(x, y)$,

mvl:syn:sub: prop:lor 3. $\widetilde{\vee}_{\mathbf{L}}(x, y) = \widetilde{\vee}_{\mathbf{C}}(x, y)$,

mvl:syn:sub: prop:lif 4. $\widetilde{\to}_{\mathbf{L}}(x, y) = \widetilde{\to}_{\mathbf{C}}(x, y)$, *if* $x, y \in \{\mathbb{T}, \mathbb{F}\}$.

Then, for any valuation $\mathfrak{v}$ *into* V *such that* $\mathfrak{v}(p) \in \{\mathbb{T}, \mathbb{F}\}$, $\overline{\mathfrak{v}}_{\mathbf{L}}(\varphi) = \overline{\mathfrak{v}}_{\mathbf{C}}(\varphi)$.

Proof. By induction on φ.

1. If $\varphi \equiv p$ is atomic, we have $\overline{\mathfrak{v}}_{\mathbf{L}}(\varphi) = \mathfrak{v}(p) = \overline{\mathfrak{v}}_{\mathbf{C}}(\varphi)$.

2. If $\varphi \equiv \neg B$, we have

$$\begin{aligned}
\overline{\mathfrak{v}}_{\mathbf{L}}(\varphi) &= \widetilde{\neg}_{\mathbf{L}}(\overline{\mathfrak{v}}_{\mathbf{L}}(\psi)) && \text{by Definition 45.8} \\
&= \widetilde{\neg}_{\mathbf{L}}(\overline{\mathfrak{v}}_{\mathbf{C}}(\psi)) && \text{by inductive hypothesis} \\
&= \widetilde{\neg}_{\mathbf{C}}(\overline{\mathfrak{v}}_{\mathbf{C}}(\psi)) && \text{by assumption (1),} \\
& && \text{since } \overline{\mathfrak{v}}_{\mathbf{C}}(\psi) \in \{\mathbb{T}, \mathbb{F}\}, \\
&= \overline{\mathfrak{v}}_{\mathbf{C}}(\varphi) && \text{by Definition 45.8.}
\end{aligned}$$

3. If $\varphi \equiv (\psi \wedge \chi)$, we have

$$\begin{aligned}
\overline{\mathfrak{v}}_{\mathbf{L}}(\varphi) &= \widetilde{\wedge}_{\mathbf{L}}(\overline{\mathfrak{v}}_{\mathbf{L}}(\psi), \overline{\mathfrak{v}}_{\mathbf{L}}(\chi)) && \text{by Definition 45.8} \\
&= \widetilde{\wedge}_{\mathbf{L}}(\overline{\mathfrak{v}}_{\mathbf{C}}(\psi), \overline{\mathfrak{v}}_{\mathbf{C}}(\chi)) && \text{by inductive hypothesis} \\
&= \widetilde{\wedge}_{\mathbf{C}}(\overline{\mathfrak{v}}_{\mathbf{C}}(\psi), \overline{\mathfrak{v}}_{\mathbf{C}}(\chi)) && \text{by assumption (2),} \\
& && \text{since } \overline{\mathfrak{v}}_{\mathbf{C}}(\psi), \overline{\mathfrak{v}}_{\mathbf{C}}(\chi) \in \{\mathbb{T}, \mathbb{F}\}, \\
&= \overline{\mathfrak{v}}_{\mathbf{C}}(\varphi) && \text{by Definition 45.8.}
\end{aligned}$$

The cases where $\varphi \equiv (\psi \vee \chi)$ and $\varphi \equiv (\psi \to \chi)$ are similar. □

Corollary 45.13. *If a many-valued logic satisfies the conditions of Proposition 45.12,* $\mathbb{T} \in V^+$ *and* $\mathbb{F} \notin V^+$, *then* $\vDash_{\mathbf{L}} \subseteq \vDash_{\mathbf{C}}$, *i.e., if* $\Gamma \vDash_{\mathbf{L}} \psi$ *then* $\Gamma \vDash_{\mathbf{C}} \psi$. *In particular, every tautology of* $\mathbf{L}$ *is also a classical tautology.*

Proof. We prove the contrapositive. Suppose $\Gamma \nvDash_{\mathbf{C}} \psi$. Then there is some valuation $\mathfrak{v}\colon \mathrm{At}_0 \to \{\mathbb{T}, \mathbb{F}\}$ such that $\overline{\mathfrak{v}}_{\mathbf{C}}(\varphi) = \mathbb{T}$ for all $\varphi \in \Gamma$ and $\overline{\mathfrak{v}}_{\mathbf{C}}(\psi) = \mathbb{F}$. Since $\mathbb{T}, \mathbb{F} \in V$, the valuation $\mathfrak{v}$ is also a valuation for $\mathbf{L}$. By Proposition 45.12, $\overline{\mathfrak{v}}_{\mathbf{L}}(\varphi) = \mathbb{T}$ for all $\varphi \in \Gamma$ and $\overline{\mathfrak{v}}_{\mathbf{L}}(\psi) = \mathbb{F}$. Since $\mathbb{T} \in V^+$ and $\mathbb{F} \notin V^+$ that means $\mathfrak{v} \vDash_{\mathbf{L}} \Gamma$ and $\mathfrak{v} \nvDash_{\mathbf{L}} \psi$, i.e., $\Gamma \nvDash_{\mathbf{L}} \psi$. □

Chapter 46

Three-valued Logics

content/many-valued-logic/three-valued-logics/introduction.tex

46.1 Introduction

mvl:thr:int:sec

If we just add one more value $\mathbb{U}$ to $\mathbb{T}$ and $\mathbb{F}$, we get a three-valued logic. Even though there is only one more truth value, the possibilities for defining the truth-functions for $\neg$, $\wedge$, $\vee$, and $\rightarrow$ are quite numerous. Then a logic might use any combination of these truth functions, and you also have a choice of making only $\mathbb{T}$ designated, or both $\mathbb{T}$ and $\mathbb{U}$.

We present here a selection of the most well-known three-valued logics, their motivations, and some of their properties.

content/many-valued-logic/three-valued-logics/lukasiewicz.tex

46.2 Łukasiewicz logic

mvl:thr:luk:sec

One of the first published, worked out proposals for a many-valued logic is due to the Polish philosopher Jan Łukasiewicz in 1921. Łukasiewicz was motivated by Aristotle's sea battle problem: It seems that, *today*, the sentence "There will be a sea battle tomorrow" is neither true nor false: its truth value is not yet settled. Łukasiewicz proposed to introduce a third truth value, to such "future contingent" sentences.

> I can assume without contradiction that my presence in Warsaw at a certain moment of next year, e.g., at noon on 21 December, is at the present time determined neither positively nor negatively. Hence it is possible, but not necessary, that I shall be present in Warsaw at the given time. On this assumption the proposition "I shall be in Warsaw at noon on 21 December of next year," can at the present time be neither true nor false. For if it were true now,

> my future presence in Warsaw would have to be necessary, which is contradictory to the assumption. If it were false now, on the other hand, my future presence in Warsaw would have to be impossible, which is also contradictory to the assumption. Therefore the proposition considered is at the moment neither true nor false and must possess a third value, different from "0" or falsity and "1" or truth. This value we can designate by "$\frac{1}{2}$." It represents "the possible," and joins "the true" and "the false" as a third value.

We will use $\mathbb{U}$ for Łukasiewicz's third truth value.[1]

The truth functions for the connectives $\neg$, $\wedge$, and $\vee$ are easy to determine on this interpretation: the negation of a future contingent sentence is also a future contingent sentence, so $\widetilde{\neg}(\mathbb{U}) = \mathbb{U}$. If one conjunct of a conjunction is undetermined and the other is true, the conjunction is also undetermined—after all, depending on how the future contingent conjunct turns out, the conjunction might turn out to be true, and it might turn out to be false. So

$$\widetilde{\wedge}(\mathbb{T}, \mathbb{U}) = \widetilde{\wedge}(\mathbb{U}, \mathbb{T}) = \mathbb{U}.$$

If the other conjunct is false, however, it cannot turn out true, so

$$\widetilde{\wedge}(\mathbb{F}, \mathbb{U}) = \widetilde{\wedge}(\mathbb{F}, \mathbb{U}) = \mathbb{F}.$$

The other values (if the arguments are settled truth values, $\mathbb{T}$ or $\mathbb{F}$, are like in classical logic.

For the conditional, the situation is a little trickier. Suppose q is a future contingent statement. If p is false, then $p \to q$ will be true, regardless of how q turns out, so we should set $\widetilde{\to}(\mathbb{F}, \mathbb{U}) = \mathbb{T}$. And if p is true, then $q \to p$ will be true, regardless of what q turns out to be, so $\widetilde{\to}(\mathbb{U}, \mathbb{T}) = \mathbb{T}$. If p is true, then $p \to q$ might turn out to be true or false, so $\widetilde{\to}(\mathbb{T}, \mathbb{U}) = \mathbb{U}$. Similarly, if p is false, then $q \to p$ might turn out to be true or false, so $\widetilde{\to}(\mathbb{U}, \mathbb{F}) = \mathbb{U}$. This leaves the case where p and q are both future contingents. On the basis of the motivation, we should really assign $\mathbb{U}$ in this case. However, this would make $\varphi \to \varphi$ *not* a tautology. Łukasiewicz had not trouble giving up $\varphi \vee \neg\varphi$ and $\neg(\varphi \wedge \neg\varphi)$, but balked at giving up $\varphi \to \varphi$. So he stipulated $\widetilde{\to}(\mathbb{U}, \mathbb{U}) = \mathbb{T}$.

mvl:thr:luk: def:lukasiewicz

Definition 46.1. Three-valued Łukasiewicz logic is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\neg$, $\wedge$, $\vee$, $\to$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ is the only designated value, i.e., $V^+ = \{\mathbb{T}\}$.
4. Truth functions are given by the following tables:

[1] Łukasiewicz here uses "possible" in a way that is uncommon today, namely to mean possible but not necessary.

$\widetilde{\neg}$	
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$

$\widetilde{\wedge}_{\mathbf{Ł}_3}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

$\widetilde{\vee}_{\mathbf{Ł}_3}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$

$\widetilde{\rightarrow}_{\mathbf{Ł}_3}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$

As can easily be seen, any formula φ containing only $\neg$, $\wedge$, and $\vee$ will take the truth value $\mathbb{U}$ if all its propositional variables are assigned $\mathbb{U}$. So for instance, the classical tautologies $p \vee \neg p$ and $\neg(p \wedge \neg p)$ are not tautologies in $\mathbf{Ł}_3$, since $\overline{\mathfrak{v}}(\varphi) = \mathbb{U}$ whenever $\mathfrak{v}(p) = \mathbb{U}$.

On valuations where $\mathfrak{v}(p) = \mathbb{T}$ or $\mathbb{F}$, $\overline{\mathfrak{v}}(\varphi)$ will coincide with its classical truth value.

Proposition 46.2. *If $\mathfrak{v}(p) \in \{\mathbb{T}, \mathbb{F}\}$ for all p in φ, then $\overline{\mathfrak{v}}_{\boldsymbol{Ł}_3}(\varphi) = \overline{\mathfrak{v}}_{\mathbf{C}}(\varphi)$.*

Problem 46.1. Suppose we define $\overline{\mathfrak{v}}(\varphi \leftrightarrow \psi) = \overline{\mathfrak{v}}((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$ in $\mathbf{Ł}_3$. What truth table would $\leftrightarrow$ have?

Many classical tautologies *are* also tautologies in $\mathbf{Ł}_3$, e.g, $\neg p \rightarrow (p \rightarrow q)$. Just like in classical logic, we can use truth tables to verify this:

p	q	$\neg$	p	$\rightarrow$	$(p$	$\rightarrow$	$q)$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$
$\mathbb{U}$	$\mathbb{F}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$

Problem 46.2. Show that the following are tautologies in $\mathbf{Ł}_3$:

1. $p \rightarrow (q \rightarrow p)$
2. $\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$
3. $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$

(In (2) and (3), take $\varphi \leftrightarrow \psi$ as an abbreviation for $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$, or refer to your solution to Problem 46.1.)

Problem 46.3. Show that the following classical tautologies are not tautologies in $\mathbf{Ł}_3$:

1. $(\neg p \wedge p) \to q)$
2. $((p \to q) \to p) \to p$
3. $(p \to (p \to q)) \to (p \to q)$

One might therefore perhaps think that although not all classical tautologies are tautologies in $\mathbf{Ł}_3$, they should at least take either the value $\mathbb{T}$ or the value $\mathbb{U}$ on every valuation. This is not the case. A counterexample is given by

$$\neg(p \to \neg p) \vee \neg(\neg p \to p)$$

which is $\mathbb{F}$ if p is $\mathbb{U}$.

Problem 46.4. Which of the following relations hold in Łukasiewicz logic? Give a truth table for each.

1. $p, p \to q \vDash q$
2. $\neg\neg p \vDash p$
3. $p \wedge q \vDash p$
4. $p \vDash p \wedge p$
5. $p \vDash p \vee q$

Łukasiewicz hoped to build a logic of possibility on the basis of his three-valued system, by introducing a one-place connective $\Diamond\varphi$ (for "φ is possible") and a corresponding $\Box\varphi$ (for "φ is necessary"):

$\widetilde{\Diamond}$	
$\mathbb{T}$	$\mathbb{T}$
$\mathbb{U}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$

$\widetilde{\Box}$	
$\mathbb{T}$	$\mathbb{T}$
$\mathbb{U}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$

In other words, p is possible iff it is not already settled as false; and p is necessary iff it is already settled as true.

Problem 46.5. Show that $\Box p \leftrightarrow \neg\Diamond\neg p$ and $\Diamond p \leftrightarrow \neg\Box\neg p$ are tautologies in $\mathbf{Ł}_3$, extended with the truth tables for $\Box$ and $\Diamond$.

However, the shortcomings of this proposed modal logic soon became evident: However things turn out, $p \wedge \neg p$ can never turn out to be true. So even if it is not now settled (and therefore undetermined), it should count as impossible, i.e., $\neg\Diamond(p \wedge \neg p)$ should be a tautology. However, if $\mathfrak{v}(p) = \mathbb{U}$, then $\overline{\mathfrak{v}}(\neg\Diamond(p \wedge \neg p)) = \mathbb{U}$. Although Łukasiewicz was correct that two truth values will not be enough to accommodate modal distinctions such as possiblity and necessity, introducing a third truth value is also not enough.

46.3 Kleene logics

mvl:thr:skl:sec

Stephen Kleene introduced two three-valued logics motivated by a logic in which truth values are thought of the outcomes of computational procedures: a procedure may yield $\mathbb{T}$ or $\mathbb{F}$, but it may also fail to terminate. In that case the corresponding truth value is undefined, represented by the truth value $\mathbb{U}$.

To compute the negation of a proposition φ, you would first compute the value of φ, and then return the opposite of the result. If the computation of φ does not terminate, then the entire procedure does not either: so the negation of $\mathbb{U}$ is $\mathbb{U}$.

To compute a conjunction $\varphi \wedge \psi$, there are two options: one can first compute φ, then ψ, and then the result would be $\mathbb{T}$ if the outcome of both is $\mathbb{T}$, and $\mathbb{F}$ otherwise. If either computation fails to halt, the entire procedure does as well. So in this case, the if one conjunct is undefined, the conjunction is as well. The same goes for disjunction.

However, if we can evaluate φ and ψ in parallel, we can do better. Then, if one of the two procedures halts and returns $\mathbb{F}$, we can stop, as the answer must be false. So in that case a conjunction with one false conjunct is false, even if the other conjunct is undefined. Similarly, when computing a disjunction in parallel, we can stop once the procedure for one of the two disjuncts has returned true: then the disjunction must be true. So in this case we can know what the outcome of a compound claim is, even if one of the components is undefined. On this interpretation, we might read $\mathbb{U}$ as "unknown" rather than "undefined."

The two interpretations give rise to Kleene's strong and weak logic. The conditional is defined as equivalent to $\neg\varphi \vee \psi$.

Definition 46.3. *Strong Kleene logic* **Ks** is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\neg$, $\wedge$, $\vee$, $\rightarrow$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ is the only designated value, i.e., $V^+ = \{\mathbb{T}\}$.
4. Truth functions are given by the following tables:

$\widetilde{\neg}$	
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$

$\widetilde{\wedge}_{\mathbf{Ks}}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

$\widetilde{\vee}_{\mathbf{Ks}}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$

$\widetilde{\rightarrow}_{\mathbf{Ks}}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$

Definition 46.4. *Weak Kleene logic* **Kw** is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\neg$, $\wedge$, $\vee$, $\to$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ is the only designated value, i.e., $V^+ = \{\mathbb{T}\}$.
4. Truth functions are given by the following tables:

$\widetilde{\neg}$	
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$

$\widetilde{\wedge}_{\mathbf{Kw}}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{U}$	$\mathbb{F}$

$\widetilde{\vee}_{\mathbf{Kw}}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{T}$
$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$

$\widetilde{\to}_{\mathbf{Kw}}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{T}$

Proposition 46.5. **Ks** *and* **Kw** *have no tautologies.*

Proof. If $\mathfrak{v}(p) = \mathbb{U}$ for all propositional variables p, then any formula φ will have truth value $\overline{\mathfrak{v}}(\varphi) = \mathbb{U}$, since

$$\widetilde{\neg}(\mathbb{U}) = \widetilde{\vee}(\mathbb{U}, \mathbb{U}) = \widetilde{\wedge}(\mathbb{U}, \mathbb{U}) = \widetilde{\to}(\mathbb{U}, \mathbb{U}) = \mathbb{U}$$

in both logics. As $\mathbb{U} \notin V^+$ for either **Ks** or **Kw**, on this valuation, φ will not be designated. □

Although both weak and strong Kleene logic have no tautologies, they have non-trivial consequence relations.

Problem 46.6. Which of the following relations hold in (a) strong and (b) weak Kleene logic? Give a truth table for each.

1. $p, p \to q \vDash q$
2. $p \vee q, \neg p \vDash q$
3. $p \wedge q \vDash p$
4. $p \vDash p \wedge p$
5. $p \vDash p \vee q$

Dmitry Bochvar interpreted $\mathbb{U}$ as "meaningless" and attempted to use it to solve paradoxes such as the Liar paradox by stipulating that paradoxical sentences take the value $\mathbb{U}$. He introduced a logic which is essentially weak Kleene logic extended by additional connectives, two of which are "external negation" and the "is undefined" operator:

$\sim$	
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$

$\widetilde{+}$		
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$

Problem 46.7. Can you define $\sim$ in Bochvar's logic in terms of $\neg$ and $+$, i.e., find a formula with only the propositional variable p and not involving $\sim$ which always takes the same truth value as $\sim p$? Give a truth table to show you're right.

---

content/many-valued-logic/three-valued-logics/goedel.tex

46.4 Gödel logics

Kurt Gödel introduced a sequence of n-valued logics that each contain all formulas valid in intuitionistic logic, and are contained in classical logic. Here is the first interesting one: mvl:thr:god:sec

Definition 46.6. 3-*valued Gödel logic* $\mathbf{G}$ is defined using the matrix: mvl:thr:god:defn:goedel

1. The standard propositional language $\mathcal{L}_0$ with $\bot$, $\neg$, $\wedge$, $\vee$, $\to$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ is the only designated value, i.e., $V^+ = \{\mathbb{T}\}$.
4. For $\bot$, we have $\widetilde{\bot} = \mathbb{F}$. Truth functions for the remaining connectives are given by the following tables:

$\widetilde{\neg}_{\mathbf{G}}$	
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$

$\widetilde{\wedge}_{\mathbf{G}}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

$\widetilde{\vee}_{\mathbf{G}}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$

$\widetilde{\to}_{\mathbf{G}}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$

You'll notice that the truth tables for $\wedge$ and $\vee$ are the same as in Łukasiewicz and strong Kleene logic, but the truth tables for $\neg$ and $\to$ differ for each. In Gödel logic, $\widetilde{\neg}(\mathbb{U}) = \mathbb{F}$. In contrast to Łukasiewicz logic and Kleene logic, $\widetilde{\to}(\mathbb{U}, \mathbb{F}) = \mathbb{F}$; in contrast to Kleene logic (but as in Łukasiewicz logic), $\widetilde{\to}(\mathbb{U}, \mathbb{U}) = \mathbb{T}$.

As the connection to intuitionistic logic alluded to above suggests, $\mathbf{G}_3$ is close to intuitionistic logic. All intuitionistic truths are tautologies in $\mathbf{G}_3$, and

many classical tautologies that are not valid intuitionistically also fail to be tautologies in $\mathbf{G}_3$. For instance, the following are not tautologies:

$$p \vee \neg p \qquad (p \to q) \to (\neg p \vee q)$$
$$\neg\neg p \to p \qquad \neg(\neg p \wedge \neg q) \to (p \vee q)$$
$$((p \to q) \to p) \to p \qquad \neg(p \to q) \to (p \wedge \neg q)$$

However, not every tautology of $\mathbf{G}_3$ is also intuitionistically valid, e.g., $\neg\neg p \vee \neg p$ or $(p \to q) \vee (q \to p)$.

Problem 46.8. Give truth tables to show that the following are tautologies of $\mathbf{G}_3$:

$$\neg\neg p \vee \neg p$$
$$(p \to q) \vee (q \to p)$$
$$\neg(p \wedge q) \to (\neg p \vee \neg q)$$
$$(p \to q) \vee (q \to r) \vee (r \to s)$$

Problem 46.9. Give truth tables that show that the following are not tautologies of $\mathbf{G}_3$

$$(p \to q) \to (\neg p \vee q)$$
$$\neg(\neg p \wedge \neg q) \to (p \vee q)$$
$$((p \to q) \to p) \to p$$
$$\neg(p \to q) \to (p \wedge \neg q)$$

Problem 46.10. Which of the following relations hold in Gödel logic? Give a truth table for each.

1. $p, p \to q \vDash q$
2. $p \vee q, \neg p \vDash q$
3. $p \wedge q \vDash p$
4. $p \vDash p \wedge p$
5. $p \vDash p \vee q$

---

content/many-valued-logic/three-valued-logics/multiple-designation.tex

46.5 Designating not just $\mathbb{T}$

So far the logics we've seen all had the set of designated truth values $V^+ = \{\mathbb{T}\}$, i.e., something counts as true iff its truth value is $\mathbb{T}$. But one might also count something as true if it's just not $\mathbb{F}$. Then one would get a logic by stipulating in the matrix, e.g., that $V^+ = \{\mathbb{T}, \mathbb{U}\}$. mvl:thr:mul:sec

Definition 46.7. The *logic of paradox* **LP** is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\neg$, $\wedge$, $\vee$, $\rightarrow$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ and $\mathbb{U}$ are designated, i.e., $V^+ = \{\mathbb{T}, \mathbb{U}\}$.
4. Truth functions are the same as in strong Kleene logic.

Definition 46.8. Halldén's *logic of nonsense* **Hal** is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\neg$, $\wedge$, $\vee$, $\rightarrow$ and a 1-place connective $+$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ and $\mathbb{U}$ are designated, i.e., $V^+ = \{\mathbb{T}, \mathbb{U}\}$.
4. Truth functions are the same as weak Kleene logic, plus the "is meaningless" operator:

$\widetilde{+}$	
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$

By contrast to the Kleene logics with which they share truth tables, these *do* have tautologies.

Proposition 46.9. *The tautologies of* **LP** *are the same as the tautologies of classical propositional logic.* mvl:thr:mul:prop:LP-taut-CL

Proof. By Proposition 45.12, if $\vDash_{\mathbf{LP}} \varphi$ then $\vDash_{\mathbf{C}} \varphi$. To show the reverse, we show that if there is a valuation $\mathfrak{v}\colon \mathrm{At}_0 \to \{\mathbb{F}, \mathbb{T}, \mathbb{U}\}$ such that $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\varphi) = \mathbb{F}$ then there is a valuation $\mathfrak{v}'\colon \mathrm{At}_0 \to \{\mathbb{F}, \mathbb{T}\}$ such that $\overline{\mathfrak{v}'}_{\mathbf{C}}(\varphi) = \mathbb{F}$. This establishes the result for **LP**, since **Ks** and **LP** have the same characteristic truth functions, and $\mathbb{F}$ is the only truth value of **LP** that is not designated (that is the only difference between **LP** and **Ks**). Thus, if $\nvDash_{\mathbf{LP}} \varphi$, for some valuation $\mathfrak{v}$, $\overline{\mathfrak{v}}_{\mathbf{LP}}(\varphi) = \overline{\mathfrak{v}}_{\mathbf{Ks}}(\varphi) = \mathbb{F}$. By the claim we're proving, $\overline{\mathfrak{v}'}_{\mathbf{C}}(\varphi) = \mathbb{F}$, i.e., $\nvDash_{\mathbf{C}} \varphi$.

To establish the claim, we first define $\mathfrak{v}'$ as

$$\mathfrak{v}'(p) = \begin{cases} \mathbb{T} & \text{if } \mathfrak{v}(p) \in \{\mathbb{T}, \mathbb{U}\} \\ \mathbb{F} & \text{otherwise} \end{cases}$$

We now show by induction on φ that (a) if $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\varphi) = \mathbb{F}$ then $\overline{\mathfrak{v}'}_{\mathbf{C}}(\varphi) = \mathbb{F}$, and (b) if $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\varphi) = \mathbb{T}$ then $\overline{\mathfrak{v}'}_{\mathbf{C}}(\varphi) = \mathbb{T}$

1. Induction basis: $\varphi \equiv p$. By Definition 45.8, $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\varphi) = \mathfrak{v}(p) = \overline{\mathfrak{v}'}_{\mathbf{C}}(\varphi)$, which implies both (a) and (b).

   For the induction step, consider the cases:

2. $\varphi \equiv \neg\psi$.

   a) Suppose $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\neg\psi) = \mathbb{F}$. By the definition of $\widetilde{\neg}_{\mathbf{Ks}}$, $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\psi) = \mathbb{T}$. By inductive hypothesis, case (b), we get $\overline{\mathfrak{v}'}_{\mathbf{C}}(\psi) = \mathbb{T}$, so $\overline{\mathfrak{v}'}_{\mathbf{C}}(\neg\psi) = \mathbb{F}$.

   b) Suppose $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\neg\psi) = \mathbb{T}$. By the definition of $\widetilde{\neg}_{\mathbf{Ks}}$, $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\psi) = \mathbb{F}$. By inductive hypothesis, case (a), we get $\overline{\mathfrak{v}'}_{\mathbf{C}}(\psi) = \mathbb{F}$, so $\overline{\mathfrak{v}'}_{\mathbf{C}}(\neg\psi) = \mathbb{T}$.

3. $\varphi \equiv (\psi \wedge \chi)$.

   a) Suppose $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\psi \wedge \chi) = \mathbb{F}$. By the definition of $\widetilde{\wedge}_{\mathbf{Ks}}$, $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\psi) = \mathbb{F}$ or $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\psi) = \mathbb{F}$. By inductive hypothesis, case (a), we get $\overline{\mathfrak{v}'}_{\mathbf{C}}(\psi) = \mathbb{F}$ or $\overline{\mathfrak{v}'}_{\mathbf{C}}(\chi) = \mathbb{F}$, so $\overline{\mathfrak{v}'}_{\mathbf{C}}(\psi \wedge \chi) = \mathbb{F}$.

   b) Suppose $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\psi \wedge \chi) = \mathbb{T}$. By the definition of $\widetilde{\wedge}_{\mathbf{Ks}}$, $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\psi) = \mathbb{T}$ and $\overline{\mathfrak{v}}_{\mathbf{Ks}}(\psi) = \mathbb{T}$. By inductive hypothesis, case (b), we get $\overline{\mathfrak{v}'}_{\mathbf{C}}(\psi) = \mathbb{T}$ and $\overline{\mathfrak{v}'}_{\mathbf{C}}(\chi) = \mathbb{T}$, so $\overline{\mathfrak{v}'}_{\mathbf{C}}(\psi \wedge \chi) = \mathbb{T}$.

The other two cases are similar, and left as exercises. Alternatively, the proof above establishes the result for all formulas only containing $\neg$ and $\wedge$. One may now appeal to the facts that in both **Ks** and **C**, for any $\mathfrak{v}$, $\overline{\mathfrak{v}}(\psi \vee \chi) = \overline{\mathfrak{v}}(\neg(\neg\psi \wedge \neg\chi))$ and $\overline{\mathfrak{v}}(\psi \to \chi) = \overline{\mathfrak{v}}(\neg(\psi \wedge \neg\chi))$. □

Problem 46.11. Complete the proof Proposition 46.9, i.e., establish (a) and (b) for the cases where $\varphi \equiv (\psi \vee \chi)$ and $\varphi \equiv (\psi \to \chi)$.

Problem 46.12. Prove that every classical tautology is a tautology in **Hal**.

Although they have the same tautologies as classical logic, their consequence relations are different. **LP**, for instance, is *paraconsistent* in that $\neg p, p \nvDash q$, and so the principle of explosion $\neg\varphi, \varphi \vDash \psi$ does not hold in general. (It holds for some cases of φ and ψ, e.g., if ψ is a tautology.)

Problem 46.13. Which of the following relations hold in (a) **LP** and in (b) **Hal**? Give a truth table for each.

1. $p, p \to q \vDash q$
2. $\neg q, p \to q \vDash \neg p$
3. $p \vee q, \neg p \vDash q$
4. $\neg p, p \vDash q$
5. $p \vDash p \vee q$

6. $p \to q, q \to r \vDash p \to r$

What if you make $\mathbb{U}$ designated in $\mathbf{Ł}_3$?

Definition 46.10. The logic *3-valued R-Mingle* $\mathbf{RM}_3$ is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\bot$, $\neg$, $\wedge$, $\vee$, $\to$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ and $\mathbb{U}$ are designated, i.e., $V^+ = \{\mathbb{T}, \mathbb{U}\}$.
4. Truth functions are the same as Łukasiewicz logic $\mathbf{Ł}_3$.

Problem 46.14. Which of the following relations hold in $\mathbf{RM}_3$?

1. $p, p \to q \vDash q$
2. $p \vee q, \neg p \vDash q$
3. $\neg p, p \vDash q$
4. $p \vDash p \vee q$

Different truth tables can sometimes generate the same logic (entailment relation) just by changing the designated values. E.g., this happens if in Gödel logic we take $V^+ = \{\mathbb{T}, \mathbb{U}\}$ instead of $\{\mathbb{T}\}$.

Proposition 46.11. *The matrix with $V = \{\mathbb{F}, \mathbb{U}, \mathbb{T}\}$, $V^+ = \{\mathbb{T}, \mathbb{U}\}$, and the truth functions of 3-valued Gödel logic defines classical logic.* *mvl:thr:mul: prop:gl-udes*

Proof. Exercise. □

Problem 46.15. Prove Proposition 46.11 by showing that for the logic $\mathbf{L}$ defined just like Gödel logic but with $V^+ = \{\mathbb{T}, \mathbb{U}\}$, if $\Gamma \nvDash_{\mathbf{L}} \psi$ then $\Gamma \nvDash_{\mathbf{C}} \psi$. Use the ideas of Proposition 46.9, except instead of proving properties (a) and (b), show that $\overline{\mathfrak{v}}_{\mathbf{G}}(\varphi) = \mathbb{F}$ iff $\overline{\mathfrak{v}'}_{\mathbf{C}}(\varphi) = \mathbb{F}$ (and hence that $\overline{\mathfrak{v}}_{\mathbf{G}}(\varphi) \in \{\mathbb{T}, \mathbb{U}\}$ iff $\overline{\mathfrak{v}'}_{\mathbf{C}}(\varphi) = \mathbb{T}$). Explain why this establishes the proposition.

Chapter 47

Infinite-valued Logics

content/many-valued-logic/infinite-valued-logics/introduction.tex

47.1 Introduction

mvl:inf:int:sec

The number of truth values of a matrix need not be finite. An obvious choice for a set of infinitely many truth values is the set of rational numbers between 0 and 1, $V_\infty = [0,1] \cap \mathbb{Q}$, i.e.,

$$V_\infty = \{\frac{n}{m} : n, m \in \mathbb{N} \text{ and } n \leq m\}.$$

When considering this infinite truth value set, it is often useful to also consider the subsets

$$V_m = \{\frac{n}{m-1} : n \in \mathbb{N} \text{ and } n \leq m\}$$

For instance, V_5 is the set with 5 evenly spaced truth values,

$$V_5 = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}.$$

In logics based on these truth value sets, usually only 1 is designated, i.e., $V^+ = \{1\}$. In other words, we let 1 play the role of (absolute) truth, 0 as absolute falsity, but formulas may take any intermediate value in V.

One can also consider the set $V_{[0,1]} = [0,1]$ of all *real* numbers between 0 and 1, or other infinite subsets of $[0,1]$, however. Logics with this truth value set are often called *fuzzy*.

content/many-valued-logic/infinite-valued-logics/lukasiewicz.tex

47.2 Łukasiewicz logic

mvl:inf:luk:sec

This is a short "stub" of a section on infinite-valued Łukasiewicz logic.

Definition 47.1. Infinite-valued Łukasiewicz logic $\mathbf{Ł}_\infty$ is defined using the matrix: mvl:inf:luk:def:lukasiewicz

1. The standard propositional language $\mathcal{L}_0$ with $\neg$, $\wedge$, $\vee$, $\to$.
2. The set of truth values V_∞.
3. 1 is the only designated value, i.e., $V^+ = \{1\}$.
4. Truth functions are given by the following functions:

$$\begin{aligned}
\widetilde{\neg}_{\mathbf{Ł}}(x) &= 1 - x \\
\widetilde{\wedge}_{\mathbf{Ł}}(x, y) &= \min(x, y) \\
\widetilde{\vee}_{\mathbf{Ł}}(x, y) &= \max(x, y) \\
\widetilde{\to}_{\mathbf{Ł}}(x, y) &= \min(1, 1 - (x - y)) = \begin{cases} 1 & \text{if } x \leq y \\ 1 - (x - y) & \text{otherwise.} \end{cases}
\end{aligned}$$

m-valued Łukasiewicz logic is defined the same, except $V = V_m$.

Proposition 47.2. *The logic $\mathbf{Ł}_3$ defined by Definition 46.1 is the same as $\mathbf{Ł}_3$ defined by Definition 47.1.*

Proof. This can be seen by comparing the truth tables for the connectives given in Definition 46.1 with the truth tables determined by the equations in Definition 47.1:

$\widetilde{\neg}$	
1	0
1/2	1/2
0	1

$\widetilde{\wedge}_{\mathbf{Ł}_3}$	1	1/2	0
1	1	1/2	0
1/2	1/2	1/2	0
0	0	0	0

$\widetilde{\vee}_{\mathbf{Ł}_3}$	1	1/2	0
1	1	1	1
1/2	1	1/2	1/2
0	1	1/2	0

$\widetilde{\to}_{\mathbf{Ł}_3}$	1	1/2	0
1	1	1/2	0
1/2	1	1	1/2
0	1	1	1

□

Proposition 47.3. *If $\Gamma \vDash_{\mathbf{Ł}_\infty} \psi$ then $\Gamma \vDash_{\mathbf{Ł}_m} \psi$ for all $m \geq 2$.* mvl:inf:luk:prop:luk-infty-m

Proof. Exercise. □

Problem 47.1. Prove Proposition 47.3.

In fact, the converse holds as well.

Infinite-valued Łukasiewicz logic is the most popular fuzzy logic. In the fuzzy logic literature, the conditional is often defined as $\neg\varphi \vee \psi$. The result would be an infinite-valued strong Kleene logic.

Problem 47.2. Show that $(p \to q) \vee (q \to p)$ is a tautology of $\mathbf{Ł}_\infty$.

---

content/many-valued-logic/infinite-valued-logics/goedel.tex

47.3 Gödel logics

mvl:inf:god:sec

This is a short "stub" of a section on infinite-valued Gödel logic.

mvl:inf:god:def:goedel

Definition 47.4. Infinite-valued Gödel logic $\mathbf{G}_\infty$ is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\bot$, $\neg$, $\wedge$, $\vee$, $\to$.
2. The set of truth values V_∞.
3. 1 is the only designated value, i.e., $V^+ = \{1\}$.
4. Truth functions are given by the following functions:

$$\widetilde{\bot} = 0$$
$$\widetilde{\neg}_{\mathbf{G}}(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$
$$\widetilde{\wedge}_{\mathbf{G}}(x, y) = \min(x, y)$$
$$\widetilde{\vee}_{\mathbf{G}}(x, y) = \max(x, y)$$
$$\widetilde{\to}_{\mathbf{G}}(x, y) = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise.} \end{cases}$$

m-valued Gödel logic is defined the same, except $V = V_m$.

Proposition 47.5. *The logic* $\mathbf{G}_3$ *defined by Definition 46.6 is the same as* $\mathbf{G}_3$ *defined by Definition 47.4.*

Proof. This can be seen by comparing the truth tables for the connectives given in Definition 46.6 with the truth tables determined by the equations in Definition 47.4:

$\widetilde{\neg}_{\mathbf{G}_3}$	
1	0
1/2	0
0	1

$\widetilde{\wedge}_{\mathbf{G}}$	1	1/2	0
1	1	1/2	0
1/2	1/2	1/2	0
0	0	0	0

$\widetilde{\vee}_{\mathbf{G}}$	1	1/2	0
1	1	1	1
1/2	1	1/2	1/2
0	1	1/2	0

$\widetilde{\rightarrow}_{\mathbf{G}}$	1	1/2	0
1	1	1/2	0
1/2	1	1	0
0	1	1	1

□

Proposition 47.6. *If $\Gamma \vDash_{\mathbf{G}_\infty} \psi$ then $\Gamma \vDash_{\mathbf{G}_m} \psi$ for all $m \geq 2$.*

mvl:inf:god: prop:god-infty-m

Proof. Exercise. □

Problem 47.3. Prove Proposition 47.6.

In fact, the converse holds as well.

Like $\mathbf{G}_3$, $\mathbf{G}_\infty$ has all intuitionistically valid formulas as tautologies, and the same examples of non-tautologies are non-tautologies of $\mathbf{G}_\infty$:

$$p \vee \neg p \qquad (p \to q) \to (\neg p \vee q)$$
$$\neg\neg p \to p \qquad \neg(\neg p \wedge \neg q) \to (p \vee q)$$
$$((p \to q) \to p) \to p \qquad \neg(p \to q) \to (p \wedge \neg q)$$

The example of an intuitionistically invalid formula that is nevertheless a tautology of $\mathbf{G}_3$, $(p \to q) \vee (q \to p)$, is also a tautology in $\mathbf{G}_\infty$. In fact, $\mathbf{G}_\infty$ can be characterized as intuitionistic logic to which the schema $(\varphi \to \psi) \vee (\psi \to \varphi)$ is added. This was shown by Michael Dummett, and so $\mathbf{G}_\infty$ is often referred to as Gödel–Dummett logic **LC**.

Problem 47.4. Show that $(p \to q) \vee (q \to p)$ is a tautology of $\mathbf{G}_\infty$.

Problem 47.5. Show that $(p \to q) \vee (q \to r) \vee (r \to s)$, which is a tautology of $\mathbf{G}_3$, is not a tautology of $\mathbf{G}_\infty$.

Chapter 48

Sequent Calculus

content/many-valued-logic/sequent-calculus/introduction.tex

48.1 Introduction

mvl:seq:int:sec

The sequent calculus for classical logic is an efficient and simple derivation system. If a many-valued logic is defined by a matrix with finitely many truth values, i.e., V is finite, it is possible to provide a sequent calculus for it. The idea for how to do this comes from considering the meanings of sequents and the form of inference rules in the classical case.

Now recall that a sequent

$$\varphi_1, \dots, \varphi_n \Rightarrow \psi_1, \dots, \psi_n$$

can be interpreted as the formula

$$(\varphi_1 \wedge \dots \wedge \varphi_m) \to (\psi_1 \vee \dots \vee \psi_n)$$

In other words, A valuation $\mathfrak{v}$ *satisfies* a sequent $\Gamma \Rightarrow \Delta$ iff either $\overline{\mathfrak{v}}(\varphi) = \mathbb{F}$ for some $\varphi \in \Gamma$ or $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$ for some $\varphi \in \Delta$. On this interpretation, initial sequents $\varphi \Rightarrow \varphi$ are always satisfied, because either $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$ or $\overline{(}\varphi) = \mathbb{F}$.

Here are the inference rules for the conditional in **LK**, with side formulas Γ, Δ left out:

$$\frac{\Rightarrow \varphi \qquad \psi \Rightarrow}{\varphi \to \psi \Rightarrow} \to\text{L} \qquad\qquad \frac{\varphi \Rightarrow \psi}{\Rightarrow \varphi \to \psi} \to\text{R}$$

If we apply the above semantic interpretation of a sequent, we can read the $\to$L rule as saying that if $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$ and $\overline{\mathfrak{v}}(\psi) = \mathbb{F}$, then $\overline{\mathfrak{v}}(\varphi \to \psi) = \mathbb{F}$. Similarly, the $\to$R rule says that if either $\overline{\mathfrak{v}}(\varphi) = \mathbb{F}$ or $\overline{\mathfrak{v}}(\psi) = \mathbb{T}$, then $\overline{\mathfrak{v}}(\varphi \to \psi) = \mathbb{T}$. And in fact, these conditionals are actually biconditionals. In the case of the $\wedge$L and $\vee$R rules in their standard formulation, the corresponding conditionals would not be biconditionals. But there are alternative versions of these rules where they are:

$$\frac{\varphi, \psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta} \wedge\text{L} \qquad\qquad \frac{\Gamma \Rightarrow \Delta, \varphi, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee\text{R}$$

This basic idea, applied to an n-valued logic, then results in a sequent calculus with n instead of two places, one for each truth value. For a three-valued logic with $V = \{\mathbb{F}, \mathbb{U}, \mathbb{T}\}$, a sequent is an expression $\Gamma \mid \Pi \mid \Delta$. It is satisfied in a valuation $\mathfrak{v}$ iff either $\overline{\mathfrak{v}}(\varphi) = \mathbb{F}$ for some $\varphi \in \Gamma$ or $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$ for some $\varphi \in \Delta$ or $\overline{\mathfrak{v}}(\varphi) = \mathbb{U}$ for some $\varphi \in \Pi$. Consequently, initial sequents $\varphi \mid \varphi \mid \varphi$ are always satisfied.

---

content/many-valued-logic/sequent-calculus/rules-and-proofs.tex

48.2 Rules and Derivations

For the following, let $\Gamma, \Delta, \Pi, \Lambda$ represent finite sequences of sentences. mvl:seq:rul:sec

Definition 48.1 (Sequent). An *n-sided sequent* is an expression of the form

$$\Gamma_1 \mid \ldots \mid \Gamma_n$$

where each Γ_1 is a finite (possibly empty) sequences of sentences of the language $\mathcal{L}$.

Definition 48.2 (Initial Sequent). An *n-sided initial sequent* is an n-sided sequent of the form $\varphi \mid \ldots \mid \varphi$ for any sentence φ in the language.

If the language contains a 0-place connective $\star$, i.e., a propositional constant, then we also take the sequent $\ldots \mid \star \mid \ldots$ where $\star$ appears in the space for the truth value associated with $\widetilde{\star} \in V$, and is empty otherwise.

For each connective of an n-valued logic $\mathbf{L}$, there is a logical rule for each truth value that this connective can take in $\mathbf{L}$. Derivations in an n-sided sequent calculus for $\mathbf{L}$ are trees of sequents, where the topmost sequents are initial sequents, and if a sequent stands below one or more other sequents, it must follow correctly by a rule of inference for the connectives of $\mathbf{L}$.

Definition 48.3 (Theorems). A sentence φ is a *theorem* of an n-valued logic $\mathbf{L}$ if there is a derivation of the n-sequent containing φ in each position corresponding to a designated truth value of $\mathbf{L}$. We write $\vdash_{\mathbf{L}} \varphi$ if φ is a theorem and $\nvdash_{\mathbf{L}} \varphi$ if it is not.

Definition 48.4 (Derivability). A sentence φ is *derivable from* a set of sentences Γ in an n-valued logic $\mathbf{L}$, $\Gamma \vdash_{\mathbf{L}} \varphi$, iff there is a finite subset $\Gamma_0 \subseteq \Gamma$ and a sequence Γ_0' of the sentences in Γ_0 such that the following sequent has a derivation:

$$\Lambda_1 \mid \ldots \mid \Lambda_n$$

where Λ_i is φ if position i corresponds to a designated truth value, and Γ_0'otherwise. If φ is not derivable from Γ we write $\Gamma \nvdash \varphi$.

For instance, 3-valued Łukasiewicz logic has a 3-sided sequent calculus. In a 3-sided sequent $\Gamma \mid \Pi \mid \Delta$, Γ corresponds to $\mathbb{F}$, Δ to $\mathbb{T}$, and Π to $\mathbb{U}$. Axioms are $\varphi \mid \varphi \mid \varphi$. Since only $\mathbb{T}$ is designated, $\Gamma \vdash_{Ł_3} \varphi$ iff the sequent $\Gamma \mid \Gamma \mid \varphi$ has a derivation. (If $\mathbb{U}$ were also designated, we would need a derivation of $\Gamma \mid \varphi \mid \varphi$.)

---

content/many-valued-logic/sequent-calculus/structural-rules.tex

48.3 Structural Rules

mvl:seq:str:sec

The structural rules for n-sided sequent calculus operate as in the classical case, except for each position i.

$$\frac{\Gamma_1 \mid \ldots \mid \quad \Gamma_i \mid \ldots \mid \Gamma_n}{\Gamma_1 \mid \ldots \mid \varphi, \Gamma_i \mid \ldots \mid \Gamma_n}\,\mathrm{W}i$$

$$\frac{\Gamma_1 \mid \ldots \mid \varphi, \varphi, \Gamma_i \mid \ldots \mid \Gamma_n}{\Gamma_1 \mid \ldots \mid \quad \varphi, \Gamma_i \mid \ldots \mid \Gamma_n}\,\mathrm{C}i$$

$$\frac{\Gamma_1 \mid \ldots \mid \Gamma_i, \varphi, \psi, \Gamma_i' \mid \ldots \mid \Gamma_n}{\Gamma_1 \mid \ldots \mid \Gamma_i, \psi, \varphi, \Gamma_i' \mid \ldots \mid \Gamma_n}\,\mathrm{X}i$$

A series of weakening, contraction, and exchange inferences will often be indicated by double inference lines.

The Cut rule comes in several forms, one for every combination of distinct positions in the sequent $i \neq j$:

$$\frac{\Gamma_1 \mid \ldots \mid \varphi, \Gamma_i \mid \ldots \mid \Gamma_n \qquad \Delta_1 \mid \ldots \mid \varphi, \Delta_j \mid \ldots \mid \Delta_n}{\Gamma_1, \Delta_1 \mid \ldots \mid \Gamma_n, \Delta_n}\,\mathrm{Cut}i, j$$

---

content/many-valued-logic/sequent-calculus/propositional-rules.tex

48.4 Propositional Rules for Selected Logics

mvl:seq:prl:sec

The inference rules for a connective in an n-sided sequent calculus only depend on the characteristic truth function for the connective. Thus, if some connective is defined by the same truth function in different logics, these n-sided sequent rules for the connective are the same in those logics.

Rules for $\neg$

The following rules for $\neg$ apply to Łukasiewicz and Kleene logics, and their variants.

$$\frac{\Gamma \mid \Pi \mid \Delta, \varphi}{\neg\varphi, \Gamma \mid \Pi \mid \Delta}\,\neg\mathbb{F}$$

$$\frac{\Gamma \mid \varphi, \Pi \mid \Delta}{\Gamma \mid \neg\varphi, \Pi \mid \Delta}\ \neg\mathbb{U}$$

$$\frac{\varphi, \Gamma \mid \Pi \mid \Delta}{\Gamma \mid \Pi \mid \Delta, \neg\varphi}\ \neg\mathbb{T}$$

The following rules for $\neg$ apply to Gödel logic.

$$\frac{\Gamma \mid \varphi, \Pi \mid \Delta, \varphi}{\neg\varphi, \Gamma \mid \Pi \mid \Delta}\ \neg_{\mathbf{G}}\mathbb{F} \qquad \frac{\varphi, \Gamma \mid \Pi \mid \Delta}{\Gamma \mid \Pi \mid \Delta, \neg\varphi}\ \neg_{\mathbf{G}}\mathbb{T}$$

(In Gödel logic, $\neg\varphi$ can never take the value $\mathbb{U}$, so there is no rule for the middle position.)

Rules for $\wedge$

These are the rules for $\wedge$ in Łukasiewicz, strong Kleene, and Gödel logic.

$$\frac{\varphi, \psi, \Gamma \mid \Pi \mid \Delta}{\varphi \wedge \psi, \Gamma \mid \Pi \mid \Delta}\ \wedge\mathbb{F}$$

$$\frac{\Gamma \mid \varphi, \Pi \mid \varphi, \Delta \qquad \Gamma \mid \psi, \Pi \mid \psi, \Delta \qquad \Gamma \mid \varphi, \psi, \Pi \mid \Delta}{\Gamma \mid \varphi \wedge \psi, \Pi \mid \Delta}\ \wedge\mathbb{U}$$

$$\frac{\Gamma \mid \Pi \mid \Delta, \varphi \qquad \Gamma \mid \Pi \mid \Delta, \psi}{\Gamma \mid \Pi \mid \Delta, \varphi \wedge \psi}\ \wedge\mathbb{T}$$

Rules for $\vee$

These are the rules for $\vee$ in Łukasiewicz, strong Kleene, and Gödel logic.

$$\frac{\varphi, \Gamma \mid \Pi \mid \Delta \qquad \psi, \Gamma \mid \Pi \mid \Delta}{\varphi \vee \psi, \Gamma \mid \Pi \mid \Delta}\ \vee\mathbb{F}$$

$$\frac{\varphi, \Gamma \mid \varphi, \Pi \mid \Delta \qquad \psi, \Gamma \mid \psi, \Pi \mid \Delta \qquad \Gamma \mid \varphi, \psi, \Pi \mid \Delta}{\Gamma \mid \varphi \vee \psi, \Pi \mid \Delta}\ \vee\mathbb{U}$$

$$\frac{\Gamma \mid \Pi \mid \Delta, \varphi, \psi}{\Gamma \mid \Pi \mid \Delta, \varphi \vee \psi}\ \vee\mathbb{T}$$

Rules for $\rightarrow$

These are the rules for $\rightarrow$ in Łukasiewicz logic.

$$\frac{\Gamma \mid \Pi \mid \Delta, \varphi \qquad \psi, \Gamma \mid \Pi \mid \Delta}{\varphi \rightarrow \psi, \Gamma \mid \Pi \mid \Delta} \rightarrow_{\mathbf{Ł}_3}\mathbb{F}$$

$$\frac{\Gamma \mid \varphi, \psi, \Pi \mid \Delta \qquad \psi, \Gamma \mid \Pi \mid \Delta, \varphi}{\Gamma \mid \varphi \rightarrow \psi, \Pi \mid \Delta} \rightarrow_{\mathbf{Ł}_3}\mathbb{U}$$

$$\frac{\varphi, \Gamma \mid \psi, \Pi \mid \Delta, \psi \qquad \varphi, \Gamma \mid \varphi, \Pi \mid \Delta, \psi}{\Gamma \mid \Pi \mid \Delta, \varphi \rightarrow \psi} \rightarrow_{\mathbf{Ł}_3}\mathbb{T}$$

These are the rules for $\rightarrow$ in strong Kleene logic.

$$\frac{\Gamma \mid \Pi \mid \Delta, \varphi \qquad \psi, \Gamma \mid \Pi \mid \Delta}{\varphi \rightarrow \psi, \Gamma \mid \Pi \mid \Delta} \rightarrow_{\mathbf{Ks}}\mathbb{F}$$

$$\frac{\psi, \Gamma \mid \psi, \Pi \mid \Delta \qquad \Gamma \mid \varphi, \psi, \Pi \mid \Delta \qquad \Gamma \mid \varphi, \Pi \mid \Delta, \varphi}{\Gamma \mid \varphi \rightarrow \psi, \Pi \mid \Delta} \rightarrow_{\mathbf{Ks}}\mathbb{U}$$

$$\frac{\varphi, \Gamma \mid \Pi \mid \Delta, \psi}{\Gamma \mid \Pi \mid \Delta, \varphi \rightarrow \psi} \rightarrow_{\mathbf{Ks}}\mathbb{T}$$

These are the rules for $\rightarrow$ in Gödel logic.

$$\frac{\Gamma \mid \varphi, \Pi \mid \Delta, \varphi \qquad \psi, \Gamma \mid \Pi \mid \Delta}{\varphi \rightarrow \psi, \Gamma \mid \Pi \mid \Delta} \rightarrow_{\mathbf{G}_3}\mathbb{F}$$

$$\frac{\Gamma \mid \psi, \Pi \mid \Delta \qquad \Gamma \mid \Pi \mid \Delta, \varphi}{\Gamma \mid \varphi \rightarrow \psi, \Pi \mid \Delta} \rightarrow_{\mathbf{G}_3}\mathbb{U}$$

$$\frac{\varphi, \Gamma \mid \psi, \Pi \mid \Delta, \psi \qquad \varphi, \Gamma \mid \varphi, \Pi \mid \Delta, \psi}{\Gamma \mid \Pi \mid \Delta, \varphi \rightarrow \psi} \rightarrow_{\mathbf{G}_3}\mathbb{T}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{A \mid A \mid A}{A \mid A \mid B, A}\,\mathrm{W}\mathbb{T}}{A \mid B, A \mid B, A}\,\mathrm{W}\mathbb{U}}{A \mid A, B, A \mid B, A}\,\mathrm{W}\mathbb{U} \qquad \dfrac{\dfrac{\dfrac{A \mid A \mid A}{A \mid A \mid A, A}\,\mathrm{W}\mathbb{T}}{A \mid A \mid B, A, A}\,\mathrm{W}\mathbb{T}}{B, A \mid A \mid B, A, A}\,\mathrm{W}\mathbb{F}}{A \mid A \to B, A \mid B, A}\,{\to}\mathbb{U} \qquad \dfrac{\dfrac{\dfrac{B \mid B \mid B}{B \mid A, B \mid B}\,\mathrm{W}\mathbb{U}}{B \mid B, A \mid B}\,\mathrm{X}\mathbb{U}}{B \mid A, B, A \mid B}\,\mathrm{W}\mathbb{U} \qquad \dfrac{\dfrac{\dfrac{A \mid A \mid A}{A \mid A \mid B, A}\,\mathrm{W}\mathbb{T}}{B, A \mid A \mid B, A}\,\mathrm{W}\mathbb{F}}{A, B \mid A, B, A \mid B}\,\mathrm{X}\mathbb{F} \qquad \dfrac{\dfrac{\dfrac{A \mid A \mid A}{A \mid A \mid B, A}\,\mathrm{W}\mathbb{T}}{B, A \mid A \mid B, A}\,\mathrm{W}\mathbb{F}}{B, B, A \mid A \mid B, A}\,\mathrm{W}\mathbb{F}}{B, A \mid A \to B, A \mid B}\,{\to}\mathbb{U}}{A \to B, A \mid A \to B, A \mid B}\,{\to}\mathbb{F}$$

Figure 48.1: Example derivation in $\mathbf{L}_3$

Part XI

Normal Modal Logics

This part covers the metatheory of normal modal logics. It currently consists of Aldo Antonelli's notes on classical correspondence theory for basic modal logic.

Chapter 49

Syntax and Semantics

content/normal-modal-logic/syntax-and-semantics/introduction.tex

49.1 Introduction

nml:syn:int:sec

Modal logic deals with *modal propositions* and the entailment relations among them. Examples of modal propositions are the following:

1. It is necessary that $2 + 2 = 4$.
2. It is necessarily possible that it will rain tomorrow.
3. If it is necessarily possible that φ then it is possible that φ.

Possibility and necessity are not the only modalities: other unary connectives are also classified as modalities, for instance, "it ought to be the case that φ," "It will be the case that φ," "Dana knows that φ," or "Dana believes that φ."

Modal logic makes its first appearance in Aristotle's *De Interpretatione*: he was the first to notice that necessity implies possibility, but not vice versa; that possibility and necessity are inter-definable; that If $\varphi \wedge \psi$ is possibly true then φ is possibly true and ψ is possibly true, but not conversely; and that if $\varphi \to \psi$ is necessary, then if φ is necessary, so is ψ.

The first modern approach to modal logic was the work of C. I. Lewis, culminating with Lewis and Langford, *Symbolic Logic* (1932). Lewis & Langford were unhappy with the representation of implication by means of the material conditional: $\varphi \to \psi$ is a poor substitute for "φ implies ψ." Instead, they proposed to characterize implication as "Necessarily, if φ then ψ," symbolized as $\varphi \dashv\!\!\exists\, \psi$. In trying to sort out the different properties, Lewis identified five different modal systems, **S1**, ..., **S4**, **S5**, the last two of which are still in use.

The approach of Lewis and Langford was purely *syntactical*: they identified reasonable axioms and rules and investigated what was provable with those means. A semantic approach remained elusive for a long time, until a first attempt was made by Rudolf Carnap in *Meaning and Necessity* (1947) using the notion of a *state description*, i.e., a collection of atomic sentences (those that are "true" in that state description). After lifting the truth definition to arbitrary sentences φ, Carnap defines φ to be *necessarily true* if it is true in all state descriptions. Carnap's approach could not handle *iterated* modalities, in that sentences of the form "Possibly necessarily ... possibly φ" always reduce to the innermost modality.

The major breakthrough in modal semantics came with Saul Kripke's article "A Completeness Theorem in Modal Logic" (JSL 1959). Kripke based his work on Leibniz's idea that a statement is necessarily true if it is true "at all possible worlds." This idea, though, suffers from the same drawbacks as Carnap's, in that the truth of statement at a world w (or a state description s) does not depend on w at all. So Kripke assumed that worlds are related by an *accessibility relation* R, and that a statement of the form "Necessarily φ" is true at a world w if and only if φ is true at all worlds w' *accessible from* w. Semantics that provide some version of this approach are called Kripke semantics and made possible the tumultuous development of modal logics (in the plural).

When interpreted by the Kripke semantics, modal logic shows us what *relational structures* look like "from the inside." A relational structure is just a set equipped with a binary relation (for instance, the set of students in the class ordered by their social security number is a relational structure). But in fact relational structures come in all sorts of domains: besides relative possibility of states of the world, we can have epistemic states of some agent related by epistemic possibility, or states of a dynamical system with their state transitions, etc. Modal logic can be used to model all of these: the first gives us ordinary, alethic, modal logic; the others give us epistemic logic, dynamic logic, etc.

We focus on one particular angle, known to modal logicians as "correspondence theory." One of the most significant early discoveries of Kripke's is that many properties of the accessibility relation R (whether it is transitive, symmetric, etc.) can be characterized *in the modal language* itself by means of appropriate "modal schemas." Modal logicians say, for instance, that the reflexivity of R "corresponds" to the schema "If necessarily φ, then φ". We explore mainly the correspondence theory of a number of classical systems of modal logic (e.g., **S4** and **S5**) obtained by a combination of the schemas D, T,

B, 4, and 5.

---

content/normal-modal-logic/syntax-and-semantics/language-modal-logic.tex

49.2 The Language of Basic Modal Logic

nml:syn:lan:sec

Definition 49.1. The basic language of modal logic contains

1. The propositional constant for falsity $\bot$.
2. The propositional constant for truth $\top$.
3. A denumerable set of propositional variables: p_0, p_1, p_2, ...
4. The propositional connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\to$ (conditional), $\leftrightarrow$ (biconditional).
5. The modal operator $\Box$.
6. The modal operator $\Diamond$.

Definition 49.2. *Formulas* of the basic modal language are inductively defined as follows:

1. $\bot$ is an atomic formula.
2. $\top$ is an atomic formula.
3. Every propositional variable p_i is an (atomic) formula.
4. If φ is a formula, then $\neg\varphi$ is a formula.
5. If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.
6. If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.
7. If φ and ψ are formulas, then $(\varphi \to \psi)$ is a formula.
8. If φ and ψ are formulas, then $(\varphi \leftrightarrow \psi)$ is a formula.
9. If φ is a formula, then $\Box\varphi$ is a formula.
10. If φ is a formula, then $\Diamond\varphi$ is a formula.
11. Nothing else is a formula.

If a formula φ does not contain $\Box$ or $\Diamond$, we say it is *modal-free*.

---

content/normal-modal-logic/syntax-and-semantics/substitution.tex

49.3 Simultaneous Substitution

nml:syn:sub:sec

An *instance* of a formula φ is the result of replacing all occurrences of a propositional variable in φ by some other formula. We will refer to instances of formulas often, both when discussing validity and when discussing derivability. It therefore is useful to define the notion precisely.

nml:syn:sub:def:subst-inst

Definition 49.3. Where φ is a modal formula all of whose propositional variables are among $p_1, \ldots, p_n$, and $\theta_1, \ldots, \theta_n$ are also modal formulas, we define $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ as the result of simultaneously substituting each θ_i for p_i in φ. Formally, this is a definition by induction on φ:

1. $\varphi \equiv \bot$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is $\bot$.
2. $\varphi \equiv \top$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is $\top$.
3. $\varphi \equiv q$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is q, provided $q \not\equiv p_i$ for $i = 1, \ldots, n$.
4. $\varphi \equiv p_i$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is θ_i.
5. $\varphi \equiv \neg\psi$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is $\neg\psi[\theta_1/p_1, \ldots, \theta_n/p_n]$.
6. $\varphi \equiv (\psi \wedge \chi)$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is
$$(\psi[\theta_1/p_1, \ldots, \theta_n/p_n] \wedge \chi[\theta_1/p_1, \ldots, \theta_n/p_n]).$$
7. $\varphi \equiv (\psi \vee \chi)$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is
$$(\psi[\theta_1/p_1, \ldots, \theta_n/p_n] \vee \chi[\theta_1/p_1, \ldots, \theta_n/p_n]).$$
8. $\varphi \equiv (\psi \to \chi)$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is
$$(\psi[\theta_1/p_1, \ldots, \theta_n/p_n] \to \chi[\theta_1/p_1, \ldots, \theta_n/p_n]).$$
9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is
$$(\psi[\theta_1/p_1, \ldots, \theta_n/p_n] \leftrightarrow \chi[\theta_1/p_1, \ldots, \theta_n/p_n]).$$
10. $\varphi \equiv \Box\psi$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is $\Box\psi[\theta_1/p_1, \ldots, \theta_n/p_n]$.
11. $\varphi \equiv \Diamond\psi$: $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is $\Diamond\psi[\theta_1/p_1, \ldots, \theta_n/p_n]$.

The formula $\varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is called a *substitution instance* of φ.

Example 49.4. Suppose φ is $p_1 \to \Box(p_1 \wedge p_2)$, θ_1 is $\Diamond(p_2 \to p_3)$ and θ_2 is $\neg\Box p_1$. Then $\varphi[\theta_1/p_1, \theta_2/p_2]$ is

$$\Diamond(p_2 \to p_3) \to \Box(\Diamond(p_2 \to p_3) \wedge \neg\Box p_1)$$

while $\varphi[\theta_2/p_1, \theta_1/p_2]$ is

$$\neg\Box p_1 \to \Box(\neg\Box p_1 \wedge \Diamond(p_2 \to p_3))$$

Note that simultaneous substitution is in general not the same as iterated substitution, e.g., compare $\varphi[\theta_1/p_1, \theta_2/p_2]$ above with $(\varphi[\theta_1/p_1])[\theta_2/p_2]$, which is:

$$\begin{aligned}\Diamond(p_2 \to p_3) &\to \Box(\Diamond(p_2 \to p_3) \wedge p_2)[\neg\Box p_1/p_2], \text{ i.e.,}\\ \Diamond(\neg\Box p_1 \to p_3) &\to \Box(\Diamond(\neg\Box p_1 \to p_3) \wedge \neg\Box p_1)\end{aligned}$$

and with $(\varphi[\theta_2/p_2])[\theta_1/p_1]$:

$$\begin{aligned}p_1 &\to \Box(p_1 \wedge \neg\Box p_1)[\Diamond(p_2 \to p_3)/p_1], \text{ i.e.,}\\ \Diamond(p_2 \to p_3) &\to \Box(\Diamond(p_2 \to p_3) \wedge \neg\Box\Diamond(p_2 \to p_3)).\end{aligned}$$

---

content/normal-modal-logic/syntax-and-semantics/relational-models.tex

49.4 Relational Models

nml:syn:rel:sec

The basic concept of semantics for normal modal logics is that of a *relational model*. It consists of a set of worlds, which are related by a binary "accessibility relation," together with an assignment which determines which propositional variables count as "true" at which worlds.

Definition 49.5. A *model* for the basic modal language is a triple $\mathfrak{M} = \langle W, R, V\rangle$, where

1. W is a nonempty set of "worlds,"
2. R is a binary accessibility relation on W, and
3. V is a function assigning to each propositional variable p a set $V(p)$ of possible worlds.

When Rww' holds, we say that w' is *accessible from* w. When $w \in V(p)$ we say p is *true at* w.

The great advantage of relational semantics is that models can be represented by means of simple diagrams, such as the one in Figure 49.1. Worlds are represented by nodes, and world w' is accessible from w precisely when there is an arrow from w to w'. Moreover, we label a node (world) by p when $w \in V(p)$, and otherwise by $\neg p$. Figure 49.1 represents the model with $W = \{w_1, w_2, w_3\}$, $R = \{\langle w_1, w_2\rangle, \langle w_1, w_3\rangle\}$, $V(p) = \{w_1, w_2\}$, and $V(q) = \{w_2\}$.

---

content/normal-modal-logic/syntax-and-semantics/truth-at-w.tex

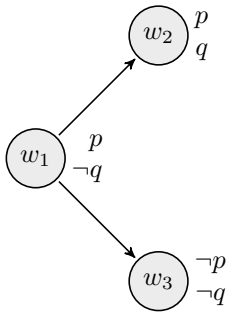


Figure 49.1: A simple model.

nml:syn:rel:fig:simple

49.5 Truth at a World

Every modal model determines which modal formulas count as true at which worlds in it. The relation "model $\mathfrak{M}$ makes formula φ true at world w" is the basic notion of relational semantics. The relation is defined inductively and coincides with the usual characterization using truth tables for the non-modal operators.

nml:syn:trw:sec

Definition 49.6. *Truth of a formula φ at w* in a $\mathfrak{M}$, in symbols: $\mathfrak{M}, w \Vdash \varphi$, is defined inductively as follows:

nml:syn:trw:defn:mmodels

1. $\varphi \equiv \bot$: Never $\mathfrak{M}, w \Vdash \bot$.
2. $\varphi \equiv \top$: Always $\mathfrak{M}, w \Vdash \top$.
3. $\mathfrak{M}, w \Vdash p$ iff $w \in V(p)$.
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$ (or both).
7. $\varphi \equiv (\psi \to \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$ or $\mathfrak{M}, w \Vdash \chi$.
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff either both $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$ or neither $\mathfrak{M}, w \Vdash \psi$ nor $\mathfrak{M}, w \Vdash \chi$.
9. $\varphi \equiv \Box\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w' \Vdash \psi$ for all $w' \in W$ with Rww'. nml:syn:trw:defn:sub:mmodels-box
10. $\varphi \equiv \Diamond\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w' \Vdash \psi$ for at least one $w' \in W$ with Rww'. nml:syn:trw:defn:sub:mmodels-diamond

Note that by clause (9), a formula $\Box\psi$ is true at w whenever there are no w' with Rww'. In such a case $\Box\psi$ is *vacuously* true at w. Also, $\Box\psi$ may be satisfied at w even if ψ is not. The truth of ψ at w does not guarantee the truth of $\Diamond\psi$ at w. This holds, however, if Rww, e.g., if R is reflexive. If there is no w' such that Rww', then $\mathfrak{M}, w \nVdash \Diamond\varphi$, for any φ.

Problem 49.1. Consider the model of Figure 49.1. Which of the following hold?

1. $\mathfrak{M}, w_1 \Vdash q$;
2. $\mathfrak{M}, w_3 \Vdash \neg q$;
3. $\mathfrak{M}, w_1 \Vdash p \vee q$;
4. $\mathfrak{M}, w_1 \Vdash \Box(p \vee q)$;
5. $\mathfrak{M}, w_3 \Vdash \Box q$;
6. $\mathfrak{M}, w_3 \Vdash \Box\bot$;
7. $\mathfrak{M}, w_1 \Vdash \Diamond q$;
8. $\mathfrak{M}, w_1 \Vdash \Box q$;
9. $\mathfrak{M}, w_1 \Vdash \neg\Box\Box\neg q$.

nml:syn:trw: prop:dual

Proposition 49.7.

1. $\mathfrak{M}, w \Vdash \Box\varphi$ *iff* $\mathfrak{M}, w \Vdash \neg\Diamond\neg\varphi$.
2. $\mathfrak{M}, w \Vdash \Diamond\varphi$ *iff* $\mathfrak{M}, w \Vdash \neg\Box\neg\varphi$.

Proof. 1. $\mathfrak{M}, w \Vdash \neg\Diamond\neg\varphi$ iff $\mathfrak{M}, w \nVdash \Diamond\neg\varphi$ by definition of $\mathfrak{M}, w \Vdash$. $\mathfrak{M}, w \Vdash \Diamond\neg\varphi$ iff for some w' with Rww', $\mathfrak{M}, w' \Vdash \neg\varphi$. Hence, $\mathfrak{M}, w \nVdash \Diamond\neg\varphi$ iff for all w' with Rww', $\mathfrak{M}, w' \nVdash \neg\varphi$. We also have $\mathfrak{M}, w' \nVdash \neg\varphi$ iff $\mathfrak{M}, w' \Vdash \varphi$. Together we have $\mathfrak{M}, w \Vdash \neg\Diamond\neg\varphi$ iff for all w' with Rww', $\mathfrak{M}, w' \Vdash \varphi$. Again by definition of $\mathfrak{M}, w \Vdash$, that is the case iff $\mathfrak{M}, w \Vdash \Box\varphi$.

2. $\mathfrak{M}, w \Vdash \neg\Box\neg\varphi$ iff $\mathfrak{M} \nVdash \Box\neg\varphi$. $\mathfrak{M}, w \Vdash \Box\neg\varphi$ iff for all w' with Rww', $\mathfrak{M}, w' \Vdash \neg\varphi$. Hence, $\mathfrak{M}, w \nVdash \Box\neg\varphi$ iff for some w' with Rww', $\mathfrak{M}, w' \nVdash \neg\varphi$. We also have $\mathfrak{M}, w' \nVdash \neg\varphi$ iff $\mathfrak{M}, w' \Vdash \varphi$. Together we have $\mathfrak{M}, w \Vdash \neg\Box\neg\varphi$ iff for some w' with Rww', $\mathfrak{M}, w' \Vdash \varphi$. Again by definition of $\mathfrak{M}, w \Vdash$, that is the case iff $\mathfrak{M}, w \Vdash \Diamond\varphi$. □

Problem 49.2. Complete the proof of Proposition 49.7.

Problem 49.3. Let $\mathfrak{M} = \langle W, R, V\rangle$ be a model, and suppose $w_1, w_2 \in W$ are such that:

1. $w_1 \in V(p)$ if and only if $w_2 \in V(p)$ (for every propositional variable p); and
2. for all $w \in W$: Rw_1w if and only if Rw_2w.

Using induction on formulas, show that for all formulas φ: $\mathfrak{M}, w_1 \Vdash \varphi$ if and only if $\mathfrak{M}, w_2 \Vdash \varphi$.

Problem 49.4. Let $\mathfrak{M} = \langle W, R, V\rangle$. Show that $\mathfrak{M}, w \Vdash \neg\Diamond\varphi$ if and only if $\mathfrak{M}, w \Vdash \Box\neg\varphi$.

---

content/normal-modal-logic/syntax-and-semantics/truth-in-model.tex

49.6 Truth in a Model

Sometimes we are interested in which formulas are true at every world in a given model. Let's introduce a notation for this. nml:syn:tru:sec

Definition 49.8. A formula φ is *true in a model* $M = \langle W, R, V\rangle$, written $\mathfrak{M} \Vdash \varphi$, if and only if $\mathfrak{M}, w \Vdash \varphi$ for every $w \in W$.

Proposition 49.9. nml:syn:tru:prop:truthfacts

1. *If* $\mathfrak{M} \Vdash \varphi$ *then* $\mathfrak{M} \nVdash \neg\varphi$, *but* not *vice-versa.*
2. *If* $\mathfrak{M} \Vdash \varphi \to \psi$ *then* $\mathfrak{M} \Vdash \varphi$ *only if* $\mathfrak{M} \Vdash \psi$, *but* not *vice-versa.*

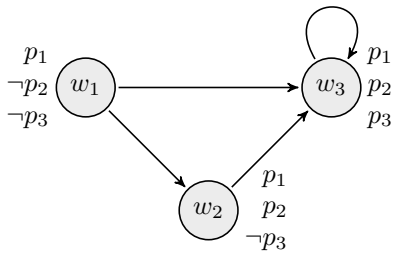
Proof. 1. If $\mathfrak{M} \Vdash \varphi$ then φ is true at all worlds in W, and since $W \neq \emptyset$, it can't be that $\mathfrak{M} \Vdash \neg\varphi$, or else φ would have to be both true and false at some world.

On the other hand, if $\mathfrak{M} \nVdash \neg\varphi$ then φ is true at some world $w \in W$. It does not follow that $\mathfrak{M}, w \Vdash \varphi$ for *every* $w \in W$. For instance, in the model of Figure 49.1, $\mathfrak{M} \nVdash \neg p$, and also $\mathfrak{M} \nVdash p$.

2. Assume $\mathfrak{M} \Vdash \varphi \to \psi$ and $\mathfrak{M} \Vdash \varphi$; to show $\mathfrak{M} \Vdash \psi$ let $w \in W$ be an arbitrary world. Then $\mathfrak{M}, w \Vdash \varphi \to \psi$ and $\mathfrak{M}, w \Vdash \varphi$, so $\mathfrak{M}, w \Vdash \psi$, and since w was arbitrary, $\mathfrak{M} \Vdash \psi$.

   To show that the converse fails, we need to find a model $\mathfrak{M}$ such that $\mathfrak{M} \Vdash \varphi$ only if $\mathfrak{M} \Vdash \psi$, but $\mathfrak{M} \nVdash \varphi \to \psi$. Consider again the model of Figure 49.1: $\mathfrak{M} \nVdash p$ and hence (vacuously) $\mathfrak{M} \Vdash p$ only if $\mathfrak{M} \Vdash q$. However, $\mathfrak{M} \nVdash p \to q$, as p is true but q false at w_1. □

Problem 49.5. Consider the following model $\mathfrak{M}$ for the language comprising p_1, p_2, p_3 as the only propositional variables:



Are the following formulas and schemas true in the model $\mathfrak{M}$, i.e., true at every world in $\mathfrak{M}$? Explain.

1. $p \to \Diamond p$ (for p atomic);
2. $\varphi \to \Diamond\varphi$ (for φ arbitrary);
3. $\Box p \to p$ (for p atomic);
4. $\neg p \to \Diamond\Box p$ (for p atomic);
5. $\Diamond\Box\varphi$ (for φ arbitrary);
6. $\Box\Diamond p$ (for p atomic).

---

content/normal-modal-logic/syntax-and-semantics/modal-validity.tex

49.7 Validity

nml:syn:val:sec

Formulas that are true in all models, i.e., true at every world in every model, are particularly interesting. They represent those modal propositions which are true regardless of how $\Box$ and $\Diamond$ are interpreted, as long as the interpretation is "normal" in the sense that it is generated by some accessibility relation on possible worlds. We call such formulas *valid*. For instance, $\Box(p \wedge q) \to \Box p$ is valid. Some formulas one might expect to be valid on the basis of the alethic interpretation of $\Box$, such as $\Box p \to p$, are not valid, however. Part of the interest of relational models is that different interpretations of $\Box$ and $\Diamond$ can be captured by different kinds of accessibility relations. This suggests that we should define validity not just relative to *all* models, but relative to all models *of a certain kind*. It will turn out, e.g., that $\Box p \to p$ is true in all models where every world is accessible from itself, i.e., R is reflexive. Defining validity relative to classes of models enables us to formulate this succinctly: $\Box p \to p$ is valid in the class of reflexive models.

explanation

Definition 49.10. A formula φ is *valid* in a class $\mathcal{C}$ of models if it is true in every model in $\mathcal{C}$ (i.e., true at every world in every model in $\mathcal{C}$). If φ is valid in $\mathcal{C}$, we write $\mathcal{C} \vDash \varphi$, and we write $\vDash \varphi$ if φ is valid in the class of *all* models.

nml:syn:val:prop:subset-class

Proposition 49.11. *If φ is valid in $\mathcal{C}$ it is also valid in each class $\mathcal{C}' \subseteq \mathcal{C}$.*

nml:syn:val:prop:Nec-rule

Proposition 49.12. *If φ is valid, then so is $\Box\varphi$.*

Proof. Assume $\vDash \varphi$. To show $\vDash \Box\varphi$ let $\mathfrak{M} = \langle W, R, V\rangle$ be a model and $w \in W$. If Rww' then $\mathfrak{M}, w' \Vdash \varphi$, since φ is valid, and so also $\mathfrak{M}, w \Vdash \Box\varphi$. Since $\mathfrak{M}$ and w were arbitrary, $\vDash \Box\varphi$. □

Problem 49.6. Show that the following are valid:

1. $\Box p \to \Box(q \to p)$;
2. $\Box\neg\bot$;
3. $\Box p \to (\Box q \to \Box p)$.

Problem 49.7. Show that $\varphi \to \Box\varphi$ is valid in the class $\mathcal{C}$ of models $\mathfrak{M} = \langle W, R, V\rangle$ where $W = \{w\}$. Similarly, show that $\psi \to \Box\varphi$ and $\Diamond\varphi \to \psi$ are valid in the class of models $\mathfrak{M} = \langle W, R, V\rangle$ where $R = \emptyset$.

---

content/normal-modal-logic/syntax-and-semantics/tautological-instances.tex

49.8 Tautological Instances

explanation

A modal-free formula is a tautology if it is true under every truth-value assignment. Clearly, every tautology is true at every world in every model. But for formulas involving $\Box$ and $\Diamond$, the notion of tautology is not defined. Is it the case, e.g., that $\Box p \vee \neg\Box p$—an instance of the principle of excluded middle—is valid? The notion of a *tautological instance* helps: a formula that is a substitution instance of a (non-modal) tautology. It is not surprising, but still requires proof, that every tautological instance is valid.

nml:syn:tau:sec

Definition 49.13. A modal formula ψ is a *tautological instance* if and only if there is a modal-free tautology φ with propositional variables $p_1, \ldots, p_n$ and formulas $\theta_1, \ldots, \theta_n$ such that $\psi \equiv \varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$.

Lemma 49.14. *Suppose φ is a modal-free formula whose propositional variables are $p_1, \ldots, p_n$, and let $\theta_1, \ldots, \theta_n$ be modal formulas. Then for any assignment $\mathfrak{v}$, any model $\mathfrak{M} = \langle W, R, V\rangle$, and any $w \in W$ such that $\mathfrak{v}(p_i) = \mathbb{T}$ if and only if $\mathfrak{M}, w \Vdash \theta_i$ we have that $\mathfrak{v} \vDash \varphi$ if and only if $\mathfrak{M}, w \Vdash \varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$.*

nml:syn:tau:lem:valid-taut

Proof. By induction on φ.

1. $\varphi \equiv \bot$: Both $\mathfrak{v} \nvDash \bot$ and $\mathfrak{M}, w \nVdash \bot$.
2. $\varphi \equiv \top$: Both $\mathfrak{v} \vDash \top$ and $\mathfrak{M}, w \Vdash \top$.
3. $\varphi \equiv p_i$:

$$\begin{aligned}
\mathfrak{v} \vDash p_i &\Leftrightarrow \mathfrak{v}(p_i) = \mathbb{T} \\
&\qquad \text{by definition of } \mathfrak{v} \vDash p_i \\
&\Leftrightarrow \mathfrak{M}, w \Vdash \theta_i \\
&\qquad \text{by assumption} \\
&\Leftrightarrow \mathfrak{M}, w \Vdash p_i[\theta_1/p_1, \ldots, \theta_n/p_n] \\
&\qquad \text{since } p_i[\theta_1/p_1, \ldots, \theta_n/p_n] \equiv \theta_i.
\end{aligned}$$

4. $\varphi \equiv \neg\psi$:

$$\begin{aligned}
\mathfrak{v} \vDash \neg\psi &\Leftrightarrow \mathfrak{v} \nvDash \psi \\
&\qquad \text{by definition of } \mathfrak{v} \vDash; \\
&\Leftrightarrow \mathfrak{M}, w \nVdash \psi[\theta_1/p_1, \ldots, \theta_n/p_n] \\
&\qquad \text{by induction hypothesis} \\
&\Leftrightarrow \mathfrak{M}, w \Vdash \neg\psi[\theta_1/p_1, \ldots, \theta_n/p_n] \\
&\qquad \text{by definition of } \mathfrak{v} \vDash.
\end{aligned}$$

5. $\varphi \equiv (\psi \wedge \chi)$:

$$\begin{aligned}
\mathfrak{v} \vDash \psi \wedge \chi &\Leftrightarrow \mathfrak{v} \vDash \psi \text{ and } \mathfrak{v} \vDash \chi \\
&\qquad \text{by definition of } \mathfrak{v} \vDash \\
&\Leftrightarrow \mathfrak{M}, w \Vdash \psi[\theta_1/p_1, \ldots, \theta_n/p_n] \text{ and} \\
&\quad \mathfrak{M}, w \Vdash \chi[\theta_1/p_1, \ldots, \theta_n/p_n] \\
&\qquad \text{by induction hypothesis} \\
&\Leftrightarrow \mathfrak{M}, w \Vdash (\psi \wedge \chi)[\theta_1/p_1, \ldots, \theta_n/p_n] \\
&\qquad \text{by definition of } \mathfrak{M}, w \Vdash.
\end{aligned}$$

6. $\varphi \equiv (\psi \vee \chi)$:

$$\begin{aligned}
\mathfrak{v} \vDash \psi \vee \chi &\Leftrightarrow \mathfrak{v} \vDash \psi \text{ or } \mathfrak{v} \vDash \chi \\
&\qquad \text{by definition of } \mathfrak{v} \vDash; \\
&\Leftrightarrow \mathfrak{M}, w \Vdash \psi[\theta_1/p_1, \ldots, \theta_n/p_n] \text{ or} \\
&\quad \mathfrak{M}, w \Vdash \chi[\theta_1/p_1, \ldots, \theta_n/p_n] \\
&\qquad \text{by induction hypothesis} \\
&\Leftrightarrow \mathfrak{M}, w \Vdash (\psi \vee \chi)[\theta_1/p_1, \ldots, \theta_n/p_n] \\
&\qquad \text{by definition of } \mathfrak{M}, w \Vdash.
\end{aligned}$$

7. $\varphi \equiv (\psi \to \chi)$:

$$\begin{aligned}
\mathfrak{v} \vDash \psi \to \chi &\Leftrightarrow \mathfrak{v} \nvDash \psi \text{ or } \mathfrak{v} \vDash \chi \\
&\qquad \text{by definition of } \mathfrak{v} \vDash \\
&\Leftrightarrow \mathfrak{M}, w \nVdash \psi[\theta_1/p_1, \ldots, \theta_n/p_n] \text{ or} \\
&\quad \mathfrak{M}, w \Vdash \chi[\theta_1/p_1, \ldots, \theta_n/p_n] \\
&\qquad \text{by induction hypothesis} \\
&\Leftrightarrow \mathfrak{M}, w \Vdash (\psi \to \chi)[\theta_1/p_1, \ldots, \theta_n/p_n] \\
&\qquad \text{by definition of } \mathfrak{M}, w \Vdash.
\end{aligned}$$

8. $\varphi \equiv (\psi \leftrightarrow \chi)$:

$$\begin{aligned}
\mathfrak{v} \vDash \psi \to \chi \Leftrightarrow\ & \text{either } \mathfrak{v} \vDash \psi \text{ and } \mathfrak{v} \vDash \chi \\
& \text{or } \mathfrak{v} \nvDash \psi \text{ and } \mathfrak{v} \nvDash \chi \\
& \text{by definition of } \mathfrak{v} \vDash \\
\Leftrightarrow\ & \text{either } \mathfrak{M}, w \Vdash \psi[\theta_1/p_1, \ldots, \theta_n/p_n] \text{ and} \\
& \mathfrak{M}, w \Vdash \chi[\theta_1/p_1, \ldots, \theta_n/p_n] \\
& \text{or } \mathfrak{M}, w \nVdash \psi[\theta_1/p_1, \ldots, \theta_n/p_n] \text{ and} \\
& \mathfrak{M}, w \nVdash \chi[\theta_1/p_1, \ldots, \theta_n/p_n] \\
& \text{by induction hypothesis} \\
\Leftrightarrow\ & \mathfrak{M}, w \Vdash (\psi \leftrightarrow \chi)[\theta_1/p_1, \ldots, \theta_n/p_n] \\
& \text{by definition of } \mathfrak{M}, w \Vdash.
\end{aligned}$$

□

Proposition 49.15. *All tautological instances are valid.*

nml:syn:tau:prop:valid-taut

Proof. Contrapositively, suppose φ is such that $\mathfrak{M}, w \nVdash \varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$, for some model $\mathfrak{M}$ and world w. Define an assignment $\mathfrak{v}$ such that $\mathfrak{v}(p_i) = \mathbb{T}$ if and only if $\mathfrak{M}, w \Vdash \theta_i$ (and $\mathfrak{v}$ assigns arbitrary values to $q \notin \{p_1, \ldots, p_n\}$). Then by Lemma 49.14, $\mathfrak{v} \nvDash \varphi$, so φ is not a tautology. □

---

content/normal-modal-logic/syntax-and-semantics/schemas.tex

49.9 Schemas and Validity

nml:syn:sch:sec

Definition 49.16. A *schema* is a set of formulas comprising all and only the substitution instances of some modal formula χ, i.e.,

$$\{\psi : \exists\theta_1, \ldots, \exists\theta_n\, (\psi = \chi[\theta_1/p_1, \ldots, \theta_n/p_n])\}.$$

The formula χ is called the *characteristic* formula of the schema, and it is unique up to a renaming of the propositional variables. A formula φ is an *instance* of a schema if it is a member of the set.

It is convenient to denote a schema by the meta-linguistic expression obtained by substituting 'φ', 'ψ', ..., for the atomic components of χ. So, for instance, the following denote schemas: 'φ', '$\varphi \to \Box\varphi$', '$\varphi \to (\psi \to \varphi)$'. They correspond to the characteristic formulas p, $p \to \Box p$, $p \to (q \to p)$. The schema 'φ' denotes the set of *all* formulas.

Definition 49.17. A schema is *true* in a model if and only if all of its instances are; and a schema is *valid* if and only if it is true in every model.

nml:syn:sch: prop:Kvalid

Proposition 49.18. *The following schema* K *is valid*

$$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi). \tag{K}$$

Proof. We need to show that all instances of the schema are true at every world in every model. So let $\mathfrak{M} = \langle W, R, V\rangle$ and $w \in W$ be arbitrary. To show that a conditional is true at a world we assume the antecedent is true to show that the consequent is true as well. In this case, let $\mathfrak{M}, w \Vdash \Box(\varphi \to \psi)$ and $\mathfrak{M}, w \Vdash \Box\varphi$. We need to show $\mathfrak{M}, w \Vdash \Box\psi$. So let w' be arbitrary such that Rww'. Then by the first assumption $\mathfrak{M}, w' \Vdash \varphi \to \psi$ and by the second assumption $\mathfrak{M}, w' \Vdash \varphi$. It follows that $\mathfrak{M}, w' \Vdash \psi$. Since w' was arbitrary, $\mathfrak{M}, w \Vdash \Box\psi$. □

nml:syn:sch: prop:Dual-valid

Proposition 49.19. *The following schema* DUAL *is valid*

$$\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi. \tag{DUAL}$$

Proof. Exercise. □

Problem 49.8. Prove Proposition 49.19.

nml:syn:sch: prop:soundMP

Proposition 49.20. *If φ and $\varphi \to \psi$ are true at a world in a model then so is ψ. Hence, the valid formulas are closed under modus ponens.*

nml:syn:sch: prop:valid-instances

Proposition 49.21. *A formula φ is valid iff all its substitution instances are. In other words, a schema is valid iff its characteristic formula is.*

Proof. The "if" direction is obvious, since φ is a substitution instance of itself.

To prove the "only if" direction, we show the following: Suppose $\mathfrak{M} = \langle W, R, V\rangle$ is a modal model, and $\psi \equiv \varphi[\theta_1/p_1, \ldots, \theta_n/p_n]$ is a substitution instance of φ. Define $\mathfrak{M}' = \langle W, R, V'\rangle$ by $V'(p_i) = \{w : \mathfrak{M}, w \Vdash \theta_i\}$. Then $\mathfrak{M}, w \Vdash \psi$ iff $\mathfrak{M}', w \Vdash \varphi$, for any $w \in W$. (We leave the proof as an exercise.) Now suppose that φ was valid, but some substitution instance ψ of φ was not valid. Then for some $\mathfrak{M} = \langle W, R, V\rangle$ and some $w \in W$, $\mathfrak{M}, w \nVdash \psi$. But then $\mathfrak{M}', w \nVdash \varphi$ by the claim, and φ is not valid, a contradiction. □

Problem 49.9. Prove the claim in the "only if" part of the proof of Proposition 49.21. (Hint: use induction on φ.)

Note, however, that it is not true that a schema is true in a model iff its characteristic formula is. Of course, the "only if" direction holds: if every instance of φ is true in $\mathfrak{M}$, φ itself is true in $\mathfrak{M}$. But it may happen that φ is true in $\mathfrak{M}$ but some instance of φ is false at some world in $\mathfrak{M}$. For a very simple counterexample consider p in a model with only one world w and $V(p) = \{w\}$, so that p is true at w. But $\bot$ is an instance of p, and not true at w.

Problem 49.10. Show that none of the following formulas are valid:

D: $\Box p \to \Diamond p$;

Valid Schemas	*Invalid Schemas*
$\Box(\varphi \to \psi) \to (\Diamond\varphi \to \Diamond\psi)$	$\Box(\varphi \vee \psi) \to (\Box\varphi \vee \Box\psi)$
$\Diamond(\varphi \to \psi) \to (\Box\varphi \to \Diamond\psi)$	$(\Diamond\varphi \wedge \Diamond\psi) \to \Diamond(\varphi \wedge \psi)$
$\Box(\varphi \wedge \psi) \leftrightarrow (\Box\varphi \wedge \Box\psi)$	$\varphi \to \Box\varphi$
$\Box\varphi \to \Box(\psi \to \varphi)$	$\Box\Diamond\varphi \to \psi$
$\neg\Diamond\varphi \to \Box(\varphi \to \psi)$	$\Box\Box\varphi \to \Box\varphi$
$\Diamond(\varphi \vee \psi) \leftrightarrow (\Diamond\varphi \vee \Diamond\psi)$	$\Box\Diamond\varphi \to \Diamond\Box\varphi$.

Table 49.1: Valid and (or?) invalid schemas.

nml:syn:sch:
tab:valid-invalidSchemas

T: $\Box p \to p$;

B: $p \to \Box\Diamond p$;

4: $\Box p \to \Box\Box p$;

5: $\Diamond p \to \Box\Diamond p$.

Problem 49.11. Prove that the schemas in the first column of Table 49.1 are valid and those in the second column are not valid.

Problem 49.12. Decide whether the following schemas are valid or invalid:

1. $(\Diamond\varphi \to \Box\psi) \to (\Box\varphi \to \Box\psi)$;
2. $\Diamond(\varphi \to \psi) \vee \Box(\psi \to \varphi)$.

Problem 49.13. For each of the following schemas find a model $\mathfrak{M}$ such that every instance of the formula is true in $\mathfrak{M}$:

1. $p \to \Diamond\Diamond p$;
2. $\Diamond p \to \Box p$.

---

content/normal-modal-logic/syntax-and-semantics/entailment.tex

49.10 Entailment

explanation

With the definition of truth at a world, we can define an entailment relation between formulas. A formula ψ entails φ iff, whenever ψ is true, φ is true as well. Here, "whenever" means both "whichever model we consider" as well as "whichever world in that model we consider."

nml:syn:ent:
sec

Definition 49.22. If Γ is a set of formulas and φ a formula, then Γ *entails* φ, in symbols: $\Gamma \vDash \varphi$, if and only if for every model $\mathfrak{M} = \langle W, R, V\rangle$ and world $w \in W$, if $\mathfrak{M}, w \Vdash \psi$ for every $\psi \in \Gamma$, then $\mathfrak{M}, w \Vdash \varphi$. If Γ contains a single formula ψ, then we write $\psi \vDash \varphi$.

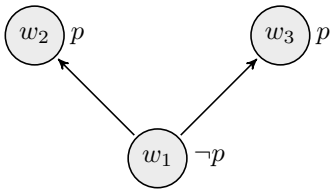


Figure 49.2: Counterexample to $p \to \Diamond p \vDash \Box p \to p$.

nml:syn:ent:
fig:counterex

Example 49.23. To show that a formula entails another, we have to reason about all models, using the definition of $\mathfrak{M}, w \Vdash$. For instance, to show $p \to \Diamond p \vDash \Box\neg p \to \neg p$, we might argue as follows: Consider a model $\mathfrak{M} = \langle W, R, V\rangle$ and $w \in W$, and suppose $\mathfrak{M}, w \Vdash p \to \Diamond p$. We have to show that $\mathfrak{M}, w \Vdash \Box\neg p \to \neg p$. Suppose not. Then $\mathfrak{M}, w \Vdash \Box\neg p$ and $\mathfrak{M}, w \nVdash \neg p$. Since $\mathfrak{M}, w \nVdash \neg p$, $\mathfrak{M}, w \Vdash p$. By assumption, $\mathfrak{M}, w \Vdash p \to \Diamond p$, hence $\mathfrak{M}, w \Vdash \Diamond p$. By definition of $\mathfrak{M}, w \Vdash \Diamond p$, there is some w' with Rww' such that $\mathfrak{M}, w' \Vdash p$. Since also $\mathfrak{M}, w \Vdash \Box\neg p$, $\mathfrak{M}, w' \Vdash \neg p$, a contradiction.

To show that a formula ψ does not entail another φ, we have to give a counterexample, i.e., a model $\mathfrak{M} = \langle W, R, V\rangle$ where we show that at some world $w \in W$, $\mathfrak{M}, w \Vdash \psi$ but $\mathfrak{M}, w \nVdash \varphi$. Let's show that $p \to \Diamond p \nvDash \Box p \to p$. Consider the model in Figure 49.2. We have $\mathfrak{M}, w_1 \Vdash \Diamond p$ and hence $\mathfrak{M}, w_1 \Vdash p \to \Diamond p$. However, since $\mathfrak{M}, w_1 \Vdash \Box p$ but $\mathfrak{M}, w_1 \nVdash p$, we have $\mathfrak{M}, w_1 \nVdash \Box p \to p$.

Often very simple counterexamples suffice. The model $\mathfrak{M}' = \{W', R', V'\}$ with $W' = \{w\}$, $R' = \emptyset$, and $V'(p) = \emptyset$ is also a counterexample: Since $\mathfrak{M}', w \nVdash p$, $\mathfrak{M}', w \Vdash p \to \Diamond p$. As no worlds are accessible from w, we have $\mathfrak{M}', w \Vdash \Box p$, and so $\mathfrak{M}', w \nVdash \Box p \to p$.

Problem 49.14. Show that $\Box(\varphi \wedge \psi) \vDash \Box\varphi$.

Problem 49.15. Show that $\Box(p \to q) \nvDash p \to \Box q$ and $p \to \Box q \nvDash \Box(p \to q)$.

Chapter 50

Frame Definability

content/normal-modal-logic/frame-definability/introduction.tex

50.1 Introduction

nml:frd:int:sec

One question that interests modal logicians is the relationship between the accessibility relation and the truth of certain formulas in models with that accessibility relation. For instance, suppose the accessibility relation is reflexive, i.e., for every $w \in W$, Rww. In other words, every world is accessible from itself. That means that when $\Box\varphi$ is true at a world w, w itself is among the accessible worlds at which φ must therefore be true. So, if the accessibility relation R of $\mathfrak{M}$ is reflexive, then whatever world w and formula φ we take, $\Box\varphi \to \varphi$ will be true there (in other words, the schema $\Box p \to p$ and all its substitution instances are true in $\mathfrak{M}$).

The converse, however, is false. It's not the case, e.g., that if $\Box p \to p$ is true in $\mathfrak{M}$, then R is reflexive. For we can easily find a non-reflexive model $\mathfrak{M}$ where $\Box p \to p$ is true at all worlds: take the model with a single world w, not accessible from itself, but with $w \in V(p)$. By picking the truth value of p suitably, we can make $\Box\varphi \to \varphi$ true in a model that is not reflexive.

The solution is to remove the variable assignment V from the equation. If we require that $\Box p \to p$ is true at all worlds in $\mathfrak{M}$, regardless of which worlds are in $V(p)$, then it is necessary that R is reflexive. For in any non-reflexive model, there will be at least one world w such that not Rww. If we set $V(p) = W \setminus \{w\}$, then p will be true at all worlds other than w, and so at all worlds accessible from w (since w is guaranteed not to be accessible from w, and w is the only world where p is false). On the other hand, p is false at w, so $\Box p \to p$ is false at w.

This suggests that we should introduce a notation for model structures without a valuation: we call these *frames*. A frame $\mathfrak{F}$ is simply a pair $\langle W, R\rangle$ consisting of a set of worlds with an accessibility relation. Every model $\langle W, R, V\rangle$ is then, as we say, *based on* the frame $\langle W, R\rangle$. Conversely, a frame determines the class of models based on it; and a class of frames determines the class of models which are based on any frame in the class. And we can define $\mathfrak{F} \vDash \varphi$, the notion of a formula being *valid* in a frame as: $\mathfrak{M} \Vdash \varphi$ for all $\mathfrak{M}$ based on $\mathfrak{F}$.

With this notation, we can establish correspondence relations between formulas and classes of frames: e.g., $\mathfrak{F} \vDash \Box p \to p$ if, and only if, $\mathfrak{F}$ is reflexive.

---

content/normal-modal-logic/frame-definability/properties-accessibility.tex

50.2 Properties of Accessibility Relations

nml:frd:acc:sec

Many modal formulas turn out to be characteristic of simple, and even familiar, properties of the accessibility relation. In one direction, that means that any model that has a given property makes a corresponding formula (and all its substitution instances) true. We begin with five classical examples of kinds of accessibility relations and the formulas the truth of which they guarantee.

If R is ...	*then ... is true in $\mathfrak{M}$:*	
serial: $\forall u \exists v Ruv$	$\Box p \to \Diamond p$	(D)
reflexive: $\forall w Rww$	$\Box p \to p$	(T)
symmetric: $\forall u \forall v (Ruv \to Rvu)$	$p \to \Box\Diamond p$	(B)
transitive: $\forall u \forall v \forall w ((Ruv \wedge Rvw) \to Ruw)$	$\Box p \to \Box\Box p$	(4)
euclidean: $\forall w \forall u \forall v ((Rwu \wedge Rwv) \to Ruv)$	$\Diamond p \to \Box\Diamond p$	(5)

Table 50.1: Five correspondence facts.

nml:frd:acc:
tab:five

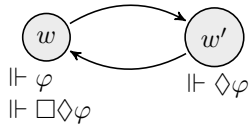


Figure 50.1: The argument from symmetry.

nml:frd:acc:
fig:Bsymm
nml:frd:acc:
thm:soundschemas

Theorem 50.1. *Let $\mathfrak{M} = \langle W, R, V \rangle$ be a model. If R has the property on the left side of Table 50.1, every instance of the formula on the right side is true in $\mathfrak{M}$.*

Proof. Here is the case for B: to show that the schema is true in a model we need to show that all of its instances are true at all worlds in the model. So let $\varphi \to \Box\Diamond\varphi$ be a given instance of B, and let $w \in W$ be an arbitrary world. Suppose the antecedent φ is true at w, in order to show that $\Box\Diamond\varphi$ is true at w. So we need to show that $\Diamond\varphi$ is true at all w' accessible from w. Now, for any w' such that Rww' we have, using the hypothesis of symmetry, that also $Rw'w$ (see Figure 50.1). Since $\mathfrak{M}, w \Vdash \varphi$, we have $\mathfrak{M}, w' \Vdash \Diamond\varphi$. Since w' was an arbitrary world such that Rww', we have $\mathfrak{M}, w \Vdash \Box\Diamond\varphi$.

We leave the other cases as exercises. □

Problem 50.1. Complete the proof of Theorem 50.1.

Notice that the converse implications of Theorem 50.1 do not hold: it's not true that if a model verifies a schema, then the accessibility relation of that model has the corresponding property. In the case of T and reflexive models, it is easy to give an example of a model in which T itself fails: let $W = \{w\}$ and $V(p) = \emptyset$. Then R is not reflexive, but $\mathfrak{M}, w \Vdash \Box p$ and $\mathfrak{M}, w \nVdash p$. But here we have just a single instance of T that fails in $\mathfrak{M}$; other instances, e.g., $\Box\neg p \to \neg p$ are true. It is harder to give examples where *every substitution instance* of T is true in $\mathfrak{M}$ and $\mathfrak{M}$ is not reflexive. But there are such models, too:

If R is ...	*then ... is true in $\mathfrak{M}$:*
partially functional: $\forall w \forall u \forall v((Rwu \wedge Rwv) \to u = v)$	$\Diamond p \to \Box p$
functional: $\forall w \exists v \forall u(Rwu \leftrightarrow u = v)$	$\Diamond p \leftrightarrow \Box p$
weakly dense: $\forall u \forall v(Ruv \to \exists w(Ruw \wedge Rwv))$	$\Box\Box p \to \Box p$
weakly connected: $\forall w \forall u \forall v((Rwu \wedge Rwv) \to (Ruv \vee u = v \vee Rvu))$	$\Box((p \wedge \Box p) \to q) \vee \Box((q \wedge \Box q) \to p)$ (L)
weakly directed: $\forall w \forall u \forall v((Rwu \wedge Rwv) \to \exists t(Rut \wedge Rvt))$	$\Diamond\Box p \to \Box\Diamond p$ (G)

Table 50.2: Five more correspondence facts.

nml:frd:acc:tab:anotherfive

nml:frd:acc:prop:reflexive

Proposition 50.2. *Let $\mathfrak{M} = \langle W, R, V\rangle$ be a model such that $W = \{u, v\}$, where worlds u and v are related by R: i.e., both Ruv and Rvu. Suppose that for all p: $u \in V(p) \Leftrightarrow v \in V(p)$. Then:*

1. *For all φ: $\mathfrak{M}, u \Vdash \varphi$ if and only if $\mathfrak{M}, v \Vdash \varphi$ (use induction on φ).*

2. *Every instance of* T *is true in $\mathfrak{M}$.*

Since $\mathfrak{M}$ is not reflexive (it is, in fact, irreflexive*), the converse of Theorem 50.1 fails in the case of* T *(similar arguments can be given for some—though not all—the other schemas mentioned in Theorem 50.1).*

Problem 50.2. Prove the claims in Proposition 50.2.

Although we will focus on the five classical formulas D, T, B, 4, and 5, we record in Table 50.2 a few more properties of accessibility relations. The accessibility relation R is partially functional, if from every world at most one world is accessible. If it is the case that from every world exactly one world is accessible, we call it functional. (Thus the functional relations are precisely those that are both serial and partially functional). They are called "functional" because the accessibility relation operates like a (partial) function. A relation is weakly dense if whenever Ruv, there is a w "between" u and v. So weakly dense relations are in a sense the opposite of transitive relations: in a transitive relation, whenever you can reach v from u by a detour via w, you can reach v from u directly; in a weakly dense relation, whenever you can reach v from u directly, you can also reach it by a detour via some w. A relation is weakly directed if whenever you can reach worlds u and v from some world w, you can reach a single world t from both u and v—this is sometimes called the "diamond property" or "confluence."

Problem 50.3. Let $\mathfrak{M} = \langle W, R, V\rangle$ be a model. Show that if R satisfies the left-hand properties of Table 50.2, every instance of the corresponding right-hand formula is true in $\mathfrak{M}$.

---

content/normal-modal-logic/frame-definability/frames.tex

50.3 Frames

nml:frd:fra:sec

Definition 50.3. A *frame* is a pair $\mathfrak{F} = \langle W, R\rangle$ where W is a non-empty set of worlds and R a binary relation on W. A model $\mathfrak{M}$ is *based on* a frame $\mathfrak{F} = \langle W, R\rangle$ if and only if $\mathfrak{M} = \langle W, R, V\rangle$ for some valuation V.

Definition 50.4. If $\mathfrak{F}$ is a frame, we say that φ is *valid in* $\mathfrak{F}$, $\mathfrak{F} \vDash \varphi$, if $\mathfrak{M} \Vdash \varphi$ for every model $\mathfrak{M}$ based on $\mathfrak{F}$.

If $\mathcal{F}$ is a class of frames, we say φ is *valid in* $\mathcal{F}$, $\mathcal{F} \vDash \varphi$, iff $\mathfrak{F} \vDash \varphi$ for every frame $\mathfrak{F} \in \mathcal{F}$.

The reason frames are interesting is that correspondence between schemas and properties of the accessibility relation R is at the level of frames, *not of models*. For instance, although T is true in all reflexive models, not every model in which T is true is reflexive. However, it *is* true that not only is T *valid* on all reflexive *frames*, also every frame in which T is valid is reflexive.

Remark 6. Validity in a class of frames is a special case of the notion of validity in a class of models: $\mathcal{F} \vDash \varphi$ iff $\mathcal{C} \vDash \varphi$ where $\mathcal{C}$ is the class of all models based on a frame in $\mathcal{F}$.

Obviously, if a formula or a schema is valid, i.e., valid with respect to the class of *all* models, it is also valid with respect to any class $\mathcal{F}$ of frames.

---

content/normal-modal-logic/frame-definability/definability.tex

50.4 Frame Definability

nml:frd:def:sec

Even though the converse implications of Theorem 50.1 fail, they hold if we replace "model" by "frame": for the properties considered in Theorem 50.1, it *is* true that if a formula is valid in a *frame* then the accessibility relation of that frame has the corresponding property. So, the formulas considered *define* the classes of frames that have the corresponding property.

Definition 50.5. If $\mathcal{F}$ is a class of frames, we say φ *defines* $\mathcal{F}$ iff $\mathfrak{F} \vDash \varphi$ for all and only frames $\mathfrak{F} \in \mathcal{F}$.

We now proceed to establish the full definability results for frames.

Theorem 50.6. *If the formula on the right side of Table 50.1 is valid in a frame $\mathfrak{F}$, then $\mathfrak{F}$ has the property on the left side.* nml:frd:def:thm:fullCorrespondence

Proof. 1. Suppose D is valid in $\mathfrak{F} = \langle W, R\rangle$, i.e., $\mathfrak{F} \vDash \Box p \to \Diamond p$. Let $\mathfrak{M} = \langle W, R, V\rangle$ be a model based on $\mathfrak{F}$, and $w \in W$. We have to show that there is a v such that Rwv. Suppose not: then both $\mathfrak{M} \Vdash \Box\varphi$ and $\mathfrak{M}, w \nVdash \Diamond\varphi$ for any φ, including p. But then $\mathfrak{M}, w \nVdash \Box p \to \Diamond p$, contradicting the assumption that $\mathfrak{F} \vDash \Box p \to \Diamond p$.

2. Suppose T is valid in $\mathfrak{F}$, i.e., $\mathfrak{F} \vDash \Box p \to p$. Let $w \in W$ be an arbitrary world; we need to show Rww. Let $u \in V(p)$ if and only if Rwu (when q is other than p, $V(q)$ is arbitrary, say $V(q) = \emptyset$). Let $\mathfrak{M} = \langle W, R, V\rangle$. By construction, for all u such that Rwu: $\mathfrak{M}, u \Vdash p$, and hence $\mathfrak{M}, w \Vdash \Box p$. But by hypothesis $\Box p \to p$ is true at w, so that $\mathfrak{M}, w \Vdash p$, but by definition of V this is possible only if Rww.

3. We prove the contrapositive: Suppose $\mathfrak{F}$ is not symmetric, we show that B, i.e., $p \to \Box\Diamond p$ is not valid in $\mathfrak{F} = \langle W, R\rangle$. If $\mathfrak{F}$ is not symmetric, there are $u, v \in W$ such that Ruv but not Rvu. Define V such that $w \in V(p)$ if and only if not Rvw (and V is arbitrary otherwise). Let $\mathfrak{M} = \langle W, R, V\rangle$. Now, by definition of V, $\mathfrak{M}, w \Vdash p$ for all w such that not Rvw, in particular, $\mathfrak{M}, u \Vdash p$ since not Rvu. Also, since Rvw iff $w \notin V(p)$, there is no w such that Rvw and $\mathfrak{M}, w \Vdash p$, and hence $\mathfrak{M}, v \nVdash \Diamond p$. Since Ruv, also $\mathfrak{M}, u \nVdash \Box\Diamond p$. It follows that $\mathfrak{M}, u \nVdash p \to \Box\Diamond p$, and so B is not valid in $\mathfrak{F}$.

4. Suppose 4 is valid in $\mathfrak{F} = \langle W, R\rangle$, i.e., $\mathfrak{F} \vDash \Box p \to \Box\Box p$, and let u, v, $w \in W$ be arbitrary worlds such that Ruv and Rvw; we need to show that Ruw. Define V such that $z \in V(p)$ if and only if Ruz (and V is arbitrary otherwise). Let $\mathfrak{M} = \langle W, R, V\rangle$. By definition of V, $\mathfrak{M}, z \Vdash p$ for all z such that Ruz, and hence $\mathfrak{M}, u \Vdash \Box p$. But by hypothesis 4, $\Box p \to \Box\Box p$, is true at u, so that $\mathfrak{M}, u \Vdash \Box\Box p$. Since Ruv and Rvw, we have $\mathfrak{M}, w \Vdash p$, but by definition of V this is possible only if Ruw, as desired.

5. We proceed contrapositively, assuming that the frame $\mathfrak{F} = \langle W, R\rangle$ is not euclidean, and show that it falsifies 5, i.e., $\mathfrak{F} \nvDash \Diamond p \to \Box\Diamond p$. Suppose there are worlds u, v, $w \in W$ such that Rwu and Rwv but not Ruv. Define V such that for all worlds z, $z \in V(p)$ if and only if it is *not* the case that Ruz. Let $\mathfrak{M} = \langle W, R, V\rangle$. Then by hypothesis $\mathfrak{M}, v \Vdash p$ and since Rwv also $\mathfrak{M}, w \Vdash \Diamond p$. However, there is no world y such that Ruy and $\mathfrak{M}, y \Vdash p$ so $\mathfrak{M}, u \nVdash \Diamond p$. Since Rwu, it follows that $\mathfrak{M}, w \nVdash \Box\Diamond p$, so that 5, $\Diamond p \to \Box\Diamond p$, fails at w. □

You'll notice a difference between the proof for D and the other cases: no mention was made of the valuation V. In effect, we proved that if $\mathfrak{M} \Vdash$ D then $\mathfrak{M}$ is serial. So D defines the class of serial *models*, not just frames.

nml:frd:def:prop:D-serial

Corollary 50.7. *Any model where* D *is true is serial.*

Corollary 50.8. *Each formula on the right side of Table 50.1 defines the class of frames which have the property on the left side.*

Proof. In Theorem 50.1, we proved that if a model has the property on the left, the formula on the right is true in it. Thus, if a frame $\mathfrak{F}$ has the property on the left, the formula on the right is valid in $\mathfrak{F}$. In Theorem 50.6, we proved the converse implications: if a formula on the right is valid in $\mathfrak{F}$, $\mathfrak{F}$ has the property on the left. □

Problem 50.4. Show that if the formula on the right side of Table 50.2 is valid in a frame $\mathfrak{F}$, then $\mathfrak{F}$ has the property on the left side. To do this, consider a frame that does *not* satisfy the property on the left, and define a suitable V such that the formula on the right is false at some world.

Theorem 50.6 also shows that the properties can be combined: for instance if both B and 4 are valid in $\mathfrak{F}$ then the frame is both symmetric and transitive, etc. Many important modal logics are characterized as the set of formulas valid in all frames that combine some frame properties, and so we can characterize them as the set of formulas valid in all frames in which the corresponding defining formulas are valid. For instance, the classical system **S4** is the set of all formulas valid in all reflexive and transitive frames, i.e., in all those where both T and 4 are valid. **S5** is the set of all formulas valid in all reflexive, symmetric, and euclidean frames, i.e., all those where all of T, B, and 5 are valid.

Logical relationships between properties of R in general correspond to relationships between the corresponding defining formulas. For instance, every reflexive relation is serial; hence, whenever T is valid in a frame, so is D. (Note that this relationship is *not* that of entailment. It is not the case that whenever $\mathfrak{M}, w \Vdash \mathrm{T}$ then $\mathfrak{M}, w \Vdash \mathrm{D}$.) We record some such relationships.

nml:frd:def:prop:relation-facts

Proposition 50.9. *Let R be a binary relation on a set W; then:*

1. *If R is reflexive, then it is serial.*
2. *If R is symmetric, then it is transitive if and only if it is euclidean.*
3. *If R is symmetric or euclidean then it is weakly directed (it has the "diamond property").*
4. *If R is euclidean then it is weakly connected.*
5. *If R is functional then it is serial.*

Problem 50.5. Prove Proposition 50.9.

---

content/normal-modal-logic/frame-definability/first-order-definability.tex

50.5 First-order Definability

nml:frd:fol:sec

We've seen that a number of properties of accessibility relations of frames can be defined by modal formulas. For instance, symmetry of frames can be defined by the formula B, $p \to \Box\Diamond p$. The conditions we've encountered so far can all be expressed by first-order formulas in a language involving a single two-place predicate symbol. For instance, symmetry is defined by $\forall x\, \forall y\, (Q(x, y) \to Q(y, x))$ in the sense that a first-order structure $\mathfrak{M}$ with $|\mathfrak{M}| = W$ and $Q^{\mathfrak{M}} = R$ satisfies the preceding formula iff R is symmetric. This suggests the following definition:

Definition 50.10. A class $\mathcal{F}$ of frames is *first-order definable* if there is a sentence φ in the first-order language with a single two-place predicate symbol Q such that $\mathfrak{F} = \langle W, R\rangle \in \mathcal{F}$ iff $\mathfrak{M} \vDash \varphi$ in the first-order structure $\mathfrak{M}$ with $|\mathfrak{M}| = W$ and $Q^{\mathfrak{M}} = R$.

It turns out that the properties and modal formulas that define them considered so far are exceptional. Not every formula defines a first-order definable class of frames, and not every first-order definable class of frames is definable by a modal formula.

A counterexample to the first is given by the Löb formula:

$$\Box(\Box p \to p) \to \Box p. \tag{W}$$

W defines the class of transitive and converse well-founded frames. A relation is well-founded if there is no infinite sequence w_1, w_2, ... such that Rw_2w_1, Rw_3w_2, ... For instance, the relation $<$ on $\mathbb{N}$ is well-founded, whereas the relation $<$ on $\mathbb{Z}$ is not. A relation is converse well-founded iff its converse is well-founded. So converse well-founded relations are those where there is no infinite sequence w_1, w_2, ... such that Rw_1w_2, Rw_2w_3, ...

There is, however, no first-order formula defining transitive converse well-founded relations. For suppose $\mathfrak{M} \vDash \beta$ iff $R = Q^{\mathfrak{M}}$ is transitive converse well-founded. Let φ_n be the formula

$$(Q(a_1, a_2) \wedge \cdots \wedge Q(a_{n-1}, a_n))$$

Now consider the set of formulas

$$\Gamma = \{\beta, \varphi_1, \varphi_2, \dots\}.$$

Every finite subset of Γ is satisfiable: Let k be largest such that φ_k is in the subset, $|\mathfrak{M}_k| = \{1, \dots, k\}$, $a_i^{\mathfrak{M}_k} = i$, and $Q^{\mathfrak{M}_k} = <$. Since $<$ on $\{1, \dots, k\}$ is transitive and converse well-founded, $\mathfrak{M}_k \vDash \beta$. $\mathfrak{M}_k \vDash \varphi_i$ by construction, for all $i \leq k$. By the Compactness Theorem for first-order logic, Γ is satisfiable in some structure $\mathfrak{M}$. By hypothesis, since $\mathfrak{M} \vDash \beta$, the relation $Q^{\mathfrak{M}}$ is converse well-founded. But clearly, $a_1^{\mathfrak{M}}$, $a_2^{\mathfrak{M}}$, ... would form an infinite sequence of the kind ruled out by converse well-foundedness.

A counterexample to the second claim is given by the property of universality: for every u and v, Ruv. Universal frames are first-order definable by the formula $\forall x \forall y\, Q(x, y)$. However, no modal formula is valid in all and only the universal frames. This is a consequence of a result that is independently interesting: the formulas valid in universal frames are exactly the same as those valid in reflexive, symmetric, and transitive frames. There are reflexive, symmetric, and transitive frames that are not universal, hence every formula valid in all universal frames is also valid in some non-universal frames.

---

content/normal-modal-logic/frame-definability/equivalence-S5.tex

50.6 Equivalence Relations and S5

nml:frd:es5:sec

The modal logic **S5** is characterized as the set of formulas valid on all universal frames, i.e., every world is accessible from every world, including itself. In such a scenario, $\Box$ corresponds to necessity and $\Diamond$ to possibility: $\Box\varphi$ is true if φ is true at *every* world, and $\Diamond\varphi$ is true if φ is true at *some* world. It turns out that **S5** can also be characterized as the formulas valid on all reflexive, symmetric, and transitive frames, i.e., on all *equivalence relations*.

Definition 50.11. A binary relation R on W is an *equivalence relation* if and only if it is reflexive, symmetric and transitive. A relation R on W is *universal* if and only if Ruv for all $u, v \in W$.

Since T, B, and 4 characterize the reflexive, symmetric, and transitive frames, the frames where the accessibility relation is an equivalence relation are exactly those in which all three formulas are valid. It turns out that the equivalence relations can also be characterized by other combinations of formulas, since the conditions with which we've defined equivalence relations are equivalent to combinations of other familiar conditions on R.

nml:frd:es5:prop:equivalences

Proposition 50.12. *The following are equivalent:*

1. *R is an equivalence relation;*
2. *R is reflexive and euclidean;*
3. *R is serial, symmetric, and euclidean;*
4. *R is serial, symmetric, and transitive.*

Proof. Exercise. □

Problem 50.6. Prove Proposition 50.12 by showing:

1. If R is symmetric and transitive, it is euclidean.
2. If R is reflexive, it is serial.

3. If R is reflexive and euclidean, it is symmetric.

4. If R is symmetric and euclidean, it is transitive.

5. If R is serial, symmetric, and transitive, it is reflexive.

Explain why this suffices for the proof that the conditions are equivalent.

Proposition 50.12 is the semantic counterpart to Proposition 51.30, in that it gives an equivalent characterization of the modal logic of frames over which R is an equivalence relation (the logic traditionally referred to as **S5**).

What is the relationship between universal and equivalence relations? Although every universal relation is an equivalence relation, clearly not every equivalence relation is universal. However, the formulas valid on all universal relations are exactly the same as those valid on all equivalence relations.

Proposition 50.13. *Let R be an equivalence relation, and for each $w \in W$ define the* equivalence class *of w as the set $[w] = \{w' \in W : Rww'\}$. Then:*

1. *$w \in [w]$;*

2. *R is universal on each equivalence class $[w]$;*

3. *The collection of equivalence classes partitions W into mutually exclusive and jointly exhaustive subsets.*

nml:frd:es5: prop:S5=univ

Proposition 50.14. *A formula φ is valid in all frames $\mathfrak{F} = \langle W, R\rangle$ where R is an equivalence relation, if and only if it is valid in all frames $\mathfrak{F} = \langle W, R\rangle$ where R is universal. Hence, the logic of universal frames is just* **S5**.

Proof. It's immediate to verify that a universal relation R on W is an equivalence. Hence, if φ is valid in all frames where R is an equivalence it is valid in all universal frames. For the other direction, we argue contrapositively: suppose ψ is a formula that fails at a world w in a model $\mathfrak{M} = \langle W, R, V\rangle$ based on a frame $\langle W, R\rangle$, where R is an equivalence on W. So $\mathfrak{M}, w \nVdash \psi$. Define a model $\mathfrak{M}' = \langle W', R', V'\rangle$ as follows:

1. $W' = [w]$;

2. R' is universal on W';

3. $V'(p) = V(p) \cap W'$.

(So the set W' of worlds in $\mathfrak{M}'$ is represented by the shaded area in Figure 50.2.) It is easy to see that R and R' agree on W'. Then one can show by induction on formulas that for all $w' \in W'$: $\mathfrak{M}', w' \Vdash \varphi$ if and only if $\mathfrak{M}, w' \Vdash \varphi$ for each φ (this makes sense since $W' \subseteq W$). In particular, $\mathfrak{M}', w \nVdash \psi$, and ψ fails in a model based on a universal frame. □

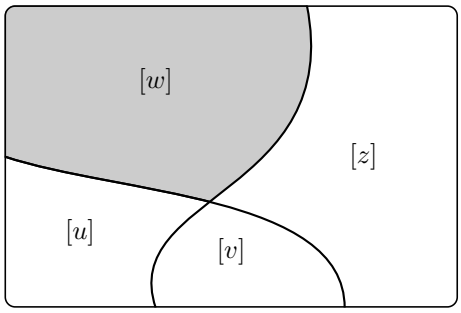


Figure 50.2: A partition of W in equivalence classes.

nml:frd:es5:fig:partition

50.7 Second-order Definability

nml:frd:st:sec

Not every frame property definable by modal formulas is first-order definable. However, if we allow quantification over one-place predicates (i.e., monadic second-order quantification), we define all modally definable frame properties. The trick is to exploit a systematic way in which the conditions under which a modal formula is true at a world are related to first-order formulas. This is the so-called standard translation of modal formulas into first-order formulas in a language containing not just a two-place predicate symbol Q for the accessibility relation, but also a one-place predicate symbol P_i for the propositional variables p_i occurring in φ.

Definition 50.15. The *standard translation* $\mathrm{ST}_x(\varphi)$ is inductively defined as follows:

1. $\varphi \equiv \bot$: $\mathrm{ST}_x(\varphi) = \bot$.
2. $\varphi \equiv \bot$: $\mathrm{ST}_x(\varphi) = \top$.
3. $\varphi \equiv p_i$: $\mathrm{ST}_x(\varphi) = P_i(x)$.
4. $\varphi \equiv \neg\psi$: $\mathrm{ST}_x(\varphi) = \neg\mathrm{ST}_x(\psi)$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathrm{ST}_x(\varphi) = (\mathrm{ST}_x(\psi) \wedge \mathrm{ST}_x(\chi))$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathrm{ST}_x(\varphi) = (\mathrm{ST}_x(\psi) \vee \mathrm{ST}_x(\chi))$.
7. $\varphi \equiv (\psi \to \chi)$: $\mathrm{ST}_x(\varphi) = (\mathrm{ST}_x(\psi) \to \mathrm{ST}_x(\chi))$.
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathrm{ST}_x(\varphi) = (\mathrm{ST}_x(\psi) \leftrightarrow \mathrm{ST}_x(\chi))$.
9. $\varphi \equiv \Box\psi$: $\mathrm{ST}_x(\varphi) = \forall y\,(Q(x,y) \to \mathrm{ST}_y(\psi))$.
10. $\varphi \equiv \Diamond\psi$: $\mathrm{ST}_x(\varphi) = \exists y\,(Q(x,y) \wedge \mathrm{ST}_y(\psi))$.

For instance, $\mathrm{ST}_x(\Box p \to p)$ is $\forall y\,(Q(x,y) \to P(y)) \to P(x)$. Any structure for the language of $\mathrm{ST}_x(\varphi)$ requires a domain, a two-place relation assigned to Q, and subsets of the domain assigned to the one-place predicate symbols P_i. In other words, the components of such a structure are exactly those of a model for φ: the domain is the set of worlds, the two-place relation assigned to Q is the accessibility relation, and the subsets assigned to P_i are just the assignments $V(p_i)$. It won't surprise that satisfaction of φ in a modal model and of $\mathrm{ST}_x(\varphi)$ in the corresponding structure agree:

nml:frd:st:prop:st

Proposition 50.16. *Let $\mathfrak{M} = \langle W, R, V\rangle$, $\mathfrak{M}'$ be the first-order structure with $|\mathfrak{M}'| = W$, $Q^{\mathfrak{M}'} = R$, and $P_i^{\mathfrak{M}'} = V(p_i)$, and $s(x) = w$. Then*

$$\mathfrak{M}, w \Vdash \varphi \text{ iff } \mathfrak{M}', s \vDash \mathrm{ST}_x(\varphi)$$

Proof. By induction on φ. □

Proposition 50.17. *Suppose φ is a modal formula and $\mathfrak{F} = \langle W, R\rangle$ is a frame. Let $\mathfrak{F}'$ be the first-order structure with $|\mathfrak{F}'| = W$ and $Q^{\mathfrak{F}'} = R$, and let φ' be the second-order formula*

$$\forall X_1 \ldots \forall X_n\, \forall x\, \mathrm{ST}_x(\varphi)[X_1/P_1, \ldots, X_n/P_n],$$

where $P_1, \ldots, P_n$ are all one-place predicate symbols in $\mathrm{ST}_x(\varphi)$. Then

$$\mathfrak{F} \vDash \varphi \text{ iff } \mathfrak{F}' \vDash \varphi'$$

Proof. $\mathfrak{F}' \vDash \varphi'$ iff for every structure $\mathfrak{M}'$ where $P_i^{\mathfrak{M}'} \subseteq W$ for $i = 1, \ldots, n$, and for every s with $s(x) \in W$, $\mathfrak{M}', s \vDash \mathrm{ST}_x(\varphi)$. By Proposition 50.16, that is the case iff for all models $\mathfrak{M}$ based on $\mathfrak{F}$ and every world $w \in W$, $\mathfrak{M}, w \Vdash \varphi$, i.e., $\mathfrak{F} \vDash \varphi$. □

Definition 50.18. A class $\mathcal{F}$ of frames is *second-order definable* if there is a sentence φ in the second-order language with a single two-place predicate symbol P and quantifiers only over monadic set variables such that $\mathfrak{F} = \langle W, R\rangle \in \mathcal{F}$ iff $\mathfrak{M} \vDash \varphi$ in the structure $\mathfrak{M}$ with $|\mathfrak{M}| = W$ and $P^{\mathfrak{M}} = R$.

Corollary 50.19. *If a class of frames is definable by a formula φ, the corresponding class of accessibility relations is definable by a monadic second-order sentence.*

Proof. The monadic second-order sentence φ' of the preceding proof has the required property. □

As an example, consider again the formula $\Box p \to p$. It defines reflexivity. Reflexivity is of course first-order definable by the sentence $\forall x\, Q(x,x)$. But it is also definable by the monadic second-order sentence

$$\forall X\, \forall x\, (\forall y\, (Q(x,y) \to X(y)) \to X(x)).$$

This means, of course, that the two sentences are equivalent. Here's how you might convince yourself of this directly: First suppose the second-order sentence is true in a structure $\mathfrak{M}$. Since x and X are universally quantified, the remainder must hold for any $x \in W$ and set $X \subseteq W$, e.g., the set $\{z : Rxz\}$ where $R = Q^{\mathfrak{M}}$. So, for any s with $s(x) \in W$ and $s(X) = \{z : Rxz\}$ we have $\mathfrak{M} \vDash \forall y\,(Q(x,y) \to X(y)) \to X(x)$. But by the way we've picked $s(X)$ that means $\mathfrak{M}, s \vDash \forall y\,(Q(x,y) \to Q(x,y)) \to Q(x,x)$, which is equivalent to $Q(x,x)$ since the antecedent is valid. Since $s(x)$ is arbitrary, we have $\mathfrak{M} \vDash \forall x\, Q(x,x)$.

Now suppose that $\mathfrak{M} \vDash \forall x\, Q(x,x)$ and show that $\mathfrak{M} \vDash \forall X\, \forall x\, (\forall y\,(Q(x,y) \to X(y)) \to X(x))$. Pick any assignment s, and assume $\mathfrak{M}, s \vDash \forall y\,(Q(x,y) \to X(y))$. Let s' be the y-variant of s with $s'(y) = s(x)$; we have $\mathfrak{M}, s' \vDash Q(x,y) \to X(y)$, i.e., $\mathfrak{M}, s \vDash Q(x,x) \to X(x)$. Since $\mathfrak{M} \vDash \forall x\, Q(x,x)$, the antecedent is true, and we have $\mathfrak{M}, s \vDash X(x)$, which is what we needed to show.

Since some definable classes of frames are not first-order definable, not every monadic second-order sentence of the form φ' is equivalent to a first-order sentence. There is no effective method to decide which ones are.

Chapter 51

Axiomatic Derivations

---

content/normal-modal-logic/axioms-systems/introduction.tex

51.1 Introduction

nml:axs:int:sec

We have a semantics for the basic modal language in terms of modal models, and a notion of a formula being valid—true at all worlds in all models—or valid with respect to some class of models or frames—true at all worlds in all models in the class, or based on the frame. Logic usually connects such semantic characterizations of validity with a proof-theoretic notion of derivability. The aim is to define a notion of derivability in some system such that a formula is derivable iff it is valid.

The simplest and historically oldest derivation systems are so-called Hilbert-type or axiomatic derivation systems. Hilbert-type derivation systems for many modal logics are relatively easy to construct: they are simple as objects of metatheoretical study (e.g., to prove soundness and completeness). However,

they are much harder to use to prove formulas in than, say, natural deduction systems.

In Hilbert-type derivation systems, a derivation of a formula is a sequence of formulas leading from certain axioms, via a handful of inference rules, to the formula in question. Since we want the derivation system to match the semantics, we have to guarantee that the set of derivable formulas are true in all models (or true in all models in which all axioms are true). We'll first isolate some properties of modal logics that are necessary for this to work: the "normal" modal logics. For normal modal logics, there are only two inference rules that need to be assumed: modus ponens and necessitation. As axioms we take all (substitution instances) of tautologies, and, depending on the modal logic we deal with, a number of modal axioms. Even if we are just interested in the class of all models, we must also count all substitution instances of K and Dual as axioms. This alone generates the minimal normal modal logic **K**.

Definition 51.1. The rule of *modus ponens* is the inference schema

$$\frac{\varphi \qquad \varphi \to \psi}{\psi}\ \text{MP}$$

We say a formula ψ *follows from* formulas φ, χ by modus ponens iff $\chi \equiv \varphi \to \psi$.

Definition 51.2. The rule of *necessitation* is the inference schema

$$\frac{\varphi}{\Box\varphi}\ \text{NEC}$$

We say the formula ψ follows from the formulas φ by necessitation iff $\psi \equiv \Box\varphi$.

Definition 51.3. A *derivation* from a set of axioms Σ is a sequence of formulas $\psi_1, \psi_2, \ldots, \psi_n$, where each ψ_i is either

1. a substitution instance of a tautology, or
2. a substitution instance of a formula in Σ, or
3. follows from two formulas ψ_j, ψ_k with $j, k < i$ by modus ponens, or
4. follows from a formula ψ_j with $j < i$ by necessitation.

If there is such a derivation with $\psi_n \equiv \varphi$, we say that φ is *derivable from* Σ, in symbols $\Sigma \vdash \varphi$.

With this definition, it will turn out that the set of derivable formulas forms a normal modal logic, and that any derivable formula is true in every model in which every axiom is true. This property of derivations is called *soundness*. The converse, *completeness*, is harder to prove.

---

content/normal-modal-logic/axioms-systems/normal-logics.tex

51.2 Normal Modal Logics

nml:prf:nor:sec

Not every set of modal formulas can easily be characterized as those formulas derivable from a set of axioms. We want modal logics to be well-behaved. First of all, everything we can derive in classical propositional logic should still be derivable, of course taking into account that the formulas may now contain also $\Box$ and $\Diamond$. To this end, we require that a modal logic contain all tautological instances and be closed under modus ponens.

Definition 51.4. A *modal logic* is a set Σ of modal formulas which

1. contains all tautologies, and
2. is closed under substitution, i.e., if $\varphi \in \Sigma$, and $\theta_1, \ldots, \theta_n$ are formulas, then
$$\varphi[\theta_1/p_1, \ldots, \theta_n/p_n] \in \Sigma,$$
3. is closed under *modus ponens*, i.e., if φ and $\varphi \to \psi \in \Sigma$, then $\psi \in \Sigma$.

In order to use the relational semantics for modal logics, we also have to require that all formulas valid in all modal models are included. It turns out that this requirement is met as soon as all instances of K and DUAL are derivable, and whenever a formula φ is derivable, so is $\Box\varphi$. A modal logic that satisfies these conditions is called *normal*. (Of course, there are also non-normal modal logics, but the usual relational models are not adequate for them.)

Definition 51.5. A modal logic Σ is *normal* if it contains

$$\Box(p \to q) \to (\Box p \to \Box q), \tag{K}$$
$$\Diamond p \leftrightarrow \neg\Box\neg p \tag{DUAL}$$

and is closed under *necessitation*, i.e., if $\varphi \in \Sigma$, then $\Box\varphi \in \Sigma$.

Observe that while tautological implication is "fine-grained" enough to preserve *truth at a world*, the rule NEC only preserves *truth in a model* (and hence also validity in a frame or in a class of frames).

nml:prf:nor:prop:rk

Proposition 51.6. *Every normal modal logic is closed under rule* RK*,*

$$\frac{\varphi_1 \to (\varphi_2 \to \cdots (\varphi_{n-1} \to \varphi_n)\cdots)}{\Box\varphi_1 \to (\Box\varphi_2 \to \cdots (\Box\varphi_{n-1} \to \Box\varphi_n)\cdots).}\ \text{RK}$$

Proof. By induction on n: If $n = 1$, then the rule is just NEC, and every normal modal logic is closed under NEC.

Now suppose the result holds for $n-1$; we show it holds for n.

Assume

$$\varphi_1 \to (\varphi_2 \to \cdots (\varphi_{n-1} \to \varphi_n)\cdots) \in \Sigma$$

By the induction hypothesis, we have

$$\Box\varphi_1 \to (\Box\varphi_2 \to \cdots \Box(\varphi_{n-1} \to \varphi_n) \cdots) \in \Sigma$$

Since Σ is a normal modal logic, it contains all instances of K, in particular

$$\Box(\varphi_{n-1} \to \varphi_n) \to (\Box\varphi_{n-1} \to \Box\varphi_n) \in \Sigma$$

Using modus ponens and suitable tautological instances we get

$$\Box\varphi_1 \to (\Box\varphi_2 \to \cdots (\Box\varphi_{n-1} \to \Box\varphi_n) \cdots) \in \Sigma. \qquad \Box$$

Proposition 51.7. *Every normal modal logic Σ contains $\neg\Diamond\bot$.*

nml:prf:nor:prop:notDiamondBot

Problem 51.1. Prove Proposition 51.7.

Proposition 51.8. *Let $\varphi_1, \ldots, \varphi_n$ be formulas. Then there is a smallest modal logic Σ containing all instances of $\varphi_1, \ldots, \varphi_n$.*

Proof. Given $\varphi_1, \ldots, \varphi_n$, define Σ as the intersection of all normal modal logics containing all instances of $\varphi_1, \ldots, \varphi_n$. The intersection is non-empty as $\mathrm{Frm}(\mathcal{L})$, the set of all formulas, is such a modal logic. $\Box$

Definition 51.9. The smallest normal modal logic containing $\varphi_1, \ldots, \varphi_n$ is called a *modal system* and denoted by $\mathbf{K}\varphi_1 \ldots \varphi_n$. The smallest normal modal logic is denoted by $\mathbf{K}$.

---

content/normal-modal-logic/axioms-systems/logics-proofs.tex

51.3 Derivations and Modal Systems

We first define what a derivation is for normal modal logics. Roughly, a derivation is a sequence of formulas in which every element is either (a substitution instance of) one of a number of *axioms*, or follows from previous elements by one of a few inference rules. For normal modal logics, all instances of tautologies, K, and DUAL count as axioms. This results in the modal system **K**, the smallest normal modal logic. We may wish to add additional axioms to obtain other systems, however. The rules are always modus ponens MP and necessitation NEC.

nml:prf:prf:sec

Definition 51.10. Given a modal system $\mathbf{K}\varphi_1 \ldots \varphi_n$ and a formula ψ we say that ψ is *derivable* in $\mathbf{K}\varphi_1 \ldots \varphi_n$, written $\mathbf{K}\varphi_1 \ldots \varphi_n \vdash \psi$, if and only if there are formulas $\chi_1, \ldots, \chi_k$ such that $\chi_k = \psi$ and each χ_i is either a tautological instance, or an instance of one of K, DUAL, $\varphi_1, \ldots, \varphi_n$, or it follows from previous formulas by means of the rules MP or NEC.

The following proposition allows us to show that $\psi \in \Sigma$ by exhibiting a Σ-derivation of ψ.

Proposition 51.11. $\mathbf{K}\varphi_1 \dots \varphi_n = \{\psi : \mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi\}$.

Proof. We use induction on the length of derivations to show that $\{\psi : \mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi\} \subseteq \mathbf{K}\varphi_1 \dots \varphi_n$.

If the derivation of ψ has length 1, it contains a single formula. That formula cannot follow from previous formulas by MP or NEC, so must be a tautological instance, an instance of K, DUAL, or an instance of one of $\varphi_1, \dots, \varphi_n$. But $\mathbf{K}\varphi_1 \dots \varphi_n$ contains these as well, so $\psi \in \mathbf{K}\varphi_1 \dots \varphi_n$.

If the derivation of ψ has length > 1, then ψ may in addition be obtained by MP or NEC from formulas not occurring as the last line in the derivation. If ψ follows from χ and $\chi \to \psi$ (by MP), then χ and $\chi \to \psi \in \mathbf{K}\varphi_1 \dots \varphi_n$ by induction hypothesis. But every modal logic is closed under modus ponens, so $\psi \in \mathbf{K}\varphi_1 \dots \varphi_n$. If $\psi \equiv \Box\chi$ follows from χ by NEC, then $\chi \in \mathbf{K}\varphi_1 \dots \varphi_n$ by induction hypothesis. But every normal modal logic is closed under NEC, so $\psi \in \mathbf{K}\varphi_1 \dots \varphi_n$.

The converse inclusion follows by showing that $\Sigma = \{\psi : \mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi\}$ is a normal modal logic containing all the instances of $\varphi_1, \dots, \varphi_n$, and the observation that $\mathbf{K}\varphi_1 \dots \varphi_n$ is, by definition, the smallest such logic.

1. Every tautology ψ is a tautological instance, so $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi$, so Σ contains all tautologies.

2. If $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \chi$ and $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \chi \to \psi$, then $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi$: Combine the derivation of χ with that of $\chi \to \psi$, and add the line ψ. The last line is justified by MP. So Σ is closed under modus ponens.

3. If ψ has a derivation, then every substitution instance of ψ also has a derivation: apply the substitution to every formula in the derivation. (Exercise: prove by induction on the length of derivations that the result is also a correct derivation). So Σ is closed under uniform substitution. (We have now established that Σ satisfies all conditions of a modal logic.)

4. We have $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \mathrm{K}$, so $K \in \Sigma$.

5. We have $\mathbf{K}\varphi_1 \dots \varphi_n \vdash$ DUAL, so DUAL $\in \Sigma$.

6. If $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \chi$, the additional line $\Box\chi$ is justified by NEC. Consequently, Σ is closed under NEC. Thus, Σ is normal. □

51.4 Proofs in K

In order to practice proofs in the smallest modal system, we show the valid formulas on the left-hand side of Table 49.1 can all be given **K**-proofs. nml:prf:prk:sec

Proposition 51.12. $\mathbf{K} \vdash \Box\varphi \to \Box(\psi \to \varphi)$

Proof.

1.	$\varphi \to (\psi \to \varphi)$	TAUT
2.	$\Box(\varphi \to (\psi \to \varphi))$	NEC, 1
3.	$\Box(\varphi \to (\psi \to \varphi)) \to (\Box\varphi \to \Box(\psi \to \varphi))$	K
4.	$\Box\varphi \to \Box(\psi \to \varphi)$	MP, 2, 3

□

Proposition 51.13. $\mathbf{K} \vdash \Box(\varphi \wedge \psi) \to (\Box\varphi \wedge \Box\psi)$

Proof.

1.	$(\varphi \wedge \psi) \to \varphi$	TAUT
2.	$\Box((\varphi \wedge \psi) \to \varphi)$	NEC
3.	$\Box((\varphi \wedge \psi) \to \varphi) \to (\Box(\varphi \wedge \psi) \to \Box\varphi)$	K
4.	$\Box(\varphi \wedge \psi) \to \Box\varphi$	MP, 2, 3
5.	$(\varphi \wedge \psi) \to \psi$	TAUT
6.	$\Box((\varphi \wedge \psi) \to \psi)$	NEC
7.	$\Box((\varphi \wedge \psi) \to \psi) \to (\Box(\varphi \wedge \psi) \to \Box\psi)$	K
8.	$\Box(\varphi \wedge \psi) \to \Box\psi$	MP, 6, 7
9.	$(\Box(\varphi \wedge \psi) \to \Box\varphi) \to$ $((\Box(\varphi \wedge \psi) \to \Box\psi) \to$ $(\Box(\varphi \wedge \psi) \to (\Box\varphi \wedge \Box\psi)))$	TAUT
10.	$(\Box(\varphi \wedge \psi) \to \Box\psi) \to$ $(\Box(\varphi \wedge \psi) \to (\Box\varphi \wedge \Box\psi))$	MP, 4, 9
11.	$\Box(\varphi \wedge \psi) \to (\Box\varphi \wedge \Box\psi)$	MP, 8, 10.

Note that the formula on line 9 is an instance of the tautology

$$(p \to q) \to ((p \to r) \to (p \to (q \wedge r))).$$

□

Proposition 51.14. $\mathbf{K} \vdash (\Box\varphi \wedge \Box\psi) \to \Box(\varphi \wedge \psi)$

Proof.

1.	$\varphi \to (\psi \to (\varphi \wedge \psi))$	TAUT
2.	$\Box(\varphi \to (\psi \to (\varphi \wedge \psi)))$	NEC, 1
3.	$\Box(\varphi \to (\psi \to (\varphi \wedge \psi))) \to (\Box\varphi \to \Box(\psi \to (\varphi \wedge \psi)))$	K
4.	$\Box\varphi \to \Box(\psi \to (\varphi \wedge \psi))$	MP, 2, 3
5.	$\Box(\psi \to (\varphi \wedge \psi)) \to (\Box\psi \to \Box(\varphi \wedge \psi))$	K
6.	$(\Box\varphi \to \Box(\psi \to (\varphi \wedge \psi))) \to$ $(\Box(\psi \to (\varphi \wedge \psi)) \to (\Box\psi \to \Box(\varphi \wedge \psi))) \to$ $(\Box\varphi \to (\Box\psi \to \Box(\varphi \wedge \psi))))$	TAUT
7.	$(\Box(\psi \to (\varphi \wedge \psi)) \to (\Box\psi \to \Box(\varphi \wedge \psi))) \to$ $(\Box\varphi \to (\Box\psi \to \Box(\varphi \wedge \psi)))$	MP, 4, 6
8.	$\Box\varphi \to (\Box\psi \to \Box(\varphi \wedge \psi)))$	MP, 5, 7
9.	$(\Box\varphi \to (\Box\psi \to \Box(\varphi \wedge \psi)))) \to$ $((\Box\varphi \wedge \Box\psi) \to \Box(\varphi \wedge \psi))$	TAUT
10.	$(\Box\varphi \wedge \Box\psi) \to \Box(\varphi \wedge \psi)$	MP, 8, 9

The formulas on lines 6 and 9 are instances of the tautologies

$$(p \to q) \to ((q \to r) \to (p \to r))$$
$$(p \to (q \to r)) \to ((p \wedge q) \to r)$$

□

Proposition 51.15. $\mathbf{K} \vdash \neg\Box p \to \Diamond\neg p$

Proof.

1.	$\Diamond\neg p \leftrightarrow \neg\Box\neg\neg p$	DUAL
2.	$(\Diamond\neg p \leftrightarrow \neg\Box\neg\neg p) \to$ $(\neg\Box\neg\neg p \to \Diamond\neg p)$	TAUT
3.	$\neg\Box\neg\neg p \to \Diamond\neg p$	MP, 1, 2
4.	$\neg\neg p \to p$	TAUT
5.	$\Box(\neg\neg p \to p)$	NEC, 4
6.	$\Box(\neg\neg p \to p) \to (\Box\neg\neg p \to \Box p)$	K
7.	$(\Box\neg\neg p \to \Box p)$	MP, 5, 6
8.	$(\Box\neg\neg p \to \Box p) \to (\neg\Box p \to \neg\Box\neg\neg p)$	TAUT
9.	$\neg\Box p \to \neg\Box\neg\neg p$	MP, 7, 8
10.	$(\neg\Box p \to \neg\Box\neg\neg p) \to$ $((\neg\Box\neg\neg p \to \Diamond\neg p) \to (\neg\Box p \to \Diamond\neg p))$	TAUT
11.	$(\neg\Box\neg\neg p \to \Diamond\neg p) \to (\neg\Box p \to \Diamond\neg p)$	MP, 9, 10
12.	$\neg\Box p \to \Diamond\neg p$	MP, 3, 11

The formulas on lines 8 and 10 are instances of the tautologies

$$(p \to q) \to (\neg q \to \neg p)$$
$$(p \to q) \to ((q \to r) \to (p \to r)).$$

□

Problem 51.2. Find derivations in **K** for the following formulas:

1. $\Box\neg p \to \Box(p \to q)$

2. $(\Box p \vee \Box q) \to \Box(p \vee q)$
3. $\Diamond p \to \Diamond(p \vee q)$

---

content/normal-modal-logic/axioms-systems/derived-rules.tex

51.5 Derived Rules

nml:prf:der:sec

Finding and writing derivations is obviously difficult, cumbersome, and repetitive. For instance, very often we want to pass from $\varphi \to \psi$ to $\Box\varphi \to \Box\psi$, i.e., apply rule RK. That requires an application of NEC, then recording the proper instance of K, then applying MP. Passing from $\varphi \to \psi$ and $\psi \to \chi$ to $\varphi \to \chi$ requires recording the (long) tautological instance

$$(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$$

and applying MP twice. Often we want to replace a sub-formula by a formula we know to be equivalent, e.g., $\Diamond\varphi$ by $\neg\Box\neg\varphi$, or $\neg\neg\varphi$ by φ. So rather than write out the actual derivation, it is more convenient to simply record why the intermediate steps are derivable. For this purpose, let us collect some facts about derivability.

Proposition 51.16. *If* $\mathbf{K} \vdash \varphi_1, \ldots, \mathbf{K} \vdash \varphi_n$, *and* ψ *follows from* $\varphi_1, \ldots, \varphi_n$ *by propositional logic, then* $\mathbf{K} \vdash \psi$.

Proof. If ψ follows from $\varphi_1, \ldots, \varphi_n$ by propositional logic, then

$$\varphi_1 \to (\varphi_2 \to \cdots (\varphi_n \to \psi) \ldots)$$

is a tautological instance. Applying MP n times gives a derivation of ψ. □

We will indicate use of this proposition by PL.

Proposition 51.17. *If* $\mathbf{K} \vdash \varphi_1 \to (\varphi_2 \to \cdots (\varphi_{n-1} \to \varphi_n) \ldots)$ *then* $\mathbf{K} \vdash \Box\varphi_1 \to (\Box\varphi_2 \to \cdots (\Box\varphi_{n-1} \to \Box\varphi_n) \ldots)$.

Proof. By induction on n, just as in the proof of Proposition 51.6. □

We will indicate use of this proposition by RK. Let's illustrate how these results help establishing derivability results more easily.

Proposition 51.18. $\mathbf{K} \vdash (\Box\varphi \wedge \Box\psi) \to \Box(\varphi \wedge \psi)$

Proof.

1.	$\mathbf{K} \vdash \varphi \to (\psi \to (\varphi \wedge \psi))$	TAUT
2.	$\mathbf{K} \vdash \Box\varphi \to (\Box\psi \to \Box(\varphi \wedge \psi)))$	RK, 1
3.	$\mathbf{K} \vdash (\Box\varphi \wedge \Box\psi) \to \Box(\varphi \wedge \psi)$	PL, 2

□

nml:prf:der:prop:rewriting

Proposition 51.19. *If* $\mathbf{K} \vdash \varphi \leftrightarrow \psi$ *and* $\mathbf{K} \vdash \chi[\varphi/q]$ *then* $\mathbf{K} \vdash \chi[B/q]$

Proof. Exercise. □

Problem 51.3. Prove Proposition 51.19 by proving, by induction on the complexity of χ, that if $\mathbf{K} \vdash \varphi \leftrightarrow \psi$ then $\mathbf{K} \vdash \chi[\varphi/q] \leftrightarrow \chi[\psi/q]$.

This proposition comes in handy especially when we want to convert $\Diamond$ into $\Box$ (or vice versa), or remove double negations inside a formula. In what follows, we will mark applications of Proposition 51.19 by "φ for ψ" whenever we re-write a formula $\chi(\psi)$ for $\chi(\varphi)$. In other words, "φ for ψ" abbreviates:

$\vdash \chi(\varphi)$
$\vdash \varphi \leftrightarrow \psi$
$\vdash \chi(\psi)$ by Proposition 51.19

For instance:

Proposition 51.20. $\mathbf{K} \vdash \neg\Box p \to \Diamond\neg p$

Proof.

1. $\mathbf{K} \vdash \Diamond\neg p \leftrightarrow \neg\Box\neg\neg p$ DUAL
2. $\mathbf{K} \vdash \neg\Box\neg\neg\neg p \to \Diamond\neg p$ PL, 1
3. $\mathbf{K} \vdash \neg\Box p \to \Diamond\neg p$ p for $\neg\neg p$

□

In the above derivation, the final step "p for $\neg\neg p$" is short for

$\mathbf{K} \vdash \neg\Box\neg\neg p \to \Diamond\neg p$
$\mathbf{K} \vdash \neg\neg p \leftrightarrow p$ TAUT
$\mathbf{K} \vdash \neg\Box p \to \Diamond\neg p$ by Proposition 51.19

The roles of $\chi(q)$, φ, and ψ in Proposition 51.19 are played here, respectively, by $\neg\Box q \to \Diamond\neg p$, $\neg\neg p$, and p.

When a formula contains a sub-formula $\neg\Diamond\varphi$, we can replace it by $\Box\neg\varphi$ using Proposition 51.19, since $\mathbf{K} \vdash \neg\Diamond\varphi \leftrightarrow \Box\neg\varphi$. We'll indicate this and similar replacements simply by "$\Box\neg$ for $\neg\Diamond$."

The following proposition justifies that we can establish derivability results schematically. E.g., the previous proposition does not just establish that $\mathbf{K} \vdash \neg\Box p \to \Diamond\neg p$, but $\mathbf{K} \vdash \neg\Box\varphi \to \Diamond\neg\varphi$ for arbitrary φ.

Proposition 51.21. *If* φ *is a substitution instance of* ψ *and* $\mathbf{K} \vdash \psi$*, then* $\mathbf{K} \vdash \varphi$*.*

Proof. It is tedious but routine to verify (by induction on the length of the derivation of ψ) that applying a substitution to an entire derivation also results in a correct derivation. Specifically, substitution instances of tautological instances are themselves tautological instances, substitution instances of instances of DUAL and K are themselves instances of DUAL and K, and applications of MP and NEC remain correct when substituting formulas for propositional variables in both premise(s) and conclusion. □

content/normal-modal-logic/axioms-systems/more-proofs-in-K.tex

51.6 More Proofs in K

Let's see some more examples of derivability in **K**, now using the simplified method introduced in section 51.5. nml:prf:mpr:sec

Proposition 51.22. $\mathbf{K} \vdash \Box(\varphi \to \psi) \to (\Diamond\varphi \to \Diamond\psi)$

Proof.

1.	$\mathbf{K} \vdash (\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$	PL
2.	$\mathbf{K} \vdash \Box(\varphi \to \psi) \to (\Box\neg\psi \to \Box\neg\varphi)$	RK, 1
3.	$\mathbf{K} \vdash (\Box\neg\psi \to \Box\neg\varphi) \to (\neg\Box\neg\varphi \to \neg\Box\neg\psi)$	TAUT
4.	$\mathbf{K} \vdash \Box(\varphi \to \psi) \to (\neg\Box\neg\varphi \to \neg\Box\neg\psi)$	PL, 2, 3
5.	$\mathbf{K} \vdash \Box(\varphi \to \psi) \to (\Diamond\varphi \to \Diamond\psi)$	$\Diamond$ for $\neg\Box\neg$.

□

Proposition 51.23. $\mathbf{K} \vdash \Box\varphi \to (\Diamond(\varphi \to \psi) \to \Diamond\psi)$

Proof.

1.	$\mathbf{K} \vdash \varphi \to (\neg\psi \to \neg(\varphi \to \psi))$	TAUT
2.	$\mathbf{K} \vdash \Box\varphi \to (\Box\neg\psi \to \Box\neg(\varphi \to \psi))$	RK, 1
3.	$\mathbf{K} \vdash \Box\varphi \to (\neg\Box\neg(\varphi \to \psi) \to \neg\Box\neg\psi)$	PL, 2
4.	$\mathbf{K} \vdash \Box\varphi \to (\Diamond(\varphi \to \psi) \to \Diamond\psi)$	$\Diamond$ for $\neg\Box\neg$.

□

Proposition 51.24. $\mathbf{K} \vdash (\Diamond\varphi \vee \Diamond\psi) \to \Diamond(\varphi \vee \psi)$

Proof.

1.	$\mathbf{K} \vdash \neg(\varphi \vee \psi) \to \neg\varphi$	TAUT
2.	$\mathbf{K} \vdash \Box\neg(\varphi \vee \psi) \to \Box\neg\varphi$	RK, 1
3.	$\mathbf{K} \vdash \neg\Box\neg\varphi \to \neg\Box\neg(\varphi \vee \psi)$	PL, 2
4.	$\mathbf{K} \vdash \Diamond\varphi \to \Diamond(\varphi \vee \psi)$	$\Diamond$ for $\neg\Box\neg$
5.	$\mathbf{K} \vdash \Diamond\psi \to \Diamond(\varphi \vee \psi)$	similarly
6.	$\mathbf{K} \vdash (\Diamond\varphi \vee \Diamond\psi) \to \Diamond(\varphi \vee \psi)$	PL, 4, 5.

□

Proposition 51.25. $\mathbf{K} \vdash \Diamond(\varphi \vee \psi) \to (\Diamond\varphi \vee \Diamond\psi)$

Proof.

1.	$\mathbf{K} \vdash \neg\varphi \to (\neg\psi \to \neg(\varphi \vee \psi))$	TAUT
2.	$\mathbf{K} \vdash \Box\neg\varphi \to (\Box\neg\psi \to \Box\neg(\varphi \vee \psi))$	RK
3.	$\mathbf{K} \vdash \Box\neg\varphi \to (\neg\Box\neg(\varphi \vee \psi) \to \neg\Box\neg\psi)$	PL, 2
4.	$\mathbf{K} \vdash \neg\Box\neg(\varphi \vee \psi) \to (\Box\neg\varphi \to \neg\Box\neg\psi)$	PL, 3
5.	$\mathbf{K} \vdash \neg\Box\neg(\varphi \vee \psi) \to (\neg\neg\Box\neg\psi \to \neg\Box\neg\varphi)$	PL, 4
6.	$\mathbf{K} \vdash \Diamond(\varphi \vee \psi) \to (\neg\Diamond\psi \to \Diamond\varphi)$	$\Diamond$ for $\neg\Box\neg$
7.	$\mathbf{K} \vdash \Diamond(\varphi \vee \psi) \to (\Diamond\psi \vee \Diamond\varphi)$	PL, 6.

□

Problem 51.4. Show that the following derivability claims hold:

1. $\mathbf{K} \vdash \Diamond\neg\bot \to (\Box\varphi \to \Diamond\varphi)$;
2. $\mathbf{K} \vdash \Box(\varphi \vee \psi) \to (\Diamond\varphi \vee \Box\psi)$;
3. $\mathbf{K} \vdash (\Diamond\varphi \to \Box\psi) \to \Box(\varphi \to \psi)$.

---

content/normal-modal-logic/axioms-systems/duals.tex

51.7 Dual Formulas

nml:prf:dua:sec
nml:prf:dua:def:duals

Definition 51.26. Each of the formulas T, B, 4, and 5 has a *dual*, denoted by a subscripted diamond, as follows:

$$p \to \Diamond p \tag{$\mathrm{T}_\Diamond$}$$
$$\Diamond\Box p \to p \tag{$\mathrm{B}_\Diamond$}$$
$$\Diamond\Diamond p \to \Diamond p \tag{$4_\Diamond$}$$
$$\Diamond\Box p \to \Box p \tag{$5_\Diamond$}$$

Each of the above dual formulas is obtained from the corresponding formula by substituting $\neg p$ for p, contraposing, replacing $\neg\Box\neg$ by $\Diamond$, and replacing $\neg\Diamond\neg$ by $\Box$. D, i.e., $\Box\varphi \to \Diamond\varphi$ is its own dual in that sense.

nml:prf:dua:prop:dualsys

Proposition 51.27. *For each formula φ in Definition 51.26:* $\mathbf{K}\varphi = \mathbf{K}\varphi_\Diamond$.

Proof. Exercise. □

Problem 51.5. Prove Proposition 51.27.

---

content/normal-modal-logic/axioms-systems/proofs-modal-systems.tex

51.8 Proofs in Modal Systems

nml:prf:prs:sec

We now come to proofs in systems of modal logic other than **K**.

nml:prf:prs:prop:S5facts

Proposition 51.28. *The following provability results obtain:*

1. $\mathbf{KT5} \vdash \mathrm{B}$;
2. $\mathbf{KT5} \vdash 4$;
3. $\mathbf{KDB4} \vdash \mathrm{T}$;
4. $\mathbf{KB4} \vdash 5$;

5. **KB5** $\vdash$ 4;

6. **KT** $\vdash$ D.

nml:prf:prs:
prop:S5facts-KT-D

Proof. We exhibit proofs for each.

1. **KT5** $\vdash$ B:

1.	**KT5** $\vdash \Diamond\varphi \to \Box\Diamond\varphi$	5
2.	**KT5** $\vdash \varphi \to \Diamond\varphi$	$T_\Diamond$
3.	**KT5** $\vdash \varphi \to \Box\Diamond\varphi$	PL, 2, 1.

2. **KT5** $\vdash$ 4:

1.	**KT5** $\vdash \Diamond\Box\varphi \to \Box\Diamond\Box\varphi$	5 with $\Box\varphi$ for p
2.	**KT5** $\vdash \Box\varphi \to \Diamond\Box\varphi$	$T_\Diamond$ with $\Box\varphi$ for p
3.	**KT5** $\vdash \Box\varphi \to \Box\Diamond\Box\varphi$	PL, 2, 1
4.	**KT5** $\vdash \Diamond\Box\varphi \to \Box\varphi$	$5_\Diamond$
5.	**KT5** $\vdash \Box\Diamond\Box\varphi \to \Box\Box\varphi$	RK, 4
6.	**KT5** $\vdash \Box\varphi \to \Box\Box\varphi$	PL, 3, 5.

3. **KDB4** $\vdash$ T:

1.	**KDB4** $\vdash \Diamond\Box\varphi \to \varphi$	$B_\Diamond$
2.	**KDB4** $\vdash \Box\Box\varphi \to \Diamond\Box\varphi$	D with $\Box\varphi$ for p
3.	**KDB4** $\vdash \Box\Box\varphi \to \varphi$	PL, 1, 2
4.	**KDB4** $\vdash \Box\varphi \to \Box\Box\varphi$	4
5.	**KDB4** $\vdash \Box\varphi \to \varphi$	PL, 4, 3.

4. **KB4** $\vdash$ 5:

1.	**KB4** $\vdash \Diamond\varphi \to \Box\Diamond\Diamond\varphi$	B with $\Diamond\varphi$ for p
2.	**KB4** $\vdash \Diamond\Diamond\varphi \to \Diamond\varphi$	$4_\Diamond$
3.	**KB4** $\vdash \Box\Diamond\Diamond\varphi \to \Box\Diamond\varphi$	RK, 2
4.	**KB4** $\vdash \Diamond\varphi \to \Box\Diamond\varphi$	PL, 1, 3.

5. **KB5** $\vdash$ 4:

1.	**KB5** $\vdash \Box\varphi \to \Box\Diamond\Box\varphi$	B with $\Box\varphi$ for p
2.	**KB5** $\vdash \Diamond\Box\varphi \to \Box\varphi$	$5_\Diamond$
3.	**KB5** $\vdash \Box\Diamond\Box\varphi \to \Box\Box\varphi$	RK, 2
4.	**KB5** $\vdash \Box\varphi \to \Box\Box\varphi$	PL, 1, 3.

6. **KT** $\vdash$ D:

1.	**KT** $\vdash \Box\varphi \to \varphi$	T
2.	**KT** $\vdash \varphi \to \Diamond\varphi$	$T_\Diamond$
3.	**KT** $\vdash \Box\varphi \to \Diamond\varphi$	PL, 1, 2

□

Definition 51.29. Following tradition, we define **S4** to be the system **KT4**, and **S5** the system **KTB4**.

The following proposition shows that the classical system **S5** has several equivalent axiomatizations. This should not surprise, as the various combinations of axioms all characterize equivalence relations (see Proposition 50.12).

nml:prf:prs:prop:S5

Proposition 51.30. **KTB4** = **KT5** = **KDB4** = **KDB5**.

Proof. Exercise. □

Problem 51.6. Prove Proposition 51.30.

---

content/normal-modal-logic/axioms-systems/soundness.tex

51.9 Soundness

nml:prf:snd:sec

A derivation system is called sound if everything that can be derived is valid. When considering modal systems, i.e., derivations where in addition to K we can use instances of some formulas $\varphi_1, \ldots, \varphi_n$, we want every derivable formula to be true in any model in which $\varphi_1, \ldots, \varphi_n$ are true.

nml:prf:snd:thm:soundness

Theorem 51.31 (Soundness Theorem). *If every instance of $\varphi_1, \ldots, \varphi_n$ is valid in the classes of models $\mathcal{C}_1, \ldots, \mathcal{C}_n$, respectively, then $\mathbf{K}\varphi_1 \ldots \varphi_n \vdash \psi$ implies that ψ is valid in the class of models $\mathcal{C}_1 \cap \cdots \cap \mathcal{C}_n$.*

Proof. By induction on length of proofs. For brevity, put $\mathcal{C} = \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_n$.

1. Induction Basis: If ψ has a proof of length 1, then it is either a tautological instance, an instance of K, or of DUAL, or an instance of one of $\varphi_1, \ldots, \varphi_n$. In the first case, ψ is valid in $\mathcal{C}$, since tautological instance are valid in *any* class of models, by Proposition 49.15. Similarly in the second case, by Proposition 49.18 and Proposition 49.19. Finally in the third case, since ψ is valid in $\mathcal{C}_i$ and $\mathcal{C} \subseteq \mathcal{C}_i$, we have that ψ is valid in $\mathcal{C}$ as well by Proposition 49.11.

2. Inductive step: Suppose ψ has a proof of length $k > 1$. If ψ is a tautological instance or an instance of one of $\varphi_1, \ldots, \varphi_n$, we proceed as in the previous step. So suppose ψ is obtained by MP from previous formulas $\chi \to \psi$ and χ. Then $\chi \to \psi$ and χ have proofs of length $< k$, and by inductive hypothesis they are valid in $\mathcal{C}$. By Proposition 49.20, ψ is valid in $\mathcal{C}$ as well. Finally suppose ψ is obtained by NEC from χ (so that $\psi = \Box\chi$). By inductive hypothesis, χ is valid in $\mathcal{C}$, and by Proposition 49.12 so is ψ. □

---

content/normal-modal-logic/axioms-systems/systems-distinct.tex

51.10 Showing Systems are Distinct

In section 51.8 we saw how to prove that two systems of modal logic are in fact the same system. Theorem 51.31 allows us to show that two modal systems Σ and Σ' are distinct, by finding a formula φ such that $\Sigma' \vdash \varphi$ that fails in a model of Σ. nml:prf:dis:sec

Proposition 51.32. $\mathbf{KD} \subsetneq \mathbf{KT}$

Proof. This is the syntactic counterpart to the semantic fact that all reflexive relations are serial. To show $\mathbf{KD} \subseteq \mathbf{KT}$ we need to see that $\mathbf{KD} \vdash \psi$ implies $\mathbf{KT} \vdash \psi$, which follows from $\mathbf{KT} \vdash \mathbf{D}$, as shown in Proposition 51.28(6). To show that the inclusion is proper, by Soundness (Theorem 51.31), it suffices to exhibit a model of **KD** where T, i.e., $\Box p \to p$, fails (an easy task left as an exercise), for then by Soundness $\mathbf{KD} \nvdash \Box p \to p$. □

Proposition 51.33. $\mathbf{KB} \neq \mathbf{K4}$.

Proof. We construct a symmetric model where some instance of 4 fails; since obviously the instance is derivable for **K4** but not in **KB**, it will follow $\mathbf{K4} \nsubseteq \mathbf{KB}$. Consider the symmetric model $\mathfrak{M}$ of Figure 51.1. Since the model is symmetric, K and B are true in $\mathfrak{M}$ (by Proposition 49.18 and Theorem 50.1, respectively). However, $\mathfrak{M}, w_1 \nVdash \Box p \to \Box\Box p$. □

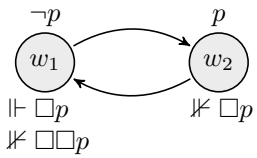


Figure 51.1: A symmetric model falsifying an instance of 4.

Theorem 51.34. $\mathbf{KTB} \nvdash \mathbf{4}$ *and* $\mathbf{KTB} \nvdash \mathbf{5}$. nml:prf:dis:nml:prf:dis:fig:Bnot4thm:KTBnot45

Proof. By Theorem 50.1 we know that all instances of T and B are true in every reflexive symmetric model (respectively). So by soundness, it suffices to find a reflexive symmetric model containing a world at which some instance of 4 fails, and similarly for 5. We use the same model for both claims. Consider the symmetric, reflexive model in Figure 51.2. Then $\mathfrak{M}, w_1 \nVdash \Box p \to \Box\Box p$, so 4 fails at w_1. Similarly, $\mathfrak{M}, w_2 \nVdash \Diamond\neg p \to \Box\Diamond\neg p$, so the instance of 5 with $\varphi = \neg p$ fails at w_2. □

Theorem 51.35. $\mathbf{KD5} \neq \mathbf{KT4} = \mathbf{S4}$. nml:prf:dis:thm:KD5not4

Proof. By Theorem 50.1 we know that all instances of D and 5 are true in all serial euclidean models. So it suffices to find a serial euclidean model containing a world at which some instance of 4 fails. Consider the model of Figure 51.3, and notice that $\mathfrak{M}, w_1 \nVdash \Box p \to \Box\Box p$. □

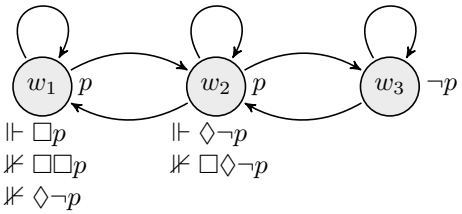


Figure 51.2: The model for Theorem 51.34.

nml:prf:dis:fig:KTBnot45

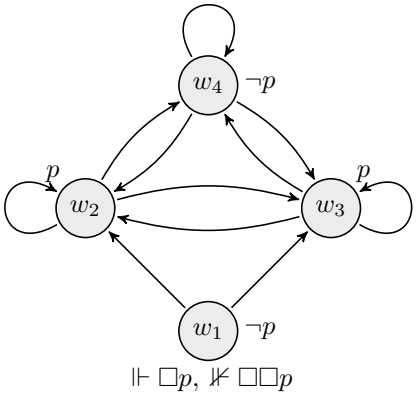


Figure 51.3: The model for Theorem 51.35.

nml:prf:dis:fig:KD5not4

Problem 51.7. Give an alternative proof of Theorem 51.35 using a model with 3 worlds.

Problem 51.8. Provide a single reflexive transitive model showing that both **KT4** $\nvdash$ B and **KT4** $\nvdash$ 5.

---

content/normal-modal-logic/axioms-systems/provability-from-set.tex

51.11 Derivability from a Set of Formulas

nml:prf:prg:sec

In section 51.8 we defined a notion of provability of a formula in a system Σ. We now extend this notion to provability in Σ from formulas in a set Γ.

nml:prf:prg:defn:Gammaproves

Definition 51.36. A formula φ is derivable in a system Σ from a set of formulas Γ, written $\Gamma \vdash_\Sigma \varphi$ if and only if there are $\psi_1, \ldots, \psi_n \in \Gamma$ such that $\Sigma \vdash \psi_1 \to (\psi_2 \to \cdots (\psi_n \to \varphi) \cdots)$.

---

content/normal-modal-logic/axioms-systems/provability-properties.tex

51.12 Properties of Derivability

nml:prf:prp:sec

Proposition 51.37. *Let Σ be a modal system and Γ a set of modal formulas. The following properties hold:* nml:prf:prp:prop:derivabilityfacts

1. Monotonicity*: If $\Gamma \vdash_\Sigma \varphi$ and $\Gamma \subseteq \Delta$ then $\Delta \vdash_\Sigma \varphi$;* nml:prf:prp:prop:derivabilityfacts-monotonicit
2. Reflexivity*: If $\varphi \in \Gamma$ then $\Gamma \vdash_\Sigma \varphi$;* nml:prf:prp:prop:derivabilityfacts-reflexivity
3. Cut*: If $\Gamma \vdash_\Sigma \varphi$ and $\Delta \cup \{\varphi\} \vdash_\Sigma \psi$ then $\Gamma \cup \Delta \vdash_\Sigma \psi$;* nml:prf:prp:prop:derivabilityfacts-cut
4. Deduction theorem*: $\Gamma \cup \{\psi\} \vdash_\Sigma \varphi$ if and only if $\Gamma \vdash_\Sigma \psi \to \varphi$;* nml:prf:prp:prop:derivabilityfacts-deduction
5. *$\Gamma \vdash_\Sigma \varphi_1$ and ... and $\Gamma \vdash_\Sigma \varphi_n$ and $\varphi_1 \to (\varphi_2 \to \cdots (\varphi_n \to \psi) \cdots)$ is a tautological instance, then $\Gamma \vdash_\Sigma \psi$.* nml:prf:prp:prop:derivabilityfacts-ruleT

The proof is an easy exercise. Part (5) of Proposition 51.37 gives us that, for instance, if $\Gamma \vdash_\Sigma \varphi \vee \psi$ and $\Gamma \vdash_\Sigma \neg\varphi$, then $\Gamma \vdash_\Sigma \psi$. Also, in what follows, we write $\Gamma, \varphi \vdash_\Sigma \psi$ instead of $\Gamma \cup \{\varphi\} \vdash_\Sigma \psi$.

Definition 51.38. A set Γ is *deductively closed* relatively to a system Σ if and only if $\Gamma \vdash_\Sigma \varphi$ implies $\varphi \in \Gamma$.

---

content/normal-modal-logic/axioms-systems/consistency.tex

51.13 Consistency

nml:prf:con:sec

Consistency is an important property of sets of formulas. A set of formulas is inconsistent if a contradiction, such as $\bot$, is derivable from it; and otherwise consistent. If a set is inconsistent, its formulas cannot all be true in a model at a world. For the completeness theorem we prove the converse: every consistent set is true at a world in a model, namely in the "canonical model."

Definition 51.39. A set Γ is *consistent* relatively to a system Σ or, as we will say, Σ-consistent, if and only if $\Gamma \nvdash_\Sigma \bot$.

So for instance, the set $\{\Box(p \to q), \Box p, \neg\Box q\}$ is consistent relatively to propositional logic, but not **K**-consistent. Similarly, the set $\{\Diamond p, \Box\Diamond p \to q, \neg q\}$ is not **K5**-consistent.

Proposition 51.40. *Let Γ be a set of formulas. Then:* nml:prf:con:prop:consistencyfacts

1. *Γ is Σ-consistent if and only if there is some formula φ such that $\Gamma \nvdash_\Sigma \varphi$.*
2. *$\Gamma \vdash_\Sigma \varphi$ if and only if $\Gamma \cup \{\neg\varphi\}$ is not Σ-consistent.* nml:prf:con:prop:consistencyfacts-b
3. *If Γ is Σ-consistent, then for any formula φ, either $\Gamma \cup \{\varphi\}$ is Σ-consistent or $\Gamma \cup \{\neg\varphi\}$ is Σ-consistent.* nml:prf:con:prop:consistencyfacts-c

Proof. These facts follow easily using classical propositional logic. We give the argument for (3). Proceed contrapositively and suppose neither $\Gamma \cup \{\varphi\}$ nor $\Gamma \cup \{\neg\varphi\}$ is Σ-consistent. Then by (2), both $\Gamma, \varphi \vdash_\Sigma \bot$ and $\Gamma, \neg\varphi \vdash_\Sigma \bot$. By the deduction theorem $\Gamma \vdash_\Sigma \varphi \to \bot$ and $\Gamma \vdash_\Sigma \neg\varphi \to \bot$. But $(\varphi \to \bot) \to ((\neg\varphi \to \bot) \to \bot)$ is a tautological instance, hence by Proposition 51.37(5), $\Gamma \vdash_\Sigma \bot$. □

Chapter 52

Completeness and Canonical Models

---

content/normal-modal-logic/completeness/introduction.tex

52.1 Introduction

nml:com:int:sec

If Σ is a modal system, then the soundness theorem establishes that if $\Sigma \vdash \varphi$, then φ is valid in any class $\mathcal{C}$ of models in which all instances of all formulas in Σ are valid. In particular that means that if $\mathbf{K} \vdash \varphi$ then φ is true in all models; if $\mathbf{KT} \vdash \varphi$ then φ is true in all reflexive models; if $\mathbf{KD} \vdash \varphi$ then φ is true in all serial models, etc.

Completeness is the converse of soundness: that $\mathbf{K}$ is complete means that if a formula φ is valid, $\vdash \varphi$, for instance. Proving completeness is a lot harder to do than proving soundness. It is useful, first, to consider the contrapositive: $\mathbf{K}$ is complete iff whenever $\nvdash \varphi$, there is a countermodel, i.e., a model $\mathfrak{M}$ such that $\mathfrak{M} \nVdash \varphi$. Equivalently (negating φ), we could prove that whenever $\nvdash \neg\varphi$, there is a model of φ. In the construction of such a model, we can use information contained in φ. When we find models for specific formulas we often do the same: e.g., if we want to find a countermodel to $p \to \Box q$, we know that it has to contain a world where p is true and $\Box q$ is false. And a world where $\Box q$ is false means there has to be a world accessible from it where q is false. And that's all we need to know: which worlds make the propositional variables true, and which worlds are accessible from which worlds.

In the case of proving completeness, however, we don't have a specific formula φ for which we are constructing a model. We want to establish that a

model exists for every φ such that $\nvdash_\Sigma \neg\varphi$. This is a minimal requirement, since *if* $\vdash_\Sigma \neg\varphi$, by soundness, there is no model for φ (in which Σ is true). Now note that $\nvdash_\Sigma \neg\varphi$ iff φ is Σ-consistent. (Recall that $\Sigma \nvdash_\Sigma \neg\varphi$ and $\varphi \nvdash_\Sigma \bot$ are equivalent.) So our task is to construct a model for every Σ-consistent formula.

The trick we'll use is to find a Σ-consistent set of formulas that contains φ, but also other formulas which tell us what the world that makes φ true has to look like. Such sets are *complete* Σ-consistent sets. It's not enough to construct a model with a single world to make φ true, it will have to contain multiple worlds and an accessibility relation. The complete Σ-consistent set containing φ will also contain other formulas of the form $\Box\psi$ and $\Diamond\chi$. In all accessible worlds, ψ has to be true; in at least one, χ has to be true. In order to accomplish this, we'll simply take *all* possible complete Σ-consistent sets as the basis for the set of worlds. A tricky part will be to figure out when a complete Σ-consistent set should count as being accessible from another in our model.

We'll show that in the model so defined, φ is true at a world—which is also a complete Σ-consistent set—iff φ is an element of that set. If φ is Σ-consistent, it will be an element of at least one complete Σ-consistent set (a fact we'll prove), and so there will be a world where φ is true. So we will have a single model where every Σ-consistent formula φ is true at some world. This single model is the *canonical* model for Σ.

---

content/normal-modal-logic/completeness/complete-consistent-sets.tex

52.2 Complete Σ-Consistent Sets

nml:com:ccs: sec

Suppose Σ is a set of modal formulas—think of them as the axioms or defining principles of a normal modal logic. A set Γ is Σ-consistent iff $\Gamma \nvdash_\Sigma \bot$, i.e., if there is no derivation of $\varphi_1 \to (\varphi_2 \to \cdots (\varphi_n \to \bot)\dots)$ from Σ, where each $\varphi_i \in \Gamma$. We will construct a "canonical" model in which each world is taken to be a special kind of Σ-consistent set: one which is not just Σ-consistent, but maximally so, in the sense that it settles the truth value of every modal formula: for every φ, either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$:

Definition 52.1. A set Γ is *complete Σ-consistent* if and only if it is Σ-consistent and for every φ, either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$.

Complete Σ-consistent sets Γ have a number of useful properties. For one, they are deductively closed, i.e., if $\Gamma \vdash_\Sigma \varphi$ then $\varphi \in \Gamma$. This means in particular that every instance of a formula $\varphi \in \Sigma$ is also $\in \Gamma$. Moreover, membership in Γ mirrors the truth conditions for the propositional connectives. This will be important when we define the "canonical model."

Proposition 52.2. *Suppose Γ is complete Σ-consistent. Then:* nml:com:ccs: prop:ccs-properties

1. *Γ is deductively closed in Σ.* nml:com:ccs: prop:ccs-closed

nml:com:ccs:prop:ccs-sigma
2. $\Sigma \subseteq \Gamma$.

nml:com:ccs:prop:ccs-lfalse
3. $\bot \notin \Gamma$

nml:com:ccs:prop:ccs-ltrue
4. $\top \in \Gamma$

nml:com:ccs:prop:ccs-lnot
5. $\neg\varphi \in \Gamma$ *if and only if* $\varphi \notin \Gamma$.

nml:com:ccs:prop:ccs-land
6. $\varphi \wedge \psi \in \Gamma$ *iff* $\varphi \in \Gamma$ *and* $\psi \in \Gamma$

nml:com:ccs:prop:ccs-lor
7. $\varphi \vee \psi \in \Gamma$ *iff* $\varphi \in \Gamma$ *or* $\psi \in \Gamma$

nml:com:ccs:prop:ccs-lif
8. $\varphi \to \psi \in \Gamma$ *iff* $\varphi \notin \Gamma$ *or* $\psi \in \Gamma$

nml:com:ccs:prop:ccs-liff
9. $\varphi \leftrightarrow \psi \in \Gamma$ *iff either* $\varphi \in \Gamma$ *and* $\psi \in \Gamma$, *or* $\varphi \notin \Gamma$ *and* $\psi \notin \Gamma$

Proof. 1. Suppose $\Gamma \vdash_\Sigma \varphi$ but $\varphi \notin \Gamma$. Then since Γ is complete Σ-consistent, $\neg\varphi \in \Gamma$. This would make Γ inconsistent, since $\varphi, \neg\varphi \vdash_\Sigma \bot$.

2. If $\varphi \in \Sigma$ then $\Gamma \vdash_\Sigma \varphi$, and $\varphi \in \Gamma$ by deductive closure, i.e., case (1).

3. If $\bot \in \Gamma$, then $\Gamma \vdash_\Sigma \bot$, so Γ would be Σ-inconsistent.

4. $\Gamma \vdash_\Sigma \top$, so $\top \in \Gamma$ by deductive closure, i.e., case (1).

5. If $\neg\varphi \in \Gamma$, then by consistency $\varphi \notin \Gamma$; and if $\varphi \notin \Gamma$ then $\varphi \in \Gamma$ since Γ is complete Σ-consistent.

6. Suppose $\varphi \wedge \psi \in \Gamma$. Since $(\varphi \wedge \psi) \to \varphi$ is a tautological instance, $\varphi \in \Gamma$ by deductive closure, i.e., case (1). Similarly for $\psi \in \Gamma$. On the other hand, suppose both $\varphi \in \Gamma$ and $\psi \in \Gamma$. Then deductive closure implies $(\varphi \wedge \psi) \in \Gamma$, since $\varphi \to (\psi \to (\varphi \wedge \psi))$ is a tautological instance.

7. Suppose $\varphi \vee \psi \in \Gamma$, and $\varphi \notin \Gamma$ and $\psi \notin \Gamma$. Since Γ is complete Σ-consistent, $\neg\varphi \in \Gamma$ and $\neg\psi \in \Gamma$. Then $\neg(\varphi \vee \psi) \in \Gamma$ since $\neg\varphi \to (\neg\psi \to \neg(\varphi \vee \psi))$ is a tautological instance. This would mean that Γ is Σ-inconsistent, a contradiction.

8. Suppose $\varphi \to \psi \in \Gamma$ and $\varphi \in \Gamma$; then $\Gamma \vdash_\Sigma \psi$, whence $\psi \in \Gamma$ by deductive closure. Conversely, if $\varphi \to \psi \notin \Gamma$ then since Γ is complete Σ-consistent, $\neg(\varphi \to \psi) \in \Gamma$. Since $\neg(\varphi \to \psi) \to \varphi$ is a tautological instance, $\varphi \in \Gamma$ by deductive closure. Since $\neg(\varphi \to \psi) \to \neg\psi$ is a tautological instance, $\neg\psi \in \Gamma$. Then $\psi \notin \Gamma$ since Γ is Σ-consistent.

9. Suppose $\varphi \leftrightarrow \psi \in \Gamma$. If $\varphi \in \Gamma$, then $\psi \in \Gamma$, since $(\varphi \leftrightarrow \psi) \to (\varphi \to \psi)$ is a tautological instance. Similarly, if $\psi \in \Gamma$, then $\varphi \in \Gamma$. So either both $\varphi \in \Gamma$ and $\psi \in \Gamma$, or neither $\varphi \in \Gamma$ nor $\psi \in \Gamma$.

   Conversely, suppose $\varphi \to \psi \notin \Gamma$. Since Γ is complete Σ-consistent, $\neg(\varphi \leftrightarrow \psi) \in \Gamma$. Since $\neg(\varphi \leftrightarrow \psi) \to (\varphi \to \neg\psi)$ is a tautological instance, if $\varphi \in \Gamma$ then $\neg\psi \in \Gamma$, and since Γ is Σ-consistent, $\psi \notin \Gamma$. Similarly, if $\psi \in \Gamma$ then $\varphi \notin \Gamma$. So neither $\varphi \in \Gamma$ and $\psi \in \Gamma$, nor $\varphi \notin \Gamma$ and $\psi \notin \Gamma$. □

Problem 52.1. Complete the proof of Proposition 52.2.

---

content/normal-modal-logic/completeness/lindenbaums-lemma.tex

52.3 Lindenbaum's Lemma

nml:com:lin:sec

Lindenbaum's Lemma establishes that every Σ-consistent set of formulas is contained in at least one *complete* Σ-consistent set. Our construction of the canonical model will show that for each complete Σ-consistent set Δ, there is a world in the canonical model where all and only the formulas in Δ are true. So Lindenbaum's Lemma guarantees that every Σ-consistent set is true at some world in the canonical model.

nml:com:lin:thm:lindenbaum

Theorem 52.3 (Lindenbaum's Lemma). *If Γ is Σ-consistent then there is a complete Σ-consistent set Δ extending Γ.*

Proof. Let $\varphi_0, \varphi_1, \ldots$ be an exhaustive listing of all formulas of the language (repetitions are allowed). For instance, start by listing p_0, and at each stage $n \geq 1$ list the finitely many formulas of length n using only variables among $p_0, \ldots, p_n$. We define sets of formulas Δ_n by induction on n, and we then set $\Delta = \bigcup_n \Delta_n$. We first put $\Delta_0 = \Gamma$. Supposing that Δ_n has been defined, we define Δ_{n+1} by:

$$\Delta_{n+1} = \begin{cases} \Delta_n \cup \{\varphi_n\}, & \text{if } \Delta_n \cup \{\varphi_n\} \text{ is } \Sigma\text{-consistent;} \\ \Delta_n \cup \{\neg\varphi_n\}, & \text{otherwise.} \end{cases}$$

Now let $\Delta = \bigcup_{n=0}^{\infty} \Delta_n$.

We have to show that this definition actually yields a set Δ with the required properties, i.e., $\Gamma \subseteq \Delta$ and Δ is complete Σ-consistent.

It's obvious that $\Gamma \subseteq \Delta$, since $\Delta_0 \subseteq \Delta$ by construction, and $\Delta_0 = \Gamma$. In fact, $\Delta_n \subseteq \Delta$ for all n, since Δ is the union of all Δ_n. (Since in each step of the construction, we add a formula to the set already constructed, $\Delta_n \subseteq \Delta_{n+1}$, so since $\subseteq$ is transitive, $\Delta_n \subseteq \Delta_m$ whenever $n \leq m$.) At each stage of the construction, we either add φ_n or $\neg\varphi_n$, and every formula appears (at least once) in the list of all φ_n. So, for every φ either $\varphi \in \Delta$ or $\neg\varphi \in \Delta$, so Δ is complete by definition.

Finally, we have to show, that Δ is Σ-consistent. To do this, we show that (a) if Δ were Σ-inconsistent, then some Δ_n would be Σ-inconsistent, and (b) all Δ_n are Σ-consistent.

So suppose Δ were Σ-inconsistent. Then $\Delta \vdash_\Sigma \bot$, i.e., there are $\varphi_1, \ldots, \varphi_k \in \Delta$ such that $\Sigma \vdash \varphi_1 \to (\varphi_2 \to \cdots (\varphi_k \to \bot) \ldots)$. Since $\Delta = \bigcup_{n=0}^{\infty} \Delta_n$, each $\varphi_i \in \Delta_{n_i}$ for some n_i. Let n be the largest of these. Since $n_i \leq n$, $\Delta_{n_i} \subseteq \Delta_n$. So, all φ_i are in some Δ_n. This would mean $\Delta_n \vdash_\Sigma \bot$, i.e., Δ_n is Σ-inconsistent.

To show that each Δ_n is Σ-consistent, we use a simple induction on n. $\Delta_0 = \Gamma$, and we assumed Γ was Σ-consistent. So the claim holds for $n = 0$. Now suppose it holds for n, i.e., Δ_n is Σ-consistent. Δ_{n+1} is either $\Delta_n \cup \{\varphi_n\}$ if that is Σ-consistent, otherwise it is $\Delta_n \cup \{\neg\varphi_n\}$. In the first case, Δ_{n+1} is clearly Σ-consistent. However, by Proposition 51.40(3), either $\Delta_n \cup \{\varphi_n\}$ or $\Delta_n \cup \{\neg\varphi_n\}$ is consistent, so Δ_{n+1} is consistent in the other case as well. □

nml:com:lin:cor:provability-characterization **Corollary 52.4.** *$\Gamma \vdash_\Sigma \varphi$ if and only if $\varphi \in \Delta$ for each complete Σ-consistent set Δ extending Γ (including when $\Gamma = \emptyset$, in which case we get another characterization of the modal system Σ.)*

Proof. Suppose $\Gamma \vdash_\Sigma \varphi$, and let Δ be any complete Σ-consistent set extending Γ. If $\varphi \notin \Delta$ then by maximality $\neg\varphi \in \Delta$ and so $\Delta \vdash_\Sigma \varphi$ (by monotonicity) and $\Delta \vdash_\Sigma \neg\varphi$ (by reflexivity), and so Δ is inconsistent. Conversely if $\Gamma \nvdash_\Sigma \varphi$, then $\Gamma \cup \{\neg\varphi\}$ is Σ-consistent, and by Lindenbaum's Lemma there is a complete consistent set Δ extending $\Gamma \cup \{\neg\varphi\}$. By consistency, $\varphi \notin \Delta$. □

---

content/normal-modal-logic/completeness/modalities-ccs.tex

52.4 Modalities and Complete Consistent Sets

nml:com:mod:sec When we construct a model $\mathfrak{M}^\Sigma$ whose set of worlds is given by the complete Σ-consistent sets Δ in some normal modal logic Σ, we will also need to define an accessibility relation R^Σ between such "worlds." We want it to be the case that the accessibility relation (and the assignment V^Σ) are defined in such a way that $\mathfrak{M}^\Sigma, \Delta \Vdash \varphi$ iff $\varphi \in \Delta$. How should we do this? explanation

Once the accessibility relation is defined, the definition of truth at a world ensures that $\mathfrak{M}^\Sigma, \Delta \Vdash \Box\varphi$ iff $\mathfrak{M}^\Sigma, \Delta' \Vdash \varphi$ for all Δ' such that $R^\Sigma\Delta\Delta'$. The proof that $\mathfrak{M}^\Sigma, \Delta \Vdash \varphi$ iff $\varphi \in \Delta$ requires that this is true in particular for formulas starting with a modal operator, i.e., $\mathfrak{M}^\Sigma, \Delta \Vdash \Box\varphi$ iff $\Box\varphi \in \Delta$. Combining this requirement with the definition of truth at a world for $\Box\varphi$ yields:

$$\Box\varphi \in \Delta \text{ iff } \varphi \in \Delta' \text{ for all } \Delta' \text{ with } R^\Sigma\Delta\Delta'$$

Consider the left-to-right direction: it says that if $\Box\varphi \in \Delta$, then $\varphi \in \Delta'$ for any φ and any Δ' with $R^\Sigma\Delta\Delta'$. If we stipulate that $R^\Sigma\Delta\Delta'$ iff $\varphi \in \Delta'$ for all $\Box\varphi \in \Delta$, then this holds. We can write the condition on the right of the "iff" more compactly as: $\{\varphi : \Box\varphi \in \Delta\} \subseteq \Delta'$.

So the question is: does this definition of R^Σ in fact guarantee that $\Box\varphi \in \Delta$ iff $\mathfrak{M}^\Sigma, \Delta \Vdash \Box\varphi$? Does it also guarantee that $\Diamond\varphi \in \Delta$ iff $\mathfrak{M}^\Sigma, \Delta \Vdash \Diamond\varphi$? The next few results will establish this.

Definition 52.5. If Γ is a set of formulas, let

$$\Box\Gamma = \{\Box\psi : \psi \in \Gamma\}$$
$$\Diamond\Gamma = \{\Diamond\psi : \psi \in \Gamma\}$$

and

$$\Box^{-1}\Gamma = \{\psi : \Box\psi \in \Gamma\}$$
$$\Diamond^{-1}\Gamma = \{\psi : \Diamond\psi \in \Gamma\}$$

In other words, $\Box\Gamma$ is Γ with $\Box$ in front of every formula in Γ; $\Box^{-1}\Gamma$ is all the $\Box$'ed formulas of Γ with the initial $\Box$'s removed. This definition is not terribly important on its own, but will simplify the notation considerably.

Note that $\Box\Box^{-1}\Gamma \subseteq \Gamma$:

$$\Box\Box^{-1}\Gamma = \{\Box\psi : \Box\psi \in \Gamma\}$$

i.e., it's just the set of all those formulas of Γ that start with $\Box$.

Lemma 52.6. *If $\Gamma \vdash_\Sigma \varphi$ then $\Box\Gamma \vdash_\Sigma \Box\varphi$.* *nml:com:mod: lem:box1*

Proof. If $\Gamma \vdash_\Sigma \varphi$ then there are $\psi_1, \ldots, \psi_k \in \Gamma$ such that $\Sigma \vdash \psi_1 \to (\psi_2 \to \cdots (\psi_n \to \varphi)\cdots)$. Since Σ is normal, by rule RK, $\Sigma \vdash \Box\psi_1 \to (\Box\psi_2 \to \cdots (\Box\psi_n \to \Box\varphi)\cdots)$, where obviously $\Box\psi_1, \ldots, \Box\psi_k \in \Box\Gamma$. Hence, by definition, $\Box\Gamma \vdash_\Sigma \Box\varphi$. □

Lemma 52.7. *If $\Box^{-1}\Gamma \vdash_\Sigma \varphi$ then $\Gamma \vdash_\Sigma \Box\varphi$.* *nml:com:mod: lem:box2*

Proof. Suppose $\Box^{-1}\Gamma \vdash_\Sigma \varphi$; then by Lemma 52.6, $\Box\Box^{-1}\Gamma \vdash \Box\varphi$. But since $\Box\Box^{-1}\Gamma \subseteq \Gamma$, also $\Gamma \vdash_\Sigma \Box\varphi$ by monotonicity. □

Proposition 52.8. *If Γ is complete Σ-consistent, then $\Box\varphi \in \Gamma$ if and only if for every complete Σ-consistent Δ such that $\Box^{-1}\Gamma \subseteq \Delta$, it holds that $\varphi \in \Delta$.* *nml:com:mod: prop:box*

Proof. Suppose Γ is complete Σ-consistent. The "only if" direction is easy: Suppose $\Box\varphi \in \Gamma$ and that $\Box^{-1}\Gamma \subseteq \Delta$. Since $\Box\varphi \in \Gamma$, $\varphi \in \Box^{-1}\Gamma \subseteq \Delta$, so $\varphi \in \Delta$.

For the "if" direction, we prove the contrapositive: Suppose $\Box\varphi \notin \Gamma$. Since Γ is complete Σ-consistent, it is deductively closed, and hence $\Gamma \nvdash_\Sigma \Box\varphi$. By Lemma 52.7, $\Box^{-1}\Gamma \nvdash_\Sigma \varphi$. By Proposition 51.40(2), $\Box^{-1}\Gamma \cup \{\neg\varphi\}$ is Σ-consistent. By Lindenbaum's Lemma, there is a complete Σ-consistent set Δ such that $\Box^{-1}\Gamma \cup \{\neg\varphi\} \subseteq \Delta$. By consistency, $\varphi \notin \Delta$. □

Lemma 52.9. *Suppose Γ and Δ are complete Σ-consistent. Then $\Box^{-1}\Gamma \subseteq \Delta$ if and only if $\Diamond\Delta \subseteq \Gamma$.* *nml:com:mod: lem:box-iff-diamond*

Proof. "Only if" direction: Assume $\Box^{-1}\Gamma \subseteq \Delta$ and suppose $\Diamond\varphi \in \Diamond\Delta$ (i.e., $\varphi \in \Delta$). In order to show $\Diamond\varphi \in \Gamma$, it suffices to show $\Box\neg\varphi \notin \Gamma$, for then by maximality, $\neg\Box\neg\varphi \in \Gamma$. Now, if $\Box\neg\varphi \in \Gamma$ then by hypothesis $\neg\varphi \in \Delta$, against the consistency of Δ (since $\varphi \in \Delta$). Hence $\Box\neg\varphi \notin \Gamma$, as required.

"If" direction: Assume $\Diamond\Delta \subseteq \Gamma$. We argue contrapositively: suppose $\varphi \notin \Delta$ in order to show $\Box\varphi \notin \Gamma$. If $\varphi \notin \Delta$ then by maximality $\neg\varphi \in \Delta$ and so by hypothesis $\Diamond\neg\varphi \in \Gamma$. But in a normal modal logic $\Diamond\neg\varphi$ is equivalent to $\neg\Box\varphi$, and if the latter is in Γ, by consistency $\Box\varphi \notin \Gamma$, as required. □

nml:com:mod: prop:diamond **Proposition 52.10.** *If Γ is complete Σ-consistent, then $\Diamond\varphi \in \Gamma$ if and only if for some complete Σ-consistent Δ such that $\Diamond\Delta \subseteq \Gamma$, it holds that $\varphi \in \Delta$.*

Proof. Suppose Γ is complete Σ-consistent. $\Diamond\varphi \in \Gamma$ iff $\neg\Box\neg\varphi \in \Gamma$ by DUAL and closure. $\neg\Box\neg\varphi \in \Gamma$ iff $\Box\neg\varphi \notin \Gamma$ by Proposition 52.2(5) since Γ is complete Σ-consistent. By Proposition 52.8, $\Box\neg\varphi \notin \Gamma$ iff, for some complete Σ-consistent Δ with $\Box^{-1}\Gamma \subseteq \Delta$, $\neg\varphi \notin \Delta$. Now consider any such Δ. By Lemma 52.9, $\Box^{-1}\Gamma \subseteq \Delta$ iff $\Diamond\Delta \subseteq \Gamma$. Also, $\neg\varphi \notin \Delta$ iff $\varphi \in \Delta$ by Proposition 52.2(5). So $\Diamond\varphi \in \Gamma$ iff, for some complete Σ-consistent Δ with $\Diamond\Delta \subseteq \Gamma$, $\varphi \in \Delta$. □

Problem 52.2. Show that if Γ is complete Σ-consistent, then $\Diamond\varphi \in \Gamma$ if and only if there is a complete Σ-consistent Δ such that $\Box^{-1}\Gamma \subseteq \Delta$ and $\varphi \in \Delta$. *Do this without using Lemma 52.9.*

---

content/normal-modal-logic/completeness/canonical-models.tex

52.5 Canonical Models

nml:com:cmd: sec The *canonical model* for a modal system Σ is a specific model $\mathfrak{M}^\Sigma$ in which the worlds are all complete Σ-consistent sets. Its accessibility relation R^Σ and valuation V^Σ are defined so as to guarantee that the formulas true at a world Δ are exactly the formulas making up Δ.

Definition 52.11. Let Σ be a normal modal logic. The *canonical model* for Σ is $\mathfrak{M}^\Sigma = \langle W^\Sigma, R^\Sigma, V^\Sigma\rangle$, where:

1. $W^\Sigma = \{\Delta : \Delta \text{ is complete } \Sigma\text{-consistent}\}$.
2. $R^\Sigma\Delta\Delta'$ holds if and only if $\Box^{-1}\Delta \subseteq \Delta'$.
3. $V^\Sigma(p) = \{\Delta : p \in \Delta\}$.

---

content/normal-modal-logic/completeness/truth-lemma.tex

52.6 The Truth Lemma

nml:com:tru: sec The canonical model $\mathfrak{M}^\Sigma$ is defined in such a way that $\mathfrak{M}^\Sigma, \Delta \Vdash \varphi$ iff $\varphi \in \Delta$. For propositional variables, the definition of V^Σ yields this directly. We have to verify that the equivalence holds for all formulas, however. We do this by induction. The inductive step involves proving the equivalence for formulas involving propositional operators (where we have to use Proposition 52.2) and the modal operators (where we invoke the results of section 52.4).

nml:com:tru: prop:truthlemma **Proposition 52.12 (Truth Lemma).** *For every formula φ, $\mathfrak{M}^\Sigma, \Delta \Vdash \varphi$ if and only if $\varphi \in \Delta$.*

Proof. By induction on φ.

1. $\varphi \equiv \bot$: $\mathfrak{M}^\Sigma, \Delta \nVdash \bot$ by Definition 49.6, and $\bot \notin \Delta$ by Proposition 52.2(3).

2. $\varphi \equiv \top$: $\mathfrak{M}^\Sigma, \Delta \Vdash \top$ by Definition 49.6, and $\top \in \Delta$ by Proposition 52.2(4).

3. $\varphi \equiv p$: $\mathfrak{M}^\Sigma, \Delta \Vdash p$ iff $\Delta \in V^\Sigma(p)$ by Definition 49.6. Also, $\Delta \in V^\Sigma(p)$ iff $p \in \Delta$ by definition of V^Σ.

4. $\varphi \equiv \neg\psi$: $\mathfrak{M}^\Sigma, \Delta \Vdash \neg\psi$ iff $\mathfrak{M}^\Sigma, \Delta \nVdash \psi$ (Definition 49.6) iff $\psi \notin \Delta$ (by inductive hypothesis) iff $\neg\psi \in \Delta$ (by Proposition 52.2(5)).

5. $\varphi \equiv \psi \wedge \chi$: $\mathfrak{M}^\Sigma, \Delta \Vdash \psi \wedge \chi$ iff $\mathfrak{M}^\Sigma, \Delta \Vdash \psi$ and $\mathfrak{M}^\Sigma, \Delta \Vdash \chi$ (by Definition 49.6) iff $\psi \in \Delta$ and $\chi \in \Delta$ (by inductive hypothesis) iff $\psi \wedge \chi \in \Delta$ (by Proposition 52.2(6)).

6. $\varphi \equiv \psi \vee \chi$: $\mathfrak{M}^\Sigma, \Delta \Vdash \psi \vee \chi$ iff $\mathfrak{M}^\Sigma, \Delta \Vdash \psi$ or $\mathfrak{M}^\Sigma, \Delta \Vdash \chi$ (by Definition 49.6) iff $\psi \in \Delta$ or $\chi \in \Delta$ (by inductive hypothesis) iff $\psi \vee \chi \in \Delta$ (by Proposition 52.2(7)).

7. $\varphi \equiv \psi \to \chi$: $\mathfrak{M}^\Sigma, \Delta \Vdash \psi \to \chi$ iff $\mathfrak{M}^\Sigma, \Delta \nVdash \psi$ or $\mathfrak{M}^\Sigma, \Delta \Vdash \chi$ (by Definition 49.6) iff $\psi \notin \Delta$ or $\chi \in \Delta$ (by inductive hypothesis) iff $\psi \to \chi \in \Delta$ (by Proposition 52.2(8)).

8. $\varphi \equiv \psi \leftrightarrow \chi$: $\mathfrak{M}^\Sigma, \Delta \Vdash \psi \leftrightarrow \chi$ iff either $\mathfrak{M}^\Sigma, \Delta \Vdash \psi$ and $\mathfrak{M}^\Sigma, \Delta \Vdash \chi$ or $\mathfrak{M}^\Sigma, \Delta \nVdash \psi$ and $\mathfrak{M}^\Sigma, \Delta \nVdash \chi$ (by Definition 49.6) iff either $\psi \in \Delta$ and $\chi \in \Delta$ or $\psi \notin \Delta$ and $\chi \notin \Delta$ (by inductive hypothesis) iff $\psi \leftrightarrow \chi \in \Delta$ (by Proposition 52.2(9)).

9. $\varphi \equiv \Box\psi$: First suppose that $\mathfrak{M}^\Sigma, \Delta \Vdash \Box\psi$. By Definition 49.6, for every Δ' such that $R^\Sigma \Delta\Delta'$, $\mathfrak{M}^\Sigma, \Delta' \Vdash \psi$. By inductive hypothesis, for every Δ' such that $R^\Sigma \Delta\Delta'$, $\psi \in \Delta'$. By definition of R^Σ, for every Δ' such that $\Box^{-1}\Delta \subseteq \Delta'$, $\psi \in \Delta'$. By Proposition 52.8, $\Box\psi \in \Delta$.

   Now assume $\Box\psi \in \Delta$. Let $\Delta' \in W^\Sigma$ be such that $R^\Sigma \Delta\Delta'$, i.e., $\Box^{-1}\Delta \subseteq \Delta'$. Since $\Box\psi \in \Delta$, $\psi \in \Box^{-1}\Delta$. Consequently, $\psi \in \Delta'$. By inductive hypothesis, $\mathfrak{M}^\Sigma, \Delta' \Vdash \psi$. Since Δ' is arbitrary with $R^\Sigma \Delta\Delta'$, for all $\Delta' \in W^\Sigma$ such that $R^\Sigma \Delta\Delta'$, $\mathfrak{M}^\Sigma, \Delta' \Vdash \psi$. By Definition 49.6, $\mathfrak{M}^\Sigma, \Delta \Vdash \Box\psi$.

10. $\varphi \equiv \Diamond\psi$: First suppose that $\mathfrak{M}^\Sigma, \Delta \Vdash \Diamond\psi$. By Definition 49.6, for some Δ' such that $R^\Sigma \Delta\Delta'$, $\mathfrak{M}^\Sigma, \Delta' \Vdash \psi$. By inductive hypothesis, for some Δ' such that $R^\Sigma \Delta\Delta'$, $\psi \in \Delta'$. By definition of R^Σ, for some Δ' such that $\Box^{-1}\Delta \subseteq \Delta'$, $\psi \in \Delta'$. By Proposition 52.10, for some Δ' such that $\Diamond\Delta' \subseteq \Delta$, $\psi \in \Delta'$. Since $\psi \in \Delta'$, $\Diamond\psi \in \Diamond\Delta'$, so $\Diamond\psi \in \Delta$.

    Now assume $\Diamond\psi \in \Delta$. By Proposition 52.10, there is a complete Σ-consistent $\Delta' \in W^\Sigma$ such that $\Diamond\Delta' \subseteq \Delta$ and $\psi \in \Delta'$. By Lemma 52.9, there is a $\Delta' \in W^\Sigma$ such that $\Box^{-1}\Delta \subseteq \Delta'$, and $\psi \in \Delta'$. By definition of R^Σ, $R^\Sigma \Delta\Delta'$, so there is a $\Delta' \in W^\Sigma$ such that $R^\Sigma \Delta\Delta'$ and $\psi \in \Delta'$. By Definition 49.6, $\mathfrak{M}^\Sigma, \Delta \Vdash \Diamond\psi$. □

Problem 52.3. Complete the proof of Proposition 52.12.

---

content/normal-modal-logic/completeness/completeness-K.tex

52.7 Determination and Completeness for K

nml:com:cmk:sec

We are now prepared to use the canonical model to establish completeness. Completeness follows from the fact that the formulas true in the canonical model for Σ are exactly the Σ-derivable ones. Models with this property are said to *determine* Σ.

Definition 52.13. A model $\mathfrak{M}$ *determines* a normal modal logic Σ precisely when $\mathfrak{M} \Vdash \varphi$ if and only if $\Sigma \vdash \varphi$, for all formulas φ.

nml:com:cmk:thm:determination

Theorem 52.14 (Determination). *$\mathfrak{M}^\Sigma \Vdash \varphi$ if and only if $\Sigma \vdash \varphi$.*

Proof. If $\mathfrak{M}^\Sigma \Vdash \varphi$, then for every complete Σ-consistent Δ, we have $\mathfrak{M}^\Sigma, \Delta \Vdash \varphi$. Hence, by the Truth Lemma, $\varphi \in \Delta$ for every complete Σ-consistent Δ, whence by Corollary 52.4 (with $\Gamma = \emptyset$), $\Sigma \vdash \varphi$.

Conversely, if $\Sigma \vdash \varphi$ then by Proposition 52.2(1), every complete Σ-consistent Δ contains φ, and hence by the Truth Lemma, $\mathfrak{M}^\Sigma, \Delta \Vdash \varphi$ for every $\Delta \in W^\Sigma$, i.e., $\mathfrak{M}^\Sigma \Vdash \varphi$. □

Since the canonical model for **K** determines **K**, we immediately have completeness of **K** as a corollary:

nml:com:cmk:cor:Kcomplete

Corollary 52.15. *The basic modal logic* **K** *is complete with respect to the class of all models, i.e., if $\vDash \varphi$ then $\mathbf{K} \vdash \varphi$.*

Proof. Contrapositively, if $\mathbf{K} \nvdash \varphi$ then by Determination $\mathfrak{M}^\mathbf{K} \nVdash \varphi$ and hence φ is not valid. □

For the general case of completeness of a system Σ with respect to a class of models, e.g., of **KTB4** with respect to the class of reflexive, symmetric, transitive models, determination alone is not enough. We must also show that the canonical model for the system Σ is a member of the class, which does not follow obviously from the canonical model construction—nor is it always true!

---

content/normal-modal-logic/completeness/frame-completeness.tex

52.8 Frame Completeness

nml:com:fra:sec

The completeness theorem for **K** can be extended to other modal systems, once we show that the canonical model for a given logic has the corresponding frame property.

Theorem 52.16. *If a normal modal logic Σ contains one of the formulas on the left-hand side of Table 52.1, then the canonical model for Σ has the corresponding property on the right-hand side.* nml:com:fra: thm:completeframeprops

If Σ contains ...	*... the canonical model for Σ is:*
D: $\Box\varphi \to \Diamond\varphi$	serial;
T: $\Box\varphi \to \varphi$	reflexive;
B: $\varphi \to \Box\Diamond\varphi$	symmetric;
4: $\Box\varphi \to \Box\Box\varphi$	transitive;
5: $\Diamond\varphi \to \Box\Diamond\varphi$	euclidean.

Table 52.1: Basic correspondence facts.

Proof. We take each of these up in turn. nml:com:fra: tab:correspondencetable

Suppose Σ contains D, and let $\Delta \in W^\Sigma$; we need to show that there is a Δ' such that $R^\Sigma\Delta\Delta'$. It suffices to show that $\Box^{-1}\Delta$ is Σ-consistent, for then by Lindenbaum's Lemma, there is a complete Σ-consistent set $\Delta' \supseteq \Box^{-1}\Delta$, and by definition of R^Σ we have $R^\Sigma\Delta\Delta'$. So, suppose for contradiction that $\Box^{-1}\Delta$ is *not* Σ-consistent, i.e., $\Box^{-1}\Delta \vdash_\Sigma \bot$. By Lemma 52.7, $\Delta \vdash_\Sigma \Box\bot$, and since Σ contains D, also $\Delta \vdash_\Sigma \Diamond\bot$. But Σ is normal, so $\Sigma \vdash \neg\Diamond\bot$ (Proposition 51.7), whence also $\Delta \vdash_\Sigma \neg\Diamond\bot$, against the consistency of Δ.

Now suppose Σ contains T, and let $\Delta \in W^\Sigma$. We want to show $R^\Sigma\Delta\Delta$, i.e., $\Box^{-1}\Delta \subseteq \Delta$. But if $\Box\varphi \in \Delta$ then by T also $\varphi \in \Delta$, as desired.

Now suppose Σ contains B, and suppose $R^\Sigma\Delta\Delta'$ for $\Delta, \Delta' \in W^\Sigma$. We need to show that $R^\Sigma\Delta'\Delta$, i.e., $\Box^{-1}\Delta' \subseteq \Delta$. By Lemma 52.9, this is equivalent to $\Diamond\Delta \subseteq \Delta'$. So suppose $\varphi \in \Delta$. By B, also $\Box\Diamond\varphi \in \Delta$. By the hypothesis that $R^\Sigma\Delta\Delta'$, we have that $\Box^{-1}\Delta \subseteq \Delta'$, and hence $\Diamond\varphi \in \Delta'$, as required.

Now suppose Σ contains 4, and suppose $R^\Sigma\Delta_1\Delta_2$ and $R^\Sigma\Delta_2\Delta_3$. We need to show $R^\Sigma\Delta_1\Delta_3$. From the hypothesis we have both $\Box^{-1}\Delta_1 \subseteq \Delta_2$ and $\Box^{-1}\Delta_2 \subseteq \Delta_3$. In order to show $R^\Sigma\Delta_1\Delta_3$ it suffices to show $\Box^{-1}\Delta_1 \subseteq \Delta_3$. So let $\psi \in \Box^{-1}\Delta_1$, i.e., $\Box\psi \in \Delta_1$. By 4, also $\Box\Box\psi \in \Delta_1$ and by hypothesis we get, first, that $\Box\psi \in \Delta_2$ and, second, that $\psi \in \Delta_3$, as desired.

Now suppose Σ contains 5, suppose $R^\Sigma\Delta_1\Delta_2$ and $R^\Sigma\Delta_1\Delta_3$. We need to show $R^\Sigma\Delta_2\Delta_3$. The first hypothesis gives $\Box^{-1}\Delta_1 \subseteq \Delta_2$, and the second hypothesis is equivalent to $\Diamond\Delta_3 \subseteq \Delta_1$, by Lemma 52.9. To show $R^\Sigma\Delta_2\Delta_3$, by Lemma 52.9, it suffices to show $\Diamond\Delta_3 \subseteq \Delta_2$. So let $\Diamond\varphi \in \Diamond\Delta_3$, i.e., $\varphi \in \Delta_3$. By the second hypothesis $\Diamond\varphi \in \Delta_1$ and by 5, $\Box\Diamond\varphi \in \Delta_1$ as well. But now the first hypothesis gives $\Diamond\varphi \in \Delta_2$, as desired. □

As a corollary we obtain completeness results for a number of systems. For instance, we know that **S5** = **KT5** = **KTB4** is complete with respect to the class of all reflexive euclidean models, which is the same as the class of all reflexive, symmetric and transitive models.

Theorem 52.17. *Let $\mathcal{C}_\mathrm{D}$, $\mathcal{C}_\mathrm{T}$, $\mathcal{C}_\mathrm{B}$, $\mathcal{C}_4$, and $\mathcal{C}_5$ be the class of all serial, reflexive, symmetric, transitive, and euclidean models (respectively). Then for* nml:com:fra: thm:generaldet

any schemas $\varphi_1, \ldots, \varphi_n$ among D, T, B, 4, *and* 5, *the system* $\mathbf{K}\varphi_1 \ldots \varphi_n$ *is determined by the class of models* $\mathcal{C} = \mathcal{C}_{\varphi_1} \cap \cdots \cap \mathcal{C}_{\varphi_n}$.

Proposition 52.18. *Let Σ be a normal modal logic; then:*

nml:com:fra:prop:anotherfive-a

1. *If Σ contains the schema $\Diamond\varphi \to \Box\varphi$ then the canonical model for Σ is partially functional.*

2. *If Σ contains the schema $\Diamond\varphi \leftrightarrow \Box\varphi$ then the canonical model for Σ is functional.*

3. *If Σ contains the schema $\Box\Box\varphi \to \Box\varphi$ then the canonical model for Σ is weakly dense.*

(see Table 50.2 for definitions of these frame properties).

Proof. 1. Suppose that Σ contains the schema $\Diamond\varphi \to \Box\varphi$, to show that R^Σ is partially functional we need to prove that for any Δ_1, Δ_2, $\Delta_3 \in W^\Sigma$, if $R^\Sigma\Delta_1\Delta_2$ and $R^\Sigma\Delta_1\Delta_3$ then $\Delta_2 = \Delta_3$. Since $R^\Sigma\Delta_1\Delta_2$ we have $\Box^{-1}\Delta_1 \subseteq \Delta_2$ and since $R^\Sigma\Delta_1\Delta_3$ also $\Box^{-1}\Delta_1 \subseteq \Delta_3$. The identity $\Delta_2 = \Delta_3$ will follow if we can establish the two inclusions $\Delta_2 \subseteq \Delta_3$ and $\Delta_3 \subseteq \Delta_2$. For the first inclusion, let $\varphi \in \Delta_2$; then $\Diamond\varphi \in \Delta_1$, and by the schema and deductive closure of Δ_1 also $\Box\varphi \in \Delta_1$, whence by the hypothesis that $R^\Sigma\Delta_1\Delta_3$, $\varphi \in \Delta_3$. The second inclusion is similar.

2. This follows immediately from part (1) and the seriality proof in Theorem 52.16.

3. Suppose Σ contains the schema $\Box\Box\varphi \to \Box\varphi$ and to show that R^Σ is weakly dense, let $R^\Sigma\Delta_1\Delta_2$. We need to show that there is a complete Σ-consistent set Δ_3 such that $R^\Sigma\Delta_1\Delta_3$ and $R^\Sigma\Delta_3\Delta_2$. Let:

$$\Gamma = \Box^{-1}\Delta_1 \cup \Diamond\Delta_2.$$

It suffices to show that Γ is Σ-consistent, for then by Lindenbaum's Lemma it can be extended to a complete Σ-consistent set Δ_3 such that $\Box^{-1}\Delta_1 \subseteq \Delta_3$ and $\Diamond\Delta_2 \subseteq \Delta_3$, i.e., $R^\Sigma\Delta_1\Delta_3$ and $R^\Sigma\Delta_3\Delta_2$ (by Lemma 52.9).

Suppose for contradiction that Γ is not consistent. Then there are formulas $\Box\varphi_1, \ldots, \Box\varphi_n \in \Delta_1$ and $\psi_1, \ldots, \psi_m \in \Delta_2$ such that

$$\varphi_1, \ldots, \varphi_n, \Diamond\psi_1, \ldots, \Diamond\psi_m \vdash_\Sigma \bot.$$

Since $\Diamond(\psi_1 \wedge \cdots \wedge \psi_m) \to (\Diamond\psi_1 \wedge \cdots \wedge \Diamond\psi_m)$ is derivable in every normal modal logic, we argue as follows, contradicting the consistency of Δ_2:

$$\begin{aligned}
&\varphi_1, \ldots, \varphi_n, \Diamond\psi_1, \ldots, \Diamond\psi_m \vdash_\Sigma \bot \\
&\varphi_1, \ldots, \varphi_n \vdash_\Sigma (\Diamond\psi_1 \wedge \cdots \wedge \Diamond\psi_m) \to \bot \\
&\qquad \text{by the deduction theorem} \\
&\qquad \text{Proposition 51.37(4), and TAUT} \\
&\varphi_1, \ldots, \varphi_n \vdash_\Sigma \Diamond(\psi_1 \wedge \cdots \wedge \psi_m) \to \bot \\
&\qquad \text{since } \Sigma \text{ is normal} \\
&\varphi_1, \ldots, \varphi_n \vdash_\Sigma \neg\Diamond(\psi_1 \wedge \cdots \wedge \psi_m) \\
&\qquad \text{by PL} \\
&\varphi_1, \ldots, \varphi_n \vdash_\Sigma \Box\neg(\psi_1 \wedge \cdots \wedge \psi_m) \\
&\qquad \Box\neg \text{ for } \neg\Diamond \\
&\Box\varphi_1, \ldots, \Box\varphi_n \vdash_\Sigma \Box\Box\neg(\psi_1 \wedge \cdots \wedge \psi_m) \\
&\qquad \text{by Lemma 52.6} \\
&\Box\varphi_1, \ldots, \Box\varphi_n \vdash_\Sigma \Box\neg(\psi_1 \wedge \cdots \wedge \psi_m) \\
&\qquad \text{by schema } \Box\Box\varphi \to \Box\varphi \\
&\Delta_1 \vdash_\Sigma \Box\neg(\psi_1 \wedge \cdots \wedge \psi_m) \\
&\qquad \text{by monotonicity, Proposition 51.37(1)} \\
&\Box\neg(\psi_1 \wedge \cdots \wedge \psi_m) \in \Delta_1 \\
&\qquad \text{by deductive closure;} \\
&\neg(\psi_1 \wedge \cdots \wedge \psi_m) \in \Delta_2 \\
&\qquad \text{since } R^\Sigma \Delta_1 \Delta_2.
\end{aligned}$$

□

On the strength of these examples, one might think that every system Σ of modal logic is *complete*, in the sense that it proves every formula which is valid in every frame in which every theorem of Σ is valid. Unfortunately, there are many systems that are not complete in this sense.

Chapter 53

Filtrations and Decidability

content/normal-modal-logic/filtrations/introduction.tex

53.1 Introduction

nml:fil:int:sec

One important question about a logic is always whether it is decidable, i.e., if there is an effective procedure which will answer the question "is this formula valid." Propositional logic is decidable: we can effectively test if a formula is a tautology by constructing a truth table, and for a given formula, the truth table is finite. But we can't obviously test if a modal formula is true in all models, for there are infinitely many of them. We can list all the finite models relevant to a given formula, since only the assignment of subsets of worlds to propositional variables which actually occur in the formula are relevant. If the accessibility relation is fixed, the possible different assignments $V(p)$ are just all the subsets of W, and if $|W| = n$ there are 2^n of those. If our formula φ contains m propositional variables there are then 2^{nm} different models with n worlds. For each one, we can test if φ is true at all worlds, simply by computing the truth value of φ in each. Of course, we also have to check all possible accessibility relations, but there are only finitely many relations on n worlds as well (specifically, the number of subsets of $W \times W$, i.e., 2^{n^2}.

If we are not interested in the logic **K**, but a logic defined by some class of models (e.g., the reflexive transitive models), we also have to be able to test if the accessibility relation is of the right kind. We can do that whenever the frames we are interested in are definable by modal formulas (e.g., by testing if T and 4 valid in the frame). So, the idea would be to run through all the finite frames, test each one if it is a frame in the class we're interested in, then list all the possible models on that frame and test if φ is true in each. If not, stop: φ is not valid in the class of models of interest.

There is a problem with this idea: we don't know when, if ever, we can stop looking. If the formula has a finite countermodel, our procedure will find it. But if it has no finite countermodel, we won't get an answer. The formula may be valid (no countermodels at all), or it may have only an infinite countermodel, which we'll never look at. This problem can be overcome if we can show that every formula that has a countermodel has a finite countermodel. If this is the case we say the logic has the *finite model property*.

But how would we show that a logic has the finite model property? One way of doing this would be to find a way to turn an infinite (counter)model of φ into a finite one. If that can be done, then whenever there is a model in which φ is not true, then the resulting finite model also makes φ not true. That finite model will show up on our list of all finite models, and we will eventually determine, for every formula that is not valid, that it isn't. Our procedure won't terminate if the formula is valid. If we can show in addition that there is some maximum size that the finite model our procedure provides can have, and that this maximum size depends only on the formula φ, we will have a size up to which we have to test finite models in our search for countermodels. If we haven't found a countermodel by then, there are none. Then our procedure

will, in fact, decide the question "is φ valid?" for any formula φ.

A strategy that often works for turning infinite structures into finite structures is that of "identifying" elements of the structure which behave the same way in relevant respects. If there are infinitely many worlds in $\mathfrak{M}$ that behave the same in relevant respects, then we may be able to collect *all* worlds in finitely many (possibly infinite) "classes" of such worlds. In other words, we should partition the set of worlds in the right way, i.e., in such a way that each partition contains infinitely many worlds, but there are only finitely many partitions. Then we define a new model $\mathfrak{M}^*$ where the worlds are the partitions. Finitely many partitions in the old model give us finitely many worlds in the new model, i.e., a finite model. Let's call the partition a world w is in $[w]$. We'll want it to be the case that $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \varphi$, since we want the new model to be a countermodel to φ if the old one was. This requires that we define the partition, as well as the accessibility relation of $\mathfrak{M}^*$ in the right way.

To see how this would go, first imagine we have no accessibility relation. $\mathfrak{M}, w \Vdash \Box\psi$ iff for some $v \in W$, $\mathfrak{M}, v \Vdash \Box\psi$, and the same for $\mathfrak{M}^*$, except with $[w]$ and $[v]$. As a first idea, let's say that two worlds u and v are equivalent (belong to the same partition) if they agree on all propositional variables in $\mathfrak{M}$, i.e., $\mathfrak{M}, u \Vdash p$ iff $\mathfrak{M}, v \Vdash p$. Let $V^*(p) = \{[w] : \mathfrak{M}, w \Vdash p\}$. Our aim is to show that $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \varphi$. Obviously, we'd prove this by induction: The base case would be $\varphi \equiv p$. First suppose $\mathfrak{M}, w \Vdash p$. Then $[w] \in V^*$ by definition, so $\mathfrak{M}^*, [w] \Vdash p$. Now suppose that $\mathfrak{M}^*, [w] \Vdash p$. That means that $[w] \in V^*(p)$, i.e., for some v equivalent to w, $\mathfrak{M}, v \Vdash p$. But "w equivalent to v" means "w and v make all the same propositional variables true," so $\mathfrak{M}, w \Vdash p$. Now for the inductive step, e.g., $\varphi \equiv \neg\psi$. Then $\mathfrak{M}, w \Vdash \neg\psi$ iff $\mathfrak{M}, w \nVdash \psi$ iff $\mathfrak{M}^*, [w] \nVdash \psi$ (by inductive hypothesis) iff $\mathfrak{M}^*, [w] \Vdash \neg\psi$. Similarly for the other non-modal operators. It also works for $\Box$: suppose $\mathfrak{M}^*, [w] \Vdash \Box\psi$. That means that for every $[u]$, $\mathfrak{M}^*, [u] \Vdash \psi$. By inductive hypothesis, for every u, $\mathfrak{M}, u \Vdash \psi$. Consequently, $\mathfrak{M}, w \Vdash \Box\psi$.

In the general case, where we have to also define the accessibility relation for $\mathfrak{M}^*$, things are more complicated. We'll call a model $\mathfrak{M}^*$ a *filtration* if its accessibility relation R^* satisfies the conditions required to make the inductive proof above go through. Then any filtration $\mathfrak{M}^*$ will make φ true at $[w]$ iff $\mathfrak{M}$ makes φ true at w. However, now we also have to show that there *are* filtrations, i.e., we can define R^* so that it satisfies the required conditions. In order for this to work, however, we have to require that worlds u, v count as equivalent not just when they agree on all propositional variables, but on all sub-formulas of φ. Since φ has only finitely many sub-formulas, this will still guarantee that the filtration is finite. There is not just one way to define a filtration, and in order to make sure that the accessibility relation of the filtration satisfies the required properties (e.g., reflexive, transitive, etc.) we have to be inventive with the definition of R^*.

---

content/normal-modal-logic/filtrations/preliminaries.tex

53.2 Preliminaries

nml:fil:pre:sec

Filtrations allow us to establish the decidability of our systems of modal logic by showing that they have the *finite model property*, i.e., that any formula that is true (false) in a model is also true (false) in a *finite* model. Filtrations are defined relative to sets of formulas which are closed under subformulas.

nml:fil:pre:defn:modallyclosed

Definition 53.1. A set Γ of formulas is *closed under subformulas* if it contains every subformula of a formula in Γ. Further, Γ is *modally closed* if it is closed under subformulas and moreover $\varphi \in \Gamma$ implies $\Box\varphi, \Diamond\varphi \in \Gamma$.

For instance, given a formula φ, the set of all its sub-formulas is closed under sub-formulas. When we're defining a filtration of a model through the set of sub-formulas of φ, it will have the property we're after: it makes φ true (false) iff the original model does.

The set of worlds of a filtration of $\mathfrak{M}$ through Γ is defined as the set of all equivalence classes of the following equivalence relation.

Definition 53.2. Let $\mathfrak{M} = \langle W, R, V\rangle$ and suppose Γ is closed under sub-formulas. Define a relation $\equiv$ on W to hold of any two worlds that make the same formulas from Γ true, i.e.:

$$u \equiv v \quad \text{if and only if} \quad \forall \varphi \in \Gamma : \mathfrak{M}, u \Vdash \varphi \Leftrightarrow \mathfrak{M}, v \Vdash \varphi.$$

The equivalence class $[w]_\equiv$ of a world w, or $[w]$ for short, is the set of all worlds $\equiv$-equivalent to w:

$$[w] = \{v : v \equiv w\}.$$

Proposition 53.3. *Given $\mathfrak{M}$ and Γ, $\equiv$ as defined above is an equivalence relation, i.e., it is reflexive, symmetric, and transitive.*

Proof. The relation $\equiv$ is reflexive, since w makes exactly the same formulas from Γ true as itself. It is symmetric since if u makes the same formulas from Γ true as v, the same holds for v and u. It is also transitive, since if u makes the same formulas from Γ true as v, and v as w, then u makes the same formulas from Γ true as w. □

The relation $\equiv$, like any equivalence relation, divides W into *partitions*, i.e., subsets of W which are pairwise disjoint, and together cover all of W. Every $w \in W$ is an element of one of the partitions, namely of $[w]$, since $w \equiv w$. So the partitions $[w]$ cover all of W. They are pairwise disjoint, for if $u \in [w]$ and $u \in [v]$, then $u \equiv w$ and $u \equiv v$, and by symmetry and transitivity, $w \equiv v$, and so $[w] = [v]$.

---

content/normal-modal-logic/filtrations/filtrations-def.tex

53.3 Filtrations

Rather than define "the" filtration of $\mathfrak{M}$ through Γ, we define when a model $\mathfrak{M}^*$ counts as a filtration of $\mathfrak{M}$. All filtrations have the same set of worlds W^* and the same valuation V^*. But different filtrations may have different accessibility relations R^*. To count as a filtration, R^* has to satisfy a number of conditions, however. These conditions are exactly what we'll require to prove the main result, namely that $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \varphi$, provided $\varphi \in \Gamma$.

nml:fil:fil:sec

Definition 53.4. Let Γ be closed under subformulas and $\mathfrak{M} = \langle W, R, V\rangle$. A *filtration of $\mathfrak{M}$ through Γ* is any model $\mathfrak{M}^* = \langle W^*, R^*, V^*\rangle$, where:

nml:fil:fil:defn:filtration

1. $W^* = \{[w] : w \in W\}$;
2. For any $u, v \in W$:
   a) If Ruv then $R^*[u][v]$;
   b) If $R^*[u][v]$ then for any $\Box\varphi \in \Gamma$, if $\mathfrak{M}, u \Vdash \Box\varphi$ then $\mathfrak{M}, v \Vdash \varphi$;
   c) If $R^*[u][v]$ then for any $\Diamond\varphi \in \Gamma$, if $\mathfrak{M}, v \Vdash \varphi$ then $\mathfrak{M}, u \Vdash \Diamond\varphi$.
3. $V^*(p) = \{[u] : u \in V(p)\}$.

nml:fil:fil:defn:filtration-R
nml:fil:fil:defn:filtration-R1
nml:fil:fil:defn:filtration-R2
nml:fil:fil:defn:filtration-R3

It's worthwhile thinking about what $V^*(p)$ is: the set consisting of the equivalence classes $[w]$ of all worlds w where p is true in $\mathfrak{M}$. On the one hand, if $w \in V(p)$, then $[w] \in V^*(p)$ by that definition. However, it is not necessarily the case that if $[w] \in V^*(p)$, then $w \in V(p)$. If $[w] \in V^*(p)$ we are only guaranteed that $[w] = [u]$ for *some* $u \in V(p)$. Of course, $[w] = [u]$ means that $w \equiv u$. So, when $[w] \in V^*(p)$ we can (only) conclude that $w \equiv u$ for some $u \in V(p)$.

Theorem 53.5. *If $\mathfrak{M}^*$ is a filtration of $\mathfrak{M}$ through Γ, then for every $\varphi \in \Gamma$ and $w \in W$, we have $\mathfrak{M}, w \Vdash \varphi$ if and only if $\mathfrak{M}^*, [w] \Vdash \varphi$.*

nml:fil:fil:thm:filtrations

Proof. By induction on φ, using the fact that Γ is closed under subformulas. Since $\varphi \in \Gamma$ and Γ is closed under sub-formulas, all sub-formulas of φ are also $\in \Gamma$. Hence in each inductive step, the induction hypothesis applies to the sub-formulas of φ.

1. $\varphi \equiv \bot$: Neither $\mathfrak{M}, w \Vdash \varphi$ nor $\mathfrak{M}^*, [w] \Vdash \varphi$.
2. $\varphi \equiv \top$: Both $\mathfrak{M}, w \Vdash \varphi$ and $\mathfrak{M}^*, [w] \Vdash \varphi$.
3. $\varphi \equiv p$: The left-to-right direction is immediate, as $\mathfrak{M}, w \Vdash \varphi$ only if $w \in V(p)$, which implies $[w] \in V^*(p)$, i.e., $\mathfrak{M}^*, [w] \Vdash \varphi$. Conversely, suppose $\mathfrak{M}^*, [w] \Vdash \varphi$, i.e., $[w] \in V^*(p)$. Then for some $v \in V(p)$, $w \equiv v$. Of course then also $\mathfrak{M}, v \Vdash p$. Since $w \equiv v$, w and v make the same formulas from Γ true. Since by assumption $p \in \Gamma$ and $\mathfrak{M}, v \Vdash p$, $\mathfrak{M}, w \Vdash \varphi$.
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$. By induction hypothesis, $\mathfrak{M}, w \nVdash \psi$ iff $\mathfrak{M}^*, [w] \nVdash \psi$. Finally, $\mathfrak{M}^*, [w] \nVdash \psi$ iff $\mathfrak{M}^*, [w] \Vdash \varphi$.

5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$. By induction hypothesis, $\mathfrak{M}, w \Vdash \psi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$, and $\mathfrak{M}, w \Vdash \chi$ iff $\mathfrak{M}^*, [w] \Vdash \chi$. And $\mathfrak{M}^*, [w] \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$ and $\mathfrak{M}^*, [w] \Vdash \chi$.

6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$. By induction hypothesis, $\mathfrak{M}, w \Vdash \psi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$, and $\mathfrak{M}, w \Vdash \chi$ iff $\mathfrak{M}^*, [w] \Vdash \chi$. And $\mathfrak{M}^*, [w] \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$ or $\mathfrak{M}^*, [w] \Vdash \chi$.

7. $\varphi \equiv (\psi \to \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$ or $\mathfrak{M}, w \Vdash \chi$. By induction hypothesis, $\mathfrak{M}, w \Vdash \psi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$, and $\mathfrak{M}, w \Vdash \chi$ iff $\mathfrak{M}^*, [w] \Vdash \chi$. And $\mathfrak{M}^*, [w] \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \nVdash \psi$ or $\mathfrak{M}^*, [w] \Vdash \chi$.

8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$, or $\mathfrak{M}, w \nVdash \psi$ and $\mathfrak{M}, w \nVdash \chi$. By induction hypothesis, $\mathfrak{M}, w \Vdash \psi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$, and $\mathfrak{M}, w \Vdash \chi$ iff $\mathfrak{M}^*, [w] \Vdash \chi$. And $\mathfrak{M}^*, [w] \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$ and $\mathfrak{M}^*, [w] \Vdash \chi$, or $\mathfrak{M}^*, [w] \nVdash \psi$ and $\mathfrak{M}^*, [w] \nVdash \chi$.

9. $\varphi \equiv \Box\psi$: Suppose $\mathfrak{M}, w \Vdash \varphi$; to show that $\mathfrak{M}^*, [w] \Vdash \varphi$, let v be such that $R^*[w][v]$. From Definition 53.4(2b), we have that $\mathfrak{M}, v \Vdash \psi$, and by inductive hypothesis $\mathfrak{M}^*, [v] \Vdash \psi$. Since v was arbitrary, $\mathfrak{M}^*, [w] \Vdash \varphi$ follows.

   Conversely, suppose $\mathfrak{M}^*, [w] \Vdash \varphi$ and let v be arbitrary such that Rwv. From Definition 53.4(2a), we have $R^*[w][v]$, so that $\mathfrak{M}^*, [v] \Vdash \psi$; by inductive hypothesis $\mathfrak{M}, v \Vdash \psi$, and since v was arbitrary, $\mathfrak{M}, w \Vdash \varphi$.

10. $\varphi \equiv \Diamond\psi$: Suppose $\mathfrak{M}, w \Vdash \varphi$. Then for some $v \in W$, Rwv and $\mathfrak{M}, v \Vdash \psi$. By inductive hypothesis $\mathfrak{M}^*, [v] \Vdash \psi$, and by Definition 53.4(2a), we have $R^*[w][v]$. Thus, $\mathfrak{M}^*, [w] \Vdash \varphi$.

    Now suppose $\mathfrak{M}^*, [w] \Vdash \varphi$. Then for some $[v] \in W^*$ with $R^*[w][v]$, $\mathfrak{M}^*, [v] \Vdash \psi$. By inductive hypothesis $\mathfrak{M}, v \Vdash \psi$. By Definition 53.4(2c), we have that $\mathfrak{M}, w \Vdash \varphi$. □

Problem 53.1. Complete the proof of Theorem 53.5

What holds for truth at worlds in a model also holds for truth in a model and validity in a class of models.

Corollary 53.6. *Let Γ be closed under subformulas. Then:*

1. *If $\mathfrak{M}^*$ is a filtration of $\mathfrak{M}$ through Γ then for any $\varphi \in \Gamma$: $\mathfrak{M} \Vdash \varphi$ if and only if $\mathfrak{M}^* \Vdash \varphi$.*

2. *If $\mathcal{C}$ is a class of models and $\Gamma(\mathcal{C})$ is the class of Γ-filtrations of models in $\mathcal{C}$, then any formula $\varphi \in \Gamma$ is valid in $\mathcal{C}$ if and only if it is valid in $\Gamma(\mathcal{C})$.*

---

content/normal-modal-logic/filtrations/examples-of-filtrations.tex

53.4 Examples of Filtrations

We have not yet shown that there are any filtrations. But indeed, for any model $\mathfrak{M}$, there are many filtrations of $\mathfrak{M}$ through Γ. We identify two, in particular: the finest and coarsest filtrations. Filtrations of the same models will differ in their accessibility relation (as Definition 53.4 stipulates directly what W^* and V^* should be). The finest filtration will have as few related worlds as possible, whereas the coarsest will have as many as possible. nml:fil:exf:sec

Definition 53.7. Where Γ is closed under subformulas, the *finest* filtration $\mathfrak{M}^*$ of a model $\mathfrak{M}$ is defined by putting:

$$R^*[u][v] \quad \text{if and only if} \quad \exists u' \in [u]\, \exists v' \in [v] : Ru'v'.$$

Proposition 53.8. *The finest filtration $\mathfrak{M}^*$ is indeed a filtration.* nml:fil:exf:prop:finest

Proof. We need to check that R^*, so defined, satisfies Definition 53.4(2). We check the three conditions in turn.

If Ruv then since $u \in [u]$ and $v \in [v]$, also $R^*[u][v]$, so (2a) is satisfied.

For (2b), suppose $\Box\varphi \in \Gamma$, $R^*[u][v]$, and $\mathfrak{M}, u \Vdash \Box\varphi$. By definition of R^*, there are $u' \equiv u$ and $v' \equiv v$ such that $Ru'v'$. Since u and u' agree on Γ, also $\mathfrak{M}, u' \Vdash \Box\varphi$, so that $\mathfrak{M}, v' \Vdash \varphi$. By closure of Γ under sub-formulas, v and v' agree on φ, so $\mathfrak{M}, v \Vdash \varphi$, as desired.

To verify (2c), suppose $\Diamond\varphi \in \Gamma$, $R^*[u][v]$, and $\mathfrak{M}, v \Vdash \varphi$. By definition of R^*, there are $u' \equiv u$ and $v' \equiv v$ such that $Ru'v'$. Since v and v' agree on Γ, and Γ is closed under sub-formulas, also $\mathfrak{M}, v' \Vdash \varphi$, so that $\mathfrak{M}, u' \Vdash \Diamond\varphi$. Since u and u' also agree on Γ, $\mathfrak{M}, u \Vdash \Diamond\varphi$. □

Problem 53.2. Complete the proof of Proposition 53.8.

Definition 53.9. Where Γ is closed under subformulas, the *coarsest* filtration $\mathfrak{M}^*$ of a model $\mathfrak{M}$ is defined by putting $R^*[u][v]$ if and only if *both* of the following conditions are met:

1. If $\Box\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Box\varphi$ then $\mathfrak{M}, v \Vdash \varphi$; nml:fil:exf:defn:coarsest-Box
2. If $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \varphi$ then $\mathfrak{M}, u \Vdash \Diamond\varphi$. nml:fil:exf:defn:coarsest-Diamond

Proposition 53.10. *The coarsest filtration $\mathfrak{M}^*$ is indeed a filtration.*

Proof. Given the definition of R^*, the only condition that is left to verify is the implication from Ruv to $R^*[u][v]$. So assume Ruv. Suppose $\Box\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Box\varphi$; then obviously $\mathfrak{M}, v \Vdash \varphi$, and (1) is satisfied. Suppose $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \varphi$. Then $\mathfrak{M}, u \Vdash \Diamond\varphi$ since Ruv, and (2) is satisfied. □

Example 53.11. Let $W = \mathbb{Z}^+$, Rnm iff $m = n + 1$, and $V(p) = \{2n : n \in \mathbb{N}\}$. The model $\mathfrak{M} = \langle W, R, V\rangle$ is depicted in Figure 53.1. The worlds are 1, 2, etc.; each world can access exactly one other world—its successor—and p is true at all and only the even numbers.

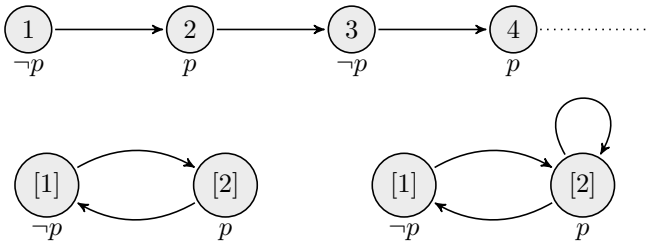


Figure 53.1: An infinite model and its filtrations.

nml:fil:exf:
fig:ex-filtration

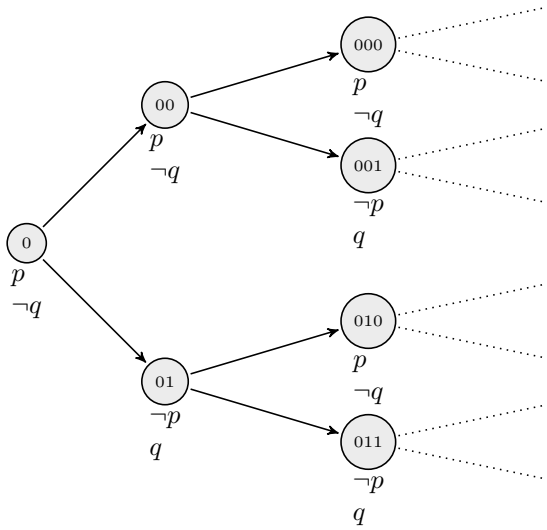
Now let Γ be the set of sub-formulas of $\Box p \to p$, i.e., $\{p, \Box p, \Box p \to p\}$. p is true at all and only the even numbers, $\Box p$ is true at all and only the odd numbers, so $\Box p \to p$ is true at all and only the even numbers. In other words, every odd number makes $\Box p$ true but p and $\Box p \to p$ false; every even number makes p and $\Box p \to p$ true, but $\Box p$ false. So $W^* = \{[1], [2]\}$, where $[1] = \{1, 3, 5, \dots\}$ and $[2] = \{2, 4, 6, \dots\}$. Since $2 \in V(p)$, $[2] \in V^*(p)$; since $1 \notin V(p)$, $[1] \notin V^*(p)$. So $V^*(p) = \{[2]\}$.

Any filtration based on W^* must have an accessibility relation that includes $\langle [1], [2] \rangle, \langle [2], [1] \rangle$: since $R12$, we must have $R^*[1][2]$ by Definition 53.4(2a), and since $R23$ we must have $R^*[2][3]$, and $[3] = [1]$. It cannot include $\langle [1], [1] \rangle$: if it did, we'd have $R^*[1][1]$, $\mathfrak{M}, 1 \Vdash \Box p$ but $\mathfrak{M}, 1 \nVdash p$, contradicting (2b). Nothing requires or rules out that $R^*[2][2]$. So, there are two possible filtrations of $\mathfrak{M}$, corresponding to the two accessibility relations

$$\{\langle [1], [2] \rangle, \langle [2], [1] \rangle\} \text{ and } \{\langle [1], [2] \rangle, \langle [2], [1] \rangle, \langle [2], [2] \rangle\}.$$

In either case, p and $\Box p \to p$ are false and $\Box p$ is true at $[1]$; p and $\Box p \to p$ are true and $\Box p$ is false at $[2]$.

Problem 53.3. Consider the following model $\mathfrak{M} = \langle W, R, V \rangle$ where $W = \{0\sigma : \sigma \in \mathbb{B}^*\}$, the set of sequences of 0s and 1s starting with 0, with $R\sigma\sigma'$ iff $\sigma' = \sigma 0$ or $\sigma' = \sigma 1$, and $V(p) = \{\sigma 0 : \sigma \in \mathbb{B}^*\}$ and $V(q) = \{\sigma 1 : \sigma \in \mathbb{B}^* \setminus \{1\}\}$. Here's a picture:



We have $\mathfrak{M}, w \nVdash \Box(p \vee q) \to (\Box p \vee \Box q)$ for every w.

Let Γ be the set of sub-formulas of $\Box(p \vee q) \to (\Box p \vee \Box q)$. What are W^* and V^*? What is the accessibility relation of the finest filtration of $\mathfrak{M}$? Of the coarsest?

---

content/normal-modal-logic/filtrations/finite.tex

53.5 Filtrations are Finite

nml:fil:fin:sec

We've defined filtrations for any set Γ that is closed under sub-formulas. Nothing in the definition itself guarantees that filtrations are finite. In fact, when Γ is infinite (e.g., is the set of all formulas), it may well be infinite. However, if Γ is finite (e.g., when it is the set of sub-formulas of a given formula φ), so is any filtration through Γ.

nml:fil:fin:prop:filt-are-finite

Proposition 53.12. *If Γ is finite then any filtration $\mathfrak{M}^*$ of a model $\mathfrak{M}$ through Γ is also finite.*

Proof. The size of W^* is the number of different classes $[w]$ under the equivalence relation $\equiv$. Any two worlds u, v in such class—that is, any u and v such that $u \equiv v$—agree on all formulas φ in Γ, $\varphi \in \Gamma$ either φ is true at both u and v, or at neither. So each class $[w]$ corresponds to subset of Γ, namely the set of all $\varphi \in \Gamma$ such that φ is true at the worlds in $[w]$. No two different classes $[u]$ and $[v]$ correspond to the same subset of Γ. For if the set of formulas true at u and that of formulas true at v are the same, then u and v agree on all formulas in Γ, i.e., $u \equiv v$. But then $[u] = [v]$. So, there is an injective function from W^* to $\wp(\Gamma)$, and hence $|W^*| \leq |\wp(\Gamma)|$. Hence if Γ contains n sentences, the cardinality of W^* is no greater than 2^n. □

content/normal-modal-logic/filtrations/S5-fmp.tex

53.6 K and S5 have the Finite Model Property

nml:fil:fmp:sec

Definition 53.13. A system Σ of modal logic is said to have the *finite model property* if whenever a formula φ is true at a world in a model of Σ then φ is true at a world in a *finite* model of Σ.

nml:fil:fmp:prop:K-fmp

Proposition 53.14. **K** *has the finite model property.*

Proof. **K** is the set of valid formulas, i.e., any model is a model of **K**. By Theorem 53.5, if $\mathfrak{M}, w \Vdash \varphi$, then $\mathfrak{M}^*, w \Vdash \varphi$ for any filtration of $\mathfrak{M}$ through the set Γ of sub-formulas of φ. Any formula only has finitely many sub-formulas, so Γ is finite. By Proposition 53.12, $|W^*| \leq 2^n$, where n is the number of formulas in Γ. And since **K** imposes no restriction on models, $\mathfrak{M}^*$ is a **K**-model. □

To show that a logic **L** has the finite model property via filtrations it is essential that the filtration of an **L**-model is itself a **L**-model. Often this requires a fair bit of work, and not any filtration yields a **L**-model. However, for universal models, this still holds.

nml:fil:fmp:prop:univ-fin

Proposition 53.15. *Let $\mathcal{U}$ be the class of universal models (see Proposition 50.14) and $\mathcal{U}_{\text{Fin}}$ the class of all finite universal models. Then any formula φ is valid in $\mathcal{U}$ if and only if it is valid in $\mathcal{U}_{\text{Fin}}$.*

Proof. Finite universal models are universal models, so the left-to-right direction is trivial. For the right-to left direction, suppose that φ is false at some world w in a universal model $\mathfrak{M}$. Let Γ contain φ as well as all of its sub-formulas; clearly Γ is finite. Take a filtration $\mathfrak{M}^*$ of $\mathfrak{M}$; then $\mathfrak{M}^*$ is finite by Proposition 53.12, and by Theorem 53.5, φ is false at $[w]$ in $\mathfrak{M}^*$. It remains to observe that $\mathfrak{M}^*$ is also universal: given u and v, by hypothesis Ruv and by Definition 53.4(2), also $R^*[u][v]$. □

nml:fil:fmp:cor:S5fmp

Corollary 53.16. **S5** *has the finite model property.*

Proof. By Proposition 50.14, if φ is true at a world in some reflexive and euclidean model then it is true at a world in a universal model. By Proposition 53.15, it is true at a world in a finite universal model (namely the filtration of the model through the set of sub-formulas of φ). Every universal model is also reflexive and euclidean; so φ is true at a world in a finite reflexive euclidean model. □

Problem 53.4. Show that any filtration of a serial or reflexive model is also serial or reflexive (respectively).

Problem 53.5. Find a non-symmetric (non-transitive, non-euclidean) filtration of a symmetric (transitive, euclidean) model.

---

content/normal-modal-logic/filtrations/S5-decidable.tex

53.7 S5 is Decidable

nml:fil:dec:sec

The finite model property gives us an easy way to show that systems of modal logic given by schemas are *decidable* (i.e., that there is a computable procedure to determine whether a formula is derivable in the system or not).

Theorem 53.17. S5 *is decidable.*

Proof. Let φ be given, and suppose the propositional variables occurring in φ are among $p_1, \ldots, p_k$. Since for each n there are only finitely many models with n worlds assigning a value to $p_1, \ldots, p_k$, we can enumerate, *in parallel*, all the theorems of **S5** by generating proofs in some systematic way; and all the models containing 1, 2, ... worlds and checking whether φ fails at a world in some such model. Eventually one of the two parallel processes will give an answer, as by Theorem 52.17 and Corollary 53.16, either φ is derivable or it fails in a finite universal model. □

The above proof works for **S5** because filtrations of universal models are automatically universal. The same holds for reflexivity and seriality, but more work is needed for other properties.

---

content/normal-modal-logic/filtrations/more-filtrations.tex

53.8 Filtrations and Properties of Accessibility

nml:fil:acc:sec

As noted, filtrations of universal, serial, and reflexive models are always also universal, serial, or reflexive. But not every filtration of a symmetric or transitive model is symmetric or transitive, respectively. In some cases, however, it is possible to define filtrations so that this does hold. In order to do so, we proceed as in the definition of the coarsest filtration, but add additional conditions to the definition of R^*. Let Γ be closed under sub-formulas. Consider the relations $C_i(u, v)$ in Table 53.1 between worlds u, v in a model $\mathfrak{M} = \langle W, R, V\rangle$. We can define $R^*[u][v]$ on the basis of combinations of these conditions. For instance, if we stipulate that $R^*[u][v]$ iff the condition $C_1(u, v)$ holds, we get exactly the coarsest filtration. If we stipulate $R^*[u][v]$ iff both $C_1(u, v)$ and $C_2(u, v)$ hold, we get a different filtration. It is "finer" than the coarsest since fewer pairs of worlds satisfy $C_1(u, v)$ and $C_2(u, v)$ than $C_1(u, v)$ alone.

Theorem 53.18. *Let $\mathfrak{M} = \langle W, R, V\rangle$ be a model, Γ closed under sub-formulas. Let W^* and V^* be defined as in Definition 53.4. Then:*

nml:fil:acc:thm:more-filtrations

$C_1(u,v)$:	if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Box\varphi$ then $\mathfrak{M}, v \Vdash \varphi$; and if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \varphi$ then $\mathfrak{M}, u \Vdash \Diamond\varphi$;
$C_2(u,v)$:	if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \Box\varphi$ then $\mathfrak{M}, u \Vdash \varphi$; and if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \varphi$ then $\mathfrak{M}, v \Vdash \Diamond\varphi$;
$C_3(u,v)$:	if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Box\varphi$ then $\mathfrak{M}, v \Vdash \Box\varphi$; and if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \Diamond\varphi$ then $\mathfrak{M}, u \Vdash \Diamond\varphi$;
$C_4(u,v)$:	if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \Box\varphi$ then $\mathfrak{M}, u \Vdash \Box\varphi$; and if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Diamond\varphi$ then $\mathfrak{M}, v \Vdash \Diamond\varphi$;

Table 53.1: Conditions on possible worlds for defining filtrations.

nml:fil:acc:
tab:Cn-filtrations

1. *Suppose $R^*[u][v]$ if and only if $C_1(u,v) \wedge C_2(u,v)$. Then R^* is symmetric, and $\mathfrak{M}^* = \langle W^*, R^*, V^* \rangle$ is a filtration if $\mathfrak{M}$ is symmetric.*

2. *Suppose $R^*[u][v]$ if and only if $C_1(u,v) \wedge C_3(u,v)$. Then R^* is transitive, and $\mathfrak{M}^* = \langle W^*, R^*, V^* \rangle$ is a filtration if $\mathfrak{M}$ is transitive.*

3. *Suppose $R^*[u][v]$ if and only if $C_1(u,v) \wedge C_2(u,v) \wedge C_3(u,v) \wedge C_4(u,v)$. Then R^* is symmetric and transitive, and $\mathfrak{M}^* = \langle W^*, R^*, V^* \rangle$ is a filtration if $\mathfrak{M}$ is symmetric and transitive.*

4. *Suppose R^* is defined as $R^*[u][v]$ if and only if $C_1(u,v) \wedge C_3(u,v) \wedge C_4(u,v)$. Then R^* is transitive and euclidean, and $\mathfrak{M}^* = \langle W^*, R^*, V^* \rangle$ is a filtration if $\mathfrak{M}$ is transitive and euclidean.*

Proof. 1. It's immediate that R^* is symmetric, since $C_1(u,v) \Leftrightarrow C_2(v,u)$ and $C_2(u,v) \Leftrightarrow C_1(v,u)$. So it's left to show that if $\mathfrak{M}$ is symmetric then $\mathfrak{M}^*$ is a filtration through Γ. Condition $C_1(u,v)$ guarantees that (2b) and (2c) of Definition 53.4 are satisfied. So we just have to verify Definition 53.4(2a), i.e., that Ruv implies $R^*[u][v]$.

So suppose Ruv. To show $R^*[u][v]$ we need to establish that $C_1(u,v)$ and $C_2(u,v)$. For C_1: if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Box\varphi$ then also $\mathfrak{M}, v \Vdash \varphi$ (since Ruv). Similarly, if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \varphi$ then $\mathfrak{M}, u \Vdash \Diamond\varphi$ since Ruv. For C_2: if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \Box\varphi$ then Ruv implies Rvu by symmetry, so that $\mathfrak{M}, u \Vdash \varphi$. Similarly, if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \varphi$ then $\mathfrak{M}, v \Vdash \Diamond\varphi$ (since Rvu by symmetry).

2. Exercise.

3. Exercise.

4. Exercise. □

Problem 53.6. Complete the proof of Theorem 53.18.

content/normal-modal-logic/filtrations/euclidean-filtrations.tex

53.9 Filtrations of Euclidean Models

nml:fil:euc:sec

The approach of section 53.8 does not work in the case of models that are euclidean or serial and euclidean. Consider the model at the top of Figure 53.2, which is both euclidean and serial. Let $\Gamma = \{p, \Box p\}$. When taking a filtration through Γ, then $[w_1] = [w_3]$ since w_1 and w_3 are the only worlds that agree on Γ. Any filtration will also have the arrow inherited from $\mathfrak{M}$, as depicted in Figure 53.3. That model isn't euclidean. Moreover, we cannot add arrows to that model in order to make it euclidean. We would have to add double arrows between $[w_2]$ and $[w_4]$, and then also between w_2 and w_5. But $\Box p$ is supposed to be true at w_2, while p is false at w_5.

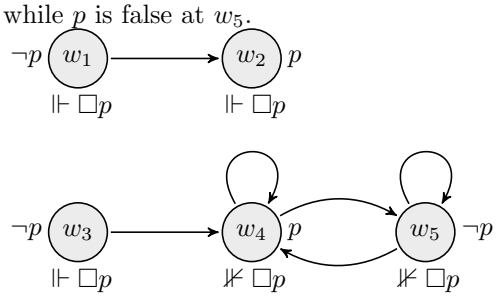


Figure 53.2: A serial and euclidean model.

nml:fil:euc:fig:ser-eucl

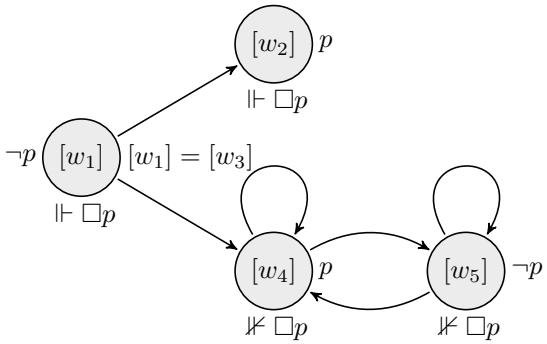


Figure 53.3: The filtration of the model in Figure 53.2.

nml:fil:euc:fig:ser-eucl2

In particular, to obtain a euclidean filtration it is not enough to consider filtrations through arbitrary Γ's closed under sub-formulas. Instead we need to consider sets Γ that are *modally closed* (see Definition 53.1). Such sets of sentences are infinite, and therefore do not immediately yield a finite model property or the decidability of the corresponding system.

Theorem 53.19. *Let Γ be modally closed, $\mathfrak{M} = \langle W, R, V\rangle$, and $\mathfrak{M}^* = \langle W^*, R^*, V^*\rangle$ be a coarsest filtration of $\mathfrak{M}$.*

nml:fil:euc:thm:modal-closed-filt

1. *If $\mathfrak{M}$ is symmetric, so is $\mathfrak{M}^*$.*
2. *If $\mathfrak{M}$ is transitive, so is $\mathfrak{M}^*$.*
3. *If $\mathfrak{M}$ is euclidean, so is $\mathfrak{M}^*$.*

Proof. 1. If $\mathfrak{M}^*$ is a coarsest filtration, then by definition $R^*[u][v]$ holds if and only if $C_1(u, v)$. For transitivity, suppose $C_1(u, v)$ and $C_1(v, w)$; we have to show $C_1(u, w)$. Suppose $\mathfrak{M}, u \Vdash \Box\varphi$; then $\mathfrak{M}, u \Vdash \Box\Box\varphi$ since 4 is valid in all transitive models; since $\Box\Box\varphi \in \Gamma$ by closure, also by $C_1(u, v)$, $\mathfrak{M}, v \Vdash \Box\varphi$ and by $C_1(v, w)$, also $\mathfrak{M}, w \Vdash \varphi$. Suppose $\mathfrak{M}, w \Vdash \varphi$; then $\mathfrak{M}, v \Vdash \Diamond\varphi$ by $C_1(v, w)$, since $\Diamond\varphi \in \Gamma$ by modal closure. By $C_1(u, v)$, we get $\mathfrak{M}, u \Vdash \Diamond\Diamond\varphi$ since $\Diamond\Diamond\varphi \in \Gamma$ by modal closure. Since $4_\Diamond$ is valid in all transitive models, $\mathfrak{M}, u \Vdash \Diamond\varphi$.

2. Exercise. Use the fact that both 5 and $5_\Diamond$ are valid in all euclidean models.

3. Exercise. Use the fact that B and $B_\Diamond$ are valid in all symmetric models. □

Problem 53.7. Complete the proof of Theorem 53.19.

Chapter 54

Modal Tableaux

Draft chapter on prefixed tableaux for modal logic. Needs more examples, completeness proofs, and discussion of how one can find countermodels from unsuccessful searches for closed tableaux.

---

content/normal-modal-logic/tableaux/introduction.tex

54.1 Introduction

nml:tab:int:sec

Tableaux are certain (downward-branching) trees of signed formulas, i.e., pairs consisting of a truth value sign ($\mathbb{T}$ or $\mathbb{F}$) and a sentence

$$\mathbb{T}\,\varphi \text{ or } \mathbb{F}\,\varphi.$$

A tableau begins with a number of *assumptions*. Each further signed formula is generated by applying one of the inference rules. Some inference rules add one or more signed formulas to a tip of the tree; others add two new tips, resulting in two branches. Rules result in signed formulas where the formula is less complex than that of the signed formula to which it was applied. When a branch contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, we say the branch is *closed*. If every branch in a tableau is closed, the entire tableau is closed. A closed tableau constitutes a derivation that shows that the set of signed formulas which were used to begin the tableau are unsatisfiable. This can be used to define a $\vdash$ relation: $\Gamma \vdash \varphi$ iff there is some finite set $\Gamma_0 = \{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ such that there is a closed tableau for the assumptions

$$\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}.$$

For modal logics, we have to both extend the notion of signed formula and add rules that cover $\Box$ and $\Diamond$ In addition to a sign($\mathbb{T}$ or $\mathbb{F}$), formulas in modal tableaux also have *prefixes* σ. The prefixes are non-empty sequences of positive integers, i.e., $\sigma \in (\mathbb{Z}^+)^* \setminus \{\Lambda\}$. When we write such prefixes without the surrounding $\langle\ \rangle$, and separate the individual elements by .'s instead of ,'s. If σ is a prefix, then $\sigma.n$ is $\sigma \frown \langle n \rangle$; e.g., if $\sigma = 1.2.1$, then $\sigma.3$ is 1.2.1.3. So for instance,

$$1.2\,\mathbb{T}\Box\varphi \to \varphi$$

is a *prefixed signed formula* (or just a *prefixed formula* for short).

Intuitively, the prefix names a world in a model that might satisfy the formulas on a branch of a tableau, and if σ names some world, then $\sigma.n$ names a world accessible from (the world named by) σ.

---

content/normal-modal-logic/tableaux/rules-for-K.tex

54.2 Rules for K

nml:tab:rul:sec

The rules for the regular propositional connectives are the same as for regular propositional signed tableaux, just with prefixes added. In each case, the rule applied to a signed formula $\sigma\, S\varphi$ produces new formulas that are also prefixed by σ. This should be intuitively clear: e.g., if $\varphi \wedge \psi$ is true at (a world named by) σ, then φ and ψ are true at σ (and not at any other world). We collect the propositional rules in Table 54.1.

The closure condition is the same as for ordinary tableaux, although we require that not just the formulas but also the prefixes must match. So a branch is closed if it contains both

$$\sigma\,\mathbb{T}\varphi \quad \text{and} \quad \sigma\,\mathbb{F}\varphi$$

for some prefix σ and formula φ.

The rules for setting up assumptions is also as for ordinary tableaux, except that for assumptions we always use the prefix 1. (It does not matter which

$\dfrac{\sigma\,\mathbb{T}\,\neg\varphi}{\sigma\,\mathbb{F}\,\varphi}\ \neg\mathbb{T}$	$\dfrac{\sigma\,\mathbb{F}\,\neg\varphi}{\sigma\,\mathbb{T}\,\varphi}\ \neg\mathbb{F}$
$\dfrac{\sigma\,\mathbb{T}\,\varphi\wedge\psi}{\begin{array}{c}\sigma\,\mathbb{T}\,\varphi\\ \sigma\,\mathbb{T}\,\psi\end{array}}\ \wedge\mathbb{T}$	$\dfrac{\sigma\,\mathbb{F}\,\varphi\wedge\psi}{\sigma\,\mathbb{F}\,\varphi \quad\mid\quad \sigma\,\mathbb{F}\,\psi}\ \wedge\mathbb{F}$
$\dfrac{\sigma\,\mathbb{T}\,\varphi\vee\psi}{\sigma\,\mathbb{T}\,\varphi \quad\mid\quad \sigma\,\mathbb{T}\,\psi}\ \vee\mathbb{T}$	$\dfrac{\sigma\,\mathbb{F}\,\varphi\vee\psi}{\begin{array}{c}\sigma\,\mathbb{F}\,\varphi\\ \sigma\,\mathbb{F}\,\psi\end{array}}\ \vee\mathbb{F}$
$\dfrac{\sigma\,\mathbb{T}\,\varphi\to\psi}{\sigma\,\mathbb{F}\,\varphi \quad\mid\quad \sigma\,\mathbb{T}\,\psi}\ \to\mathbb{T}$	$\dfrac{\sigma\,\mathbb{F}\,\varphi\to\psi}{\begin{array}{c}\sigma\,\mathbb{T}\,\varphi\\ \sigma\,\mathbb{F}\,\psi\end{array}}\ \to\mathbb{F}$

Table 54.1: Prefixed tableau rules for the propositional connectives

nml:tab:rul:
tab:prop-rules

prefix we use, as long as it's the same for all assumptions.) So, e.g., we say that

$$\psi_1, \ldots, \psi_n \vdash \varphi$$

iff there is a closed tableau for the assumptions

$$1\,\mathbb{T}\,\psi_1, \ldots, 1\,\mathbb{T}\,\psi_n, 1\,\mathbb{F}\,\varphi.$$

For the modal operators $\Box$ and $\Diamond$, the prefix of the conclusion of the rule applied to a formula with prefix σ is $\sigma.n$. However, which n is allowed depends on whether the sign is $\mathbb{T}$ or $\mathbb{F}$.

The $\Box\mathbb{T}$ rule extends a branch containing $\sigma\,\mathbb{T}\,\Box\varphi$ by $\sigma.n\,\mathbb{T}\,\varphi$. Similarly, the $\Diamond\mathbb{F}$ rule extends a branch containing $\sigma\,\mathbb{F}\,\Diamond\varphi$ by $\sigma.n\,\mathbb{F}\,\varphi$. They can only be applied for a prefix $\sigma.n$ which *already* occurs on the branch in which it is applied. Let's call such a prefix "used" (on the branch).

The $\Box\mathbb{F}$ rule extends a branch containing $\sigma\,\mathbb{F}\,\Box\varphi$ by $\sigma.n\,\mathbb{F}\,\varphi$. Similarly, the $\Diamond\mathbb{T}$ rule extends a branch containing $\sigma\,\mathbb{T}\,\Diamond\varphi$ by $\sigma.n\,\mathbb{T}\,\varphi$. These rules, however, can only be applied for a prefix $\sigma.n$ which *does not* already occur on the branch in which it is applied. We call such prefixes "new" (to the branch).

The rules are given in Table 54.2.

The requirement that the restriction that the prefix for $\Box\mathbb{T}$ must be used is necessary as otherwise we would count the following as a closed tableau:

$\dfrac{\sigma\,\mathbb{T}\,\Box\varphi}{\sigma.n\,\mathbb{T}\,\varphi}\,\Box\mathbb{T}$ $\sigma.n$ is used	$\dfrac{\sigma\,\mathbb{F}\,\Box\varphi}{\sigma.n\,\mathbb{F}\,\varphi}\,\Box\mathbb{F}$ $\sigma.n$ is new
$\dfrac{\sigma\,\mathbb{T}\,\Diamond\varphi}{\sigma.n\,\mathbb{T}\,\varphi}\,\Diamond\mathbb{T}$ $\sigma.n$ is new	$\dfrac{\sigma\,\mathbb{F}\,\Diamond\varphi}{\sigma.n\,\mathbb{F}\,\varphi}\,\Diamond\mathbb{F}$ $\sigma.n$ is used

Table 54.2: The modal rules for K.

nml:tab:rul:
tab:rules-K

1.	$1\,\mathbb{T}\ \ \Box\varphi$	Assumption
2.	$1\,\mathbb{F}\ \ \Diamond\varphi$	Assumption
3.	$1.1\,\mathbb{T}\ \ \varphi$	$\Box\mathbb{T}\,1$
4.	$1.1\,\mathbb{F}\ \ \varphi$	$\Diamond\mathbb{F}\,2$
	$\otimes$	

But $\Box\varphi \not\models \Diamond\varphi$, so our proof system would be unsound. Likewise, $\Diamond\varphi \not\models \Box\varphi$, but without the restriction that the prefix for $\Box\mathbb{F}$ must be new, this would be a closed tableau:

1.	$1\,\mathbb{T}\ \ \Diamond\varphi$	Assumption
2.	$1\,\mathbb{F}\ \ \Box\varphi$	Assumption
3.	$1.1\,\mathbb{T}\ \ \varphi$	$\Diamond\mathbb{T}\,1$
4.	$1.1\,\mathbb{F}\ \ \varphi$	$\Box\mathbb{F}\,2$
	$\otimes$	

---

content/normal-modal-logic/tableaux/proofs-in-K.tex

54.3 Tableaux for K

nml:tab:prk:
sec

Example 54.1. We give a closed tableau that shows $\vdash (\Box\varphi \wedge \Box\psi) \to \Box(\varphi \wedge \psi)$.

1.	$1\,\mathbb{F}\ \ (\Box\varphi \land \Box\psi) \to \Box(\varphi \land \psi)$		Assumption
2.	$1\,\mathbb{T}\ \ \Box\varphi \land \Box\psi$		$\to\mathbb{F}\,1$
3.	$1\,\mathbb{F}\ \ \Box(\varphi \land \psi)$		$\to\mathbb{F}\,1$
4.	$1\,\mathbb{T}\ \ \Box\varphi$		$\land\mathbb{T}\,2$
5.	$1\,\mathbb{T}\ \ \Box\psi$		$\land\mathbb{T}\,2$
6.	$1.1\,\mathbb{F}\ \ \varphi \land \psi$		$\Box\mathbb{F}\,3$
7.	$1.1\,\mathbb{F}\ \ \varphi$	$1.1\,\mathbb{F}\ \ \psi$	$\land\mathbb{F}\,6$
8.	$1.1\,\mathbb{T}\ \ \varphi$	$1.1\,\mathbb{T}\ \ \psi$	$\Box\mathbb{T}\,4$; $\Box\mathbb{T}\,5$
	$\otimes$	$\otimes$	

Example 54.2. We give a closed tableau that shows $\vdash \Diamond(\varphi \lor \psi) \to (\Diamond\varphi \lor \Diamond\psi)$:

1.	$1\,\mathbb{F}\ \ \Diamond(\varphi \lor \psi) \to (\Diamond\varphi \lor \Diamond\psi)$		Assumption
2.	$1\,\mathbb{T}\ \ \Diamond(\varphi \lor \psi)$		$\to\mathbb{F}\,1$
3.	$1\,\mathbb{F}\ \ \Diamond\varphi \lor \Diamond\psi$		$\to\mathbb{F}\,1$
4.	$1\,\mathbb{F}\ \ \Diamond\varphi$		$\lor\mathbb{F}\,3$
5.	$1\,\mathbb{F}\ \ \Diamond\psi$		$\lor\mathbb{F}\,3$
6.	$1.1\,\mathbb{T}\ \ \varphi \lor \psi$		$\Diamond\mathbb{T}\,2$
7.	$1.1\,\mathbb{T}\ \ \varphi$	$1.1\,\mathbb{T}\ \ \psi$	$\lor\mathbb{T}\,6$
8.	$1.1\,\mathbb{F}\ \ \varphi$	$1.1\,\mathbb{F}\ \ \psi$	$\Diamond\mathbb{F}\,4$; $\Diamond\mathbb{F}\,5$
	$\otimes$	$\otimes$	

Problem 54.1. Find closed tableaux in **K** for the following formulas:

1. $\Box\neg p \to \Box(p \to q)$
2. $(\Box p \lor \Box q) \to \Box(p \lor q)$
3. $\Diamond p \to \Diamond(p \lor q)$
4. $\Box(p \land q) \to \Box p$

---

content/normal-modal-logic/tableaux/soundness.tex

54.4 Soundness for K

nml:tab:sou:sec

This soundness proof reuses the soundness proof for classical propositional logic, i.e., it proves everything from scratch. That's ok if you want a self-contained soundness proof. If you already have seen soundness for ordinary tableau this will be repetitive. It's planned to make it possible to switch between self-contained version and a version building on the non-modal case.

explanation

In order to show that prefixed tableaux are sound, we have to show that if

$$1\,\mathbb{T}\,\psi_1, \ldots, 1\,\mathbb{T}\,\psi_n, 1\,\mathbb{F}\,\varphi$$

has a closed tableau then $\psi_1, \ldots, \psi_n \vDash \varphi$. It is easier to prove the contrapositive: if for some $\mathfrak{M}$ and world w, $\mathfrak{M}, w \Vdash \psi_i$ for all $i = 1, \ldots, n$ but $\mathfrak{M}, w \nVdash \varphi$, then no tableau can close. Such a countermodel shows that the initial assumptions of the tableau are satisfiable. The strategy of the proof is to show that whenever all the prefixed formulas on a tableau branch are satisfiable, any application of a rule results in at least one extended branch that is also satisfiable. Since closed branches are unsatisfiable, any tableau for a satisfiable set of prefixed formulas must have at least one open branch.

In order to apply this strategy in the modal case, we have to extend our definition of "satisfiable" to modal modals and prefixes. With that in hand, however, the proof is straightforward.

Definition 54.3. Let P be some set of prefixes, i.e., $P \subseteq (\mathbb{Z}^+)^* \setminus \{\Lambda\}$ and let $\mathfrak{M}$ be a model. A function $f\colon P \to W$ is an *interpretation of* P in $\mathfrak{M}$ if, whenever σ and $\sigma.n$ are both in P, then $Rf(\sigma)f(\sigma.n)$.

Relative to an interpretation of prefixes P we can define:

1. $\mathfrak{M}$ satisfies $\sigma\,\mathbb{T}\,\varphi$ iff $\mathfrak{M}, f(\sigma) \Vdash \varphi$.

2. $\mathfrak{M}$ satisfies $\sigma\,\mathbb{F}\,\varphi$ iff $\mathfrak{M}, f(\sigma) \nVdash \varphi$.

Definition 54.4. Let Γ be a set of prefixed formulas, and let $P(\Gamma)$ be the set of prefixes that occur in it. If f is an interpretation of $P(\Gamma)$ in $\mathfrak{M}$, we say that $\mathfrak{M}$ satisfies Γ with respect to f, $\mathfrak{M}, f \Vdash \Gamma$, if $\mathfrak{M}$ satisfies every prefixed formula in Γ with respect to f. Γ is *satisfiable* iff there is a model $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that $\mathfrak{M}, f \Vdash \Gamma$.

Proposition 54.5. *If Γ contains both $\sigma\,\mathbb{T}\,\varphi$ and $\sigma\,\mathbb{F}\,\varphi$, for some formula φ and prefix σ, then Γ is unsatisfiable.*

Proof. There cannot be a model $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that both $\mathfrak{M}, f(\sigma) \Vdash \varphi$ and $\mathfrak{M}, f(\sigma) \nVdash \varphi$. □

Theorem 54.6 (Soundness). *If Γ has a closed tableau, Γ is unsatisfiable.* nml:tab:sou: thm:tableau-soundness

Proof. We call a branch of a tableau satisfiable iff the set of signed formulas on it is satisfiable, and let's call a tableau satisfiable if it contains at least one satisfiable branch.

We show the following: Extending a satisfiable tableau by one of the rules of inference always results in a satisfiable tableau. This will prove the theorem: any closed tableau results by applying rules of inference to the tableau consisting only of assumptions from Γ. So if Γ were satisfiable, any tableau for it would be satisfiable. A closed tableau, however, is clearly not satisfiable, since all its branches are closed and closed branches are unsatisfiable.

Suppose we have a satisfiable tableau, i.e., a tableau with at least one satisfiable branch. Applying a rule of inference either adds signed formulas to a branch, or splits a branch in two. If the tableau has a satisfiable branch which is not extended by the rule application in question, it remains a satisfiable branch in the extended tableau, so the extended tableau is satisfiable. So we only have to consider the case where a rule is applied to a satisfiable branch.

Let Γ be the set of signed formulas on that branch, and let $\sigma\, S\, \varphi \in \Gamma$ be the signed formula to which the rule is applied. If the rule does not result in a split branch, we have to show that the extended branch, i.e., Γ together with the conclusions of the rule, is still satisfiable. If the rule results in split branch, we have to show that at least one of the two resulting branches is satisfiable. First, we consider the possible inferences with only one premise.

1. The branch is expanded by applying $\neg\mathbb{T}$ to $\sigma\, \mathbb{T}\, \neg\psi \in \Gamma$. Then the extended branch contains the signed formulas $\Gamma \cup \{\sigma\, \mathbb{F}\, \psi\}$. Suppose $\mathfrak{M}, f \Vdash \Gamma$. In particular, $\mathfrak{M}, f(\sigma) \Vdash \neg\psi$. Thus, $\mathfrak{M}, f(\sigma) \nVdash \psi$, i.e., $\mathfrak{M}$ satisfies $\sigma\, \mathbb{F}\, \psi$ with respect to f.

2. The branch is expanded by applying $\neg\mathbb{F}$ to $\sigma\, \mathbb{F}\, \neg\psi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\wedge\mathbb{T}$ to $\sigma\, \mathbb{T}\, \psi \wedge \chi \in \Gamma$, which results in two new signed formulas on the branch: $\sigma\, \mathbb{T}\, \psi$ and $\sigma\, \mathbb{T}\, \chi$. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \Vdash \psi \wedge \chi$. Then $\mathfrak{M}, f(\sigma) \Vdash \psi$ and $\mathfrak{M}, f(\sigma) \Vdash \chi$. This means that $\mathfrak{M}$ satisfies both $\sigma\, \mathbb{T}\, \psi$ and $\sigma\, \mathbb{T}\, \chi$ with respect to f.

4. The branch is expanded by applying $\vee\mathbb{F}$ to $\mathbb{F}\, \psi \vee \chi \in \Gamma$: Exercise.

5. The branch is expanded by applying $\to\mathbb{F}$ to $\sigma\, \mathbb{F}\, \psi \to \chi \in \Gamma$: This results in two new signed formulas on the branch: $\sigma\, \mathbb{T}\, \psi$ and $\sigma\, \mathbb{F}\, \chi$. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \nVdash \psi \to \chi$. Then $\mathfrak{M}, f(\sigma) \Vdash \psi$ and $\mathfrak{M}, f(\sigma) \nVdash \chi$. This means that $\mathfrak{M}, f$ satisfies both $\sigma\, \mathbb{T}\, \psi$ and $\sigma\, \mathbb{F}\, \chi$.

6. The branch is expanded by applying $\Box\mathbb{T}$ to $\sigma\, \mathbb{T}\, \Box\psi \in \Gamma$: This results in a new signed formula $\sigma.n\, \mathbb{T}\, \psi$ on the branch, for some $\sigma.n \in P(\Gamma)$ (since $\sigma.n$ must be used). Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$. Since f is an interpretation of prefixes and both σ, $\sigma.n \in P(\Gamma)$, we know that $Rf(\sigma)f(\sigma.n)$. Hence, $\mathfrak{M}, f(\sigma.n) \Vdash \psi$, i.e., $\mathfrak{M}, f$ satisfies $\sigma.n\, \mathbb{T}\, \psi$.

7. The branch is expanded by applying $\Box\mathbb{F}$ to $\sigma\,\mathbb{F}\,\Box\psi \in \Gamma$: This results in a new signed formula $\sigma.n\,\mathbb{F}\,\varphi$, where $\sigma.n$ is a new prefix on the branch, i.e., $\sigma.n \notin P(\Gamma)$. Since Γ is satisfiable, there is a $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that $\mathfrak{M}, f \vDash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \nVdash \Box\psi$. We have to show that $\Gamma \cup \{\sigma.n\,\mathbb{F}\,\psi\}$ is satisfiable. To do this, we define an interpretation of $P(\Gamma) \cup \{\sigma.n\}$ as follows:

   Since $\mathfrak{M}, f(\sigma) \nVdash \Box\psi$, there is a $w \in W$ such that $Rf(\sigma)w$ and $\mathfrak{M}, w \nVdash \psi$. Let f' be like f, except that $f'(\sigma.n) = w$. Since $f'(\sigma) = f(\sigma)$ and $Rf(\sigma)w$, we have $Rf'(\sigma)f'(\sigma.n)$, so f' is an interpretation of $P(\Gamma)\cup\{\sigma.n\}$. Obviously $\mathfrak{M}, f'(\sigma.n) \nVdash \psi$. Since $f(\sigma') = f'(\sigma')$ for all prefixes $\sigma' \in P(\Gamma)$, $\mathfrak{M}, f' \Vdash \Gamma$. So, $\mathfrak{M}, f'$ satisfies $\Gamma \cup \{\sigma.n\,\mathbb{F}\,\psi\}$.

Now let's consider the possible inferences with two premises.

1. The branch is expanded by applying $\wedge\mathbb{F}$ to $\sigma\,\mathbb{F}\,\psi \wedge \chi \in \Gamma$, which results in two branches, a left one continuing through $\sigma\,\mathbb{F}\,\psi$ and a right one through $\sigma\,\mathbb{F}\,\chi$. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \nVdash \psi \wedge \chi$. Then $\mathfrak{M}, f(\sigma) \nVdash \psi$ or $\mathfrak{M}, f(\sigma) \nVdash \chi$. In the former case, $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{F}\,\psi$, i.e., the left branch is satisfiable. In the latter, $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{F}\,\chi$, i.e., the right branch is satisfiable.

2. The branch is expanded by applying $\vee\mathbb{T}$ to $\sigma\,\mathbb{T}\,\psi \vee \chi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\to\mathbb{T}$ to $\sigma\,\mathbb{T}\,\psi \to \chi \in \Gamma$: Exercise. □

Problem 54.2. Complete the proof of Theorem 54.6.

Corollary 54.7. *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.*

nml:tab:sou:cor:entailment-soundness

Proof. If $\Gamma \vdash \varphi$ then for some $\psi_1, \ldots, \psi_n \in \Gamma$, $\Delta = \{1\,\mathbb{F}\,\varphi, 1\,\mathbb{T}\,\psi_1, \ldots, 1\,\mathbb{T}\,\psi_n\}$ has a closed tableau. We want to show that $\Gamma \vDash \varphi$. Suppose not, so for some $\mathfrak{M}$ and w, $\mathfrak{M}, w \Vdash \psi_i$ for $i = 1, \ldots, n$, but $\mathfrak{M}, w \nVdash \varphi$. Let $f(1) = w$; then f is an interpretation of $P(\Delta)$ into $\mathfrak{M}$, and $\mathfrak{M}$ satisfies Δ with respect to f. But by Theorem 54.6, Δ is unsatisfiable since it has a closed tableau, a contradiction. So we must have $\Gamma \vdash \varphi$ after all. □

Corollary 54.8. *If $\vdash \varphi$ then φ is true in all models.*

nml:tab:sou:cor:weak-soundness

---

content/normal-modal-logic/tableaux/more-rules.tex

54.5 Rules for Other Accessibility Relations

In order to deal with logics determined by special accessibility relations, we consider the additional rules in Table 54.3.

nml:tab:mru:sec

Adding these rules results in systems that are sound and complete for the logics given in Table 54.4.

$\dfrac{\sigma\,\mathbb{T}\,\Box\varphi}{\sigma\,\mathbb{T}\,\varphi}$ T$\Box$	$\dfrac{\sigma\,\mathbb{F}\,\Diamond\varphi}{\sigma\,\mathbb{F}\,\varphi}$ T$\Diamond$
$\dfrac{\sigma\,\mathbb{T}\,\Box\varphi}{\sigma\,\mathbb{T}\,\Diamond\varphi}$ D$\Box$	$\dfrac{\sigma\,\mathbb{F}\,\Diamond\varphi}{\sigma\,\mathbb{F}\,\Box\varphi}$ D$\Diamond$
$\dfrac{\sigma.n\,\mathbb{T}\,\Box\varphi}{\sigma\,\mathbb{T}\,\varphi}$ B$\Box$	$\dfrac{\sigma.n\,\mathbb{F}\,\Diamond\varphi}{\sigma\,\mathbb{F}\,\varphi}$ B$\Diamond$
$\dfrac{\sigma\,\mathbb{T}\,\Box\varphi}{\sigma.n\,\mathbb{T}\,\Box\varphi}$ 4$\Box$ $\sigma.n$ is used	$\dfrac{\sigma\,\mathbb{F}\,\Diamond\varphi}{\sigma.n\,\mathbb{F}\,\Diamond\varphi}$ 4$\Diamond$ $\sigma.n$ is used
$\dfrac{\sigma.n\,\mathbb{T}\,\Box\varphi}{\sigma\,\mathbb{T}\,\Box\varphi}$ 4r$\Box$	$\dfrac{\sigma.n\,\mathbb{F}\,\Diamond\varphi}{\sigma\,\mathbb{F}\,\Diamond\varphi}$ 4r$\Diamond$

Table 54.3: More modal rules.

nml:tab:mru:
tab:more-rules

Logic	R is ...	Rules
T = **KT**	reflexive	T$\Box$, T$\Diamond$
D = **KD**	serial	D$\Box$, D$\Diamond$
K4	transitive	4$\Box$, 4$\Diamond$
B = **KTB**	reflexive,	T$\Box$, T$\Diamond$
	symmetric	B$\Box$, B$\Diamond$
S4 = **KT4**	reflexive,	T$\Box$, T$\Diamond$,
	transitive	4$\Box$, 4$\Diamond$
S5 = **KT4B**	reflexive,	T$\Box$, T$\Diamond$,
	transitive,	4$\Box$, 4$\Diamond$,
	euclidean	4r$\Box$, 4r$\Diamond$

Table 54.4: Tableau rules for various modal logics.

nml:tab:mru:
tab:logics-rules

Example 54.9. We give a closed tableau that shows **S5** $\vdash 5$, i.e., $\Box\varphi \to \Box\Diamond\varphi$.

1.	$1\,\mathbb{F}\ \ \Box\varphi \to \Box\Diamond\varphi$	Assumption
2.	$1\,\mathbb{T}\ \ \Box\varphi$	$\to\mathbb{F}\,1$
3.	$1\,\mathbb{F}\ \ \Box\Diamond\varphi$	$\to\mathbb{F}\,1$
4.	$1.1\,\mathbb{F}\ \ \Diamond\varphi$	$\Box\mathbb{F}\,3$
5.	$1\,\mathbb{F}\ \ \Diamond\varphi$	$4\mathrm{r}\Diamond\ 4$
6.	$1.1\,\mathbb{F}\ \ \varphi$	$\Diamond\mathbb{F}\,5$
7.	$1.1\,\mathbb{T}\ \ \varphi$	$\Box\mathbb{T}\,2$
	$\otimes$	

Problem 54.3. Give closed tableaux that show the following:

1. **KT5** $\vdash$ B;
2. **KT5** $\vdash$ 4;
3. **KDB4** $\vdash$ T;
4. **KB4** $\vdash$ 5;
5. **KB5** $\vdash$ 4;
6. **KT** $\vdash$ D.

---

content/normal-modal-logic/tableaux/more-soundness.tex

54.6 Soundness for Additional Rules

We say a rule is sound for a class of models if, whenever a branch in a tableau is satisfiable in a model from that class, the branch resulting from applying the rule is also satisfiable in a model from that class.

nml:tab:msn:sec

Proposition 54.10. T$\Box$ *and* T$\Diamond$ *are sound for reflexive models.*

nml:tab:msn:prop:soundness-T

Proof.

1. The branch is expanded by applying T$\Box$ to $\sigma\,\mathbb{T}\,\Box\psi \in \Gamma$: This results in a new signed formula $\sigma\,\mathbb{T}\,\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$. Since R is reflexive, we know that $Rf(\sigma)f(\sigma)$. Hence, $\mathfrak{M}, f(\sigma) \Vdash \psi$, i.e., $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{T}\,\psi$.

2. The branch is expanded by applying T$\Diamond$ to $\sigma\,\mathbb{F}\,\Diamond\psi \in \Gamma$: This results in a new signed formula $\sigma\,\mathbb{F}\,\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \nVdash \Diamond\psi$. Since R is reflexive, we know that $Rf(\sigma)f(\sigma)$. Hence, $\mathfrak{M}, f(\sigma) \nVdash \psi$, i.e., $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{F}\,\psi$. □

Proposition 54.11. D$\Box$ *and* D$\Diamond$ *are sound for serial models.*

nml:tab:msn:prop:soundness-D

Proof. 1. The branch is expanded by applying D$\Box$ to $\sigma\,\mathbb{T}\,\Box\psi \in \Gamma$: This results in a new signed formula $\sigma\,\mathbb{T}\,\Diamond\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$. Since R is serial, there is a $w \in W$ such that $Rf(\sigma)w$. Then $\mathfrak{M}, w \Vdash \psi$, and hence $\mathfrak{M}, f(\sigma) \Vdash \Diamond\psi$. So, $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{T}\,\Diamond\psi$.

2. The branch is expanded by applying D$\Diamond$ to $\sigma\,\mathbb{F}\,\Diamond\psi \in \Gamma$: This results in a new signed formula $\sigma\,\mathbb{F}\,\Box\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \nVdash \Diamond\psi$. Since R is serial, there is a $w \in W$ such that $Rf(\sigma)w$. Then $\mathfrak{M}, w \nVdash \psi$, and hence $\mathfrak{M}, f(\sigma) \nVdash \Box\psi$. So, $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{F}\,\Box\psi$. □

nml:tab:msn: prop:soundness-B

Proposition 54.12. B$\Box$ *and* B$\Diamond$ *are sound for symmetric models.*

Proof. 1. The branch is expanded by applying B$\Box$ to $\sigma.n\,\mathbb{T}\,\Box\psi \in \Gamma$: This results in a new signed formula $\sigma\,\mathbb{T}\,\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma.n) \Vdash \Box\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$, we know that $Rf(\sigma)f(\sigma.n)$. Since R is symmetric, $Rf(\sigma.n)f(\sigma)$. Since $\mathfrak{M}, f(\sigma.n) \Vdash \Box\psi$, $\mathfrak{M}, f(\sigma) \Vdash \psi$. Hence, $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{T}\,\psi$.

2. The branch is expanded by applying B$\Diamond$ to $\sigma.n\,\mathbb{F}\,\Diamond\psi \in \Gamma$: This results in a new signed formula $\sigma\,\mathbb{F}\,\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma.n) \nVdash \Diamond\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$, we know that $Rf(\sigma)f(\sigma.n)$. Since R is symmetric, $Rf(\sigma.n)f(\sigma)$. Since $\mathfrak{M}, f(\sigma.n) \nVdash \Diamond\psi$, $\mathfrak{M}, f(\sigma) \nVdash \psi$. Hence, $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{F}\,\psi$. □

nml:tab:msn: prop:soundness-4

Proposition 54.13. 4$\Box$ *and* 4$\Diamond$ *are sound for transitive models.*

Proof. 1. The branch is expanded by applying 4$\Box$ to $\sigma\,\mathbb{T}\,\Box\psi \in \Gamma$: This results in a new signed formula $\sigma.n\,\mathbb{T}\,\Box\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$ and $\sigma.n$ must be used, we know that $Rf(\sigma)f(\sigma.n)$. Now let w be any world such that $Rf(\sigma.n)w$. Since R is transitive, $Rf(\sigma)w$. Since $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$, $\mathfrak{M}, w \Vdash \psi$. Hence, $\mathfrak{M}, f(\sigma.n) \Vdash \Box\psi$, and $\mathfrak{M}, f$ satisfies $\sigma.n\,\mathbb{T}\,\Box\psi$.

2. The branch is expanded by applying 4$\Diamond$ to $\sigma\,\mathbb{F}\,\Diamond\psi \in \Gamma$: This results in a new signed formula $\sigma.n\,\mathbb{F}\,\Diamond\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \nVdash \Diamond\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$ and $\sigma.n$ must be used, we know that $Rf(\sigma)f(\sigma.n)$. Now let w be any world such that $Rf(\sigma.n)w$. Since R is transitive, $Rf(\sigma)w$. Since $\mathfrak{M}, f(\sigma) \nVdash \Diamond\psi$, $\mathfrak{M}, w \nVdash \psi$. Hence, $\mathfrak{M}, f(\sigma.n) \nVdash \Diamond\psi$, and $\mathfrak{M}, f$ satisfies $\sigma.n\,\mathbb{F}\,\Diamond\psi$. □

Proposition 54.14. 4r$\Box$ *and* 4r$\Diamond$ *are sound for euclidean models.* nml:tab:msn: prop:soundness-4r

Proof. 1. The branch is expanded by applying 4r$\Box$ to $\sigma.n\,\mathbb{T}\,\Box\psi \in \Gamma$: This results in a new signed formula $\sigma\,\mathbb{T}\,\Box\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma.n) \Vdash \Box\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$, we know that $Rf(\sigma)f(\sigma.n)$. Now let w be any world such that $Rf(\sigma)w$. Since R is euclidean, $Rf(\sigma.n)w$. Since $\mathfrak{M}, f(\sigma).n \Vdash \Box\psi$, $\mathfrak{M}, w \Vdash \psi$. Hence, $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$, and $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{T}\,\Box\psi$.

2. The branch is expanded by applying 4r$\Diamond$ to $\sigma.n\,\mathbb{F}\,\Diamond\psi \in \Gamma$: This results in a new signed formula $\sigma\,\mathbb{T}\,\Box\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma.n) \nVdash \Diamond\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$, we know that $Rf(\sigma)f(\sigma.n)$. Now let w be any world such that $Rf(\sigma)w$. Since R is euclidean, $Rf(\sigma.n)w$. Since $\mathfrak{M}, f(\sigma).n \nVdash \Diamond\psi$, $\mathfrak{M}, w \nVdash \psi$. Hence, $\mathfrak{M}, f(\sigma) \nVdash \Diamond\psi$, and $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{F}\,\Diamond\psi$. □

Corollary 54.15. *The tableau systems given in Table 54.4 are sound for the respective classes of models.* nml:tab:msn: cor:soundness-logics

---

content/normal-modal-logic/tableaux/simple-S5.tex

54.7 Simple Tableaux for S5

S5 is sound and complete with respect to the class of universal models, i.e., models where every world is accessible from every world. In universal models the accessibility relation doesn't matter: "there is a world w where $\mathfrak{M}, w \Vdash \varphi$" is true if and only if there is such a w that's accessible from u. So in **S5**, we can define models as simply a set of worlds and a valuation V. This suggests that we should be able to simplify the tableau rules as well. In the general case, we take as prefixes sequences of positive integers, so that we can keep track of which such prefixes name worlds which are accessible from others: $\sigma.n$ names a world accessible from σ. But in **S5** any world is accessible from any world, so there is no need to so keep track. Instead, we can use positive integers as prefixes. The simplified rules are given in Table 54.5. nml:tab:s5: sec

Example 54.16. We give a simplified closed tableau that shows **S5** $\vdash 5$, i.e., $\Diamond\varphi \to \Box\Diamond\varphi$.

1.	$1\,\mathbb{F}\ \ \Diamond\varphi \to \Box\Diamond\varphi$	Assumption
2.	$1\,\mathbb{T}\ \ \Diamond\varphi$	$\to\mathbb{F}\,1$
3.	$1\,\mathbb{F}\ \ \Box\Diamond\varphi$	$\to\mathbb{F}\,1$
4.	$2\,\mathbb{F}\ \ \Diamond\varphi$	$\Box\mathbb{F}\,3$
5.	$3\,\mathbb{T}\ \ \varphi$	$\Diamond\mathbb{T}\,2$
6.	$3\,\mathbb{F}\ \ \varphi$	$\Diamond\mathbb{F}\,4$
	$\otimes$	

$\dfrac{n\,\mathbb{T}\,\Box\varphi}{m\,\mathbb{T}\,\varphi}\,\Box\mathbb{T}$ m is used	$\dfrac{n\,\mathbb{F}\,\Box\varphi}{m\,\mathbb{F}\,\varphi}\,\Box\mathbb{F}$ m is new
$\dfrac{n\,\mathbb{T}\,\Diamond\varphi}{m\,\mathbb{T}\,\varphi}\,\Diamond\mathbb{T}$ m is new	$\dfrac{n\,\mathbb{F}\,\Diamond\varphi}{m\,\mathbb{F}\,\varphi}\,\Diamond\mathbb{F}$ m is used

Table 54.5: Simplified rules for **S5**.

nml:tab:s5:tab:rules-S5

content/normal-modal-logic/tableaux/completeness.tex

54.8 Completeness for K

nml:tab:cpl:sec

To show that the method of tableaux is complete, we have to show that whenever there is no closed tableau to show $\Gamma \vdash \varphi$, then $\Gamma \nvdash \varphi$, i.e., there is a countermodel. But "there is no closed tableau" means that every way we could try to construct one has to fail to close. The trick is to see that if every such way fails to close, then a specific, *systematic and exhaustive* way also fails to close. And this systematic and exhaustive way would close if a closed tableau exists. The single tableau will contain, among its open branches, all the information required to define a countermodel. The countermodel given by an open branch in this tableau will contain the all the prefixes used on that branch as the worlds, and a propositional variable p is true at σ iff $\sigma\,\mathbb{T}\,p$ occurs on the branch. explanation

Definition 54.17. A branch in a tableau is called complete if, whenever it contains a prefixed formula $\sigma\,S\,\varphi$ to which a rule can be applied, it also contains

1. the prefixed formulas that are the corresponding conclusions of the rule, in the case of propositional stacking rules;
2. one of the corresponding conclusion formulas in the case of propositional branching rules;
3. at least one possible conclusion in the case of modal rules that require a new prefix;
4. the corresponding conclusion for every prefix occurring on the branch in the case of modal rules that require a used prefix.

explanation

For instance, a complete branch contains $\sigma\,\mathbb{T}\,\psi$ and $\sigma\,\mathbb{T}\,\chi$ whenever it contains $\mathbb{T}\,\psi \wedge \chi$. If it contains $\sigma\,\mathbb{T}\,\psi \vee \chi$ it contains at least one of $\sigma\,\mathbb{F}\,\psi$ and $\sigma\,\mathbb{T}\,\chi$. If it contains $\sigma\,\mathbb{F}\,\Box$ it also contains $\sigma.n\,\mathbb{F}\,\Box$ for at least one n. And whenever it contains $\sigma\,\mathbb{T}\,\Box$ it also contains $\sigma.n\,\mathbb{T}\,\Box$ for every n such that $\sigma.n$ is used on the branch.

Proposition 54.18. *Every finite Γ has a tableau in which every branch is complete.*

nml:tab:cpl:prop:complete-tableau

Proof. Consider an open branch in a tableau for Γ. There are finitely many prefixed formulas in the branch to which a rule could be applied. In some fixed order (say, top to bottom), for each of these prefixed formulas for which the conditions (1)–(4) do not already hold, apply the rules that can be applied to it to extend the branch. In some cases this will result in branching; apply the rule at the tip of each resulting branch for all remaining prefixed formulas. Since the number of prefixed formulas is finite, and the number of used prefixes on the branch is finite, this procedure eventually results in (possibly many) branches extending the original branch. Apply the procedure to each, and repeat. But by construction, every branch is closed. □

Theorem 54.19 (Completeness). *If Γ has no closed tableau, Γ is satisfiable.*

nml:tab:cpl:thm:tableau-completeness

Proof. By the proposition, Γ has a tableau in which every branch is complete. Since it has no closed tableau, it thas has a tableau in which at least one branch is open and complete. Let Δ be the set of prefixed formulas on the branch, and $P(\Delta)$ the set of prefixes occurring in it.

We define a model $\mathfrak{M}(\Delta) = \langle P(\Delta), R, V\rangle$ where the worlds are the prefixes occurring in Δ, the accessibility relation is given by:

$$R\sigma\sigma' \quad \text{iff} \quad \sigma' = \sigma.n \quad \text{for some } n$$

and

$$V(p) = \{\sigma : \sigma\,\mathbb{T}\,p \in \Delta\}.$$

We show by induction on φ that if $\sigma\,\mathbb{T}\,\varphi \in \Delta$ then $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$, and if $\sigma\,\mathbb{F}\,\varphi \in \Delta$ then $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

1. $\varphi \equiv p$: If $\sigma\,\mathbb{T}\,\varphi \in \Delta$ then $\sigma \in V(p)$ (by definition of V) and so $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

   If $\sigma\,\mathbb{F}\,\varphi \in \Delta$ then $\sigma\,\mathbb{T}\,\varphi \notin \Delta$, since the branch would otherwise be closed. So $\sigma \notin V(p)$ and thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

2. $\varphi \equiv \neg\psi$: If $\sigma\,\mathbb{T}\,\varphi \in \Delta$, then $\sigma\,\mathbb{F}\,\psi \in \Delta$ since the branch is complete. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma \nVdash \psi$ and thus $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

   If $\sigma\,\mathbb{F}\,\varphi \in \Delta$, then $\sigma\,\mathbb{T}\,\psi \in \Delta$ since the branch is complete. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ and thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

3. $\varphi \equiv \psi \wedge \chi$: If $\sigma\,\mathbb{T}\,\varphi \in \Delta$, then both $\sigma\,\mathbb{T}\,\psi \in \Delta$ and $\sigma\,\mathbb{T}\,\chi \in \Delta$ since the branch is complete. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ and $\mathfrak{M}(\Delta), \sigma \Vdash \chi$. Thus $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

   If $\sigma\,\mathbb{F}\,\varphi \in \Delta$, then either $\sigma\,\mathbb{F}\,\psi \in \Delta$ or $\sigma\,\mathbb{F}\,\chi \in \Delta$ since the branch is complete. By induction hypothesis, either $\mathfrak{M}(\Delta), \sigma \nVdash \psi$ or $\mathfrak{M}(\Delta), \sigma \nVdash \psi$. Thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

4. $\varphi \equiv \psi \vee \chi$: If $\sigma\,\mathbb{T}\,\varphi \in \Delta$, then either $\sigma\,\mathbb{T}\,\psi \in \Delta$ or $\sigma\,\mathbb{T}\,\chi \in \Delta$ since the branch is complete. By induction hypothesis, either $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ or $\mathfrak{M}(\Delta), \sigma \Vdash \chi$. Thus $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

   If $\sigma\,\mathbb{F}\,\varphi \in \Delta$, then both $\sigma\,\mathbb{F}\,\psi \in \Delta$ and $\sigma\,\mathbb{F}\,\chi \in \Delta$ since the branch is complete. By induction hypothesis, both $\mathfrak{M}(\Delta), \sigma \nVdash \psi$ and $\mathfrak{M}(\Delta), \sigma \nVdash \psi$. Thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

5. $\varphi \equiv \psi \to \chi$: If $\sigma\,\mathbb{T}\,\varphi \in \Delta$, then either $\sigma\,\mathbb{F}\,\psi \in \Delta$ or $\sigma\,\mathbb{T}\,\chi \in \Delta$ since the branch is complete. By induction hypothesis, either $\mathfrak{M}(\Delta), \sigma \nVdash \psi$ or $\mathfrak{M}(\Delta), \sigma \Vdash \chi$. Thus $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

   If $\sigma\,\mathbb{F}\,\varphi \in \Delta$, then both $\sigma\,\mathbb{T}\,\psi \in \Delta$ and $\sigma\,\mathbb{F}\,\chi \in \Delta$ since the branch is complete. By induction hypothesis, both $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ and $\mathfrak{M}(\Delta), \sigma \nVdash \psi$. Thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

6. $\varphi \equiv \Box\psi$: If $\sigma\,\mathbb{T}\,\varphi \in \Delta$, then, since the branch is complete, $\sigma.n\,\mathbb{T}\,\psi \in \Delta$ for every $\sigma.n$ used on the branch, i.e., for every $\sigma' \in P(\Delta)$ such that $R\sigma\sigma'$. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma' \Vdash \psi$ for every σ' such that $R\sigma\sigma'$. Therefore, $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

   If $\sigma\,\mathbb{F}\,\varphi \in \Delta$, then for some $\sigma.n$, $\sigma.n\,\mathbb{F}\,\psi \in \Delta$ since the branch is complete. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma.n \nVdash \psi$. Since $R\sigma(\sigma.n)$, there is a σ' such that $\mathfrak{M}(\Delta), \sigma' \nVdash \psi$. Thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

7. $\varphi \equiv \Diamond\psi$: If $\sigma\,\mathbb{T}\,\varphi \in \Delta$, then for some $\sigma.n$, $\sigma.n\,\mathbb{T}\,\psi \in \Delta$ since the branch is complete. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma.n \Vdash \psi$. Since $R\sigma(\sigma.n)$, there is a σ' such that $\mathfrak{M}(\Delta), \sigma' \Vdash \psi$. Thus $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

   If $\sigma\,\mathbb{F}\,\varphi \in \Delta$, then, since the branch is complete, $\sigma.n\,\mathbb{F}\,\psi \in \Delta$ for every $\sigma.n$ used on the branch, i.e., for every $\sigma' \in P(\Delta)$ such that $R\sigma\sigma'$. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma' \nVdash \psi$ for every σ' such that $R\sigma\sigma'$. Therefore, $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

Since $\Gamma \subseteq \Delta$, $\mathfrak{M}(\Delta) \Vdash \Gamma$. □

Problem 54.4. Complete the proof of Theorem 54.19.

nml:tab:cpl: cor:entailment-completeness

Corollary 54.20. *If $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$.*

nml:tab:cpl: cor:weak-completeness

Corollary 54.21. *If φ is true in all models, then $\vdash \varphi$.*

54.9 Countermodels from Tableaux

explanation

The proof of the completeness theorem doesn't just show that if $\vDash \varphi$ then $\vdash \varphi$, it also gives us a method for constructing countermodels to φ if $\nvDash A$. In the case of **K**, this method constitutes a *decision procedure*. For suppose $\nvDash \varphi$. Then the proof of Proposition 54.18 gives a method for constructing a complete tableau. The method in fact always terminates. The propositional rules for **K** only add prefixed formulas of lower complexity, i.e., each propositional rule need only be applied once on a branch for any signed formula $\sigma S \varphi$. New prefixes are only generated by the $\Box\mathbb{F}$ and $\Diamond\mathbb{T}$ rules, and also only have to be applied once (and produce a single new prefix). $\Box\mathbb{T}$ and $\Diamond\mathbb{F}$ have to be applied potentially multiple times, but only once per prefix, and only finitely many new prefixes are generated. So the construction either results in a closed branch or a complete branch after finitely many stages.

nml:tab:cou:sec

Once a tableau with an open complete branch is constructed, the proof of Theorem 54.19 gives us an explict model that satisfies the original set of prefixed formulas. So not only is it the case that if $\Gamma \vDash \varphi$, then a closed tableau exists and $\Gamma \vdash \varphi$, if we look for the closed tableau in the right way and end up with a "complete" tableau, we'll not only know that $\Gamma \nvDash \varphi$ but actually be able to construct a countermodel.

Example 54.22. We know that $\nvdash \Box(p \vee q) \to (\Box p \vee \Box q)$. The construction of a tableau begins with:

1.	$1\,\mathbb{F}\ \ \Box(p \vee q) \to (\Box p \vee \Box q)$ ✓	Assumption
2.	$1\,\mathbb{T}\ \ \Box(p \vee q)$	$\to\mathbb{F}\,1$
3.	$1\,\mathbb{F}\ \ \Box p \vee \Box q$ ✓	$\to\mathbb{F}\,1$
4.	$1\,\mathbb{F}\ \ \Box p$ ✓	$\vee\mathbb{F}\,3$
5.	$1\,\mathbb{F}\ \ \Box q$ ✓	$\vee\mathbb{F}\,3$
6.	$1.1\,\mathbb{F}\ \ p$ ✓	$\Box\mathbb{F}\,4$
7.	$1.2\,\mathbb{F}\ \ q$ ✓	$\Box\mathbb{F}\,5$

The tableau is of course not finished yet. In the next step, we consider the only line without a checkmark: the prefixed formula $1\,\mathbb{T}\Box(p \vee q)$ on line 2. The construction of the closed tableau says to apply the $\Box\mathbb{T}$ rule for every prefix used on the branch, i.e., for both 1.1 and 1.2:

1.	$1\,\mathbb{F}\ \ \Box(p \vee q) \to (\Box p \vee \Box q)$ ✓	Assumption
2.	$1\,\mathbb{T}\ \ \Box(p \vee q)$	$\to\mathbb{F}\,1$
3.	$1\,\mathbb{F}\ \ \Box p \vee \Box q$ ✓	$\to\mathbb{F}\,1$
4.	$1\,\mathbb{F}\ \ \Box p$ ✓	$\vee\mathbb{F}\,3$
5.	$1\,\mathbb{F}\ \ \Box q$ ✓	$\vee\mathbb{F}\,3$
6.	$1.1\,\mathbb{F}\ \ p$ ✓	$\Box\mathbb{F}\,4$
7.	$1.2\,\mathbb{F}\ \ q$ ✓	$\Box\mathbb{F}\,5$
8.	$1.1\,\mathbb{T}\ \ p \vee q$	$\Box\mathbb{T}\,2$
9.	$1.2\,\mathbb{T}\ \ p \vee q$	$\Box\mathbb{T}\,2$

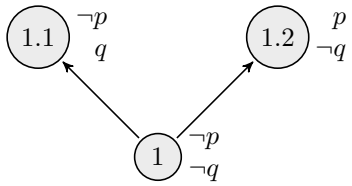


Figure 54.1: A countermodel to $\Box(p \vee q) \to (\Box p \vee \Box q)$.

nml:tab:cou:
fig:counter-Box

Now lines 2, 8, and 9, don't have checkmarks. But no new prefix has been added, so we apply $\vee\mathbb{T}$ to lines 8 and 9, on all resulting branches (as long as they don't close):

1.	$1\,\mathbb{F}\ \ \Box(p \vee q) \to (\Box p \vee \Box q)$ ✓	Assumption
2.	$1\,\mathbb{T}\ \ \Box(p \vee q)$ ✓	$\to\mathbb{F}\,1$
3.	$1\,\mathbb{F}\ \ \Box p \vee \Box q$ ✓	$\to\mathbb{F}\,1$
4.	$1\,\mathbb{F}\ \ \Box p$ ✓	$\vee\mathbb{F}\,3$
5.	$1\,\mathbb{F}\ \ \Box q$ ✓	$\vee\mathbb{F}\,3$
6.	$1.1\,\mathbb{F}\ \ p$ ✓	$\Box\mathbb{F}\,4$
7.	$1.2\,\mathbb{F}\ \ q$ ✓	$\Box\mathbb{F}\,5$
8.	$1.1\,\mathbb{T}\ \ p \vee q$ ✓	$\Box\mathbb{T}\,2$
9.	$1.2\,\mathbb{T}\ \ p \vee q$ ✓	$\Box\mathbb{T}\,2$
10.	$1.1\,\mathbb{T}\ \ p$ ✓ ⊗ | $1.1\,\mathbb{T}\ \ q$ ✓	$\vee\mathbb{T}\,8$
11.	$1.2\,\mathbb{T}\ \ p$ ✓ | $1.2\,\mathbb{T}\ \ q$ ✓ ⊗	$\vee\mathbb{T}\,9$

There is one remaining open branch, and it is complete. From it we define the model with worlds $W = \{1, 1.1, 1.2\}$ (the only prefixes appearing on the open branch), the accessibility relation $R = \{\langle 1, 1.1\rangle, \langle 1, 1.2\rangle\}$, and the assignment $V(p) = \{1.2\}$ (because line 11 contains $1.2\,\mathbb{T}\,p$) and $V(q) = \{1.1\}$ (because line 10 contains $1.1\,\mathbb{T}\,q$). The model is pictured in Figure 54.1, and you can verify that it is a countermodel to $\Box(p \vee q) \to (\Box p \vee \Box q)$.

Chapter 55

Modal Sequent Calculus

Draft chapter on sequent calculi for modal logic. Needs more examples, soundness and completeness proofs.

---

content/normal-modal-logic/sequent-calculus/introduction.tex

55.1 Introduction

The sequent calculus for propositional logic can be extended by additional rules that deal with $\Box$ and $\Diamond$. For instance, for **K**, we have **LK** plus: nml:seq:int:sec

$$\frac{\Gamma \Rightarrow \Delta, \varphi}{\Box\Gamma \Rightarrow \Diamond\Delta, \Box\varphi}\,\Box \qquad \frac{\varphi, \Gamma \Rightarrow \Delta}{\Diamond\varphi, \Box\Gamma \Rightarrow \Diamond\Delta}\,\Diamond$$

For extensions of **K**, additional rules have to be added as well.

Not every modal logic has such a sequent calculus. Even **S5**, which is semantically simple (it can be defined without using accessibility relations at all) is not known to have a sequent calculus that results from **LK** which is complete without the rule Cut. However, it has a cut-free complete *hypersequent* calculus.

---

content/normal-modal-logic/sequent-calculus/rules-for-K.tex

55.2 Rules for K

The rules for the regular propositional connectives are the same as for regular sequent calculus **LK**. Axioms are also the same: any sequent of the form $\varphi \Rightarrow \varphi$ counts as an axiom. nml:seq:rul:sec

For the modal operators $\Box$ and $\Diamond$, we have the following additional rules:

$$\frac{\Gamma \Rightarrow \Delta, \varphi}{\Box\Gamma \Rightarrow \Diamond\Delta, \Box\varphi}\,\Box \qquad \frac{\varphi, \Gamma \Rightarrow \Delta}{\Diamond\varphi, \Box\Gamma \Rightarrow \Diamond\Delta}\,\Diamond$$

Here, $\Box\Gamma$ means the sequence of formulas resulting from Γ by putting $\Box$ in front of every formula in Γ and $\Diamond\Delta$ is the sequence of formulas resulting from Δ by putting $\Diamond$ in front of every formula in Δ. Γ and Δ may be empty; in that case the corresponding part $\Box\Gamma$ and $\Diamond\Delta$ of the conclusion sequent is empty as well.

The restriction of adding a $\Box$ on the right and $\Diamond$ on the left to a single formula φ is necessary. If we allowed to add $\Box$ to any number of formulas on the right or to add $\Diamond$ to any number of formulas on the left we would be able to derive:

$$\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg\varphi}\,\neg\mathrm{R}}{\Rightarrow \Box\varphi, \Box\neg\varphi}\,\Box*}{\Rightarrow \Box\varphi \vee \Box\neg\varphi}\,\vee\mathrm{R} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow}\,\neg\mathrm{L}}{\Diamond\neg\varphi, \Diamond\varphi \Rightarrow}\,\Diamond*}{\Diamond\varphi \Rightarrow \neg\Diamond\neg\varphi}\,\neg\mathrm{R}}{\Rightarrow \Diamond\varphi \to \neg\Diamond\neg\varphi}\,\to\mathrm{R}$$

But $\Box\varphi \vee \Box\neg\varphi$ and $\Diamond\varphi \to \neg\Diamond\neg\varphi$ are not valid in **K**.

If we allowed side formulas in addition to φ in the premise, and allowed the $\Box$ rule to add $\Box$ to only φ on the right, or allowed the $\Diamond$ rule to add $\Diamond$ to only φ on the left (but do nothing to the side formulas) we would be able to derive:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg\varphi}\,\neg\mathrm{R}}{\Rightarrow \neg\varphi, \varphi}\,\mathrm{XR}}{\Rightarrow \neg\varphi, \Box\varphi}\,\Box*}{\Rightarrow \neg\varphi \vee \Box\varphi}\,\vee\mathrm{R} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow}\,\neg\mathrm{L}}{\Diamond\neg\varphi, \varphi \Rightarrow}\,\Diamond*}{\varphi \Rightarrow \neg\Diamond\neg\varphi}\,\neg\mathrm{R}}{\Rightarrow \varphi \to \neg\Diamond\neg\varphi}\,\to\mathrm{R}$$

But $\neg\varphi \vee \Box\varphi$ (which is equivalent to $\varphi \to \Box\varphi$) and $\varphi \to \neg\Diamond\neg\varphi$ are not valid in **K**.

---

content/normal-modal-logic/sequent-calculus/proofs-in-K.tex

55.3 Sequent Derivations for K

nml:seq:prk:sec

Example 55.1. We give a sequent calculus derivation that shows $\vdash (\Box\varphi \wedge \Box\psi) \to \Box(\varphi \wedge \psi)$.

$$\dfrac{\dfrac{\dfrac{

$$\frac{\varphi, \Gamma \Rightarrow \Delta}{\Box\varphi, \Gamma \Rightarrow \Delta}\ \mathrm{T}\Box \qquad \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \Diamond\varphi}\ \mathrm{T}\Diamond$$

$$\frac{\Gamma \Rightarrow \Delta}{\Box\Gamma \Rightarrow \Diamond\Delta}\ \mathrm{D}$$

$$\frac{\Gamma, \Diamond\Pi \Rightarrow \Box\Delta, \Lambda, \varphi}{\Box\Gamma, \Pi \Rightarrow \Delta, \Diamond\Lambda, \Box\varphi}\ \mathrm{B}\Box \qquad \frac{\varphi, \Diamond\Gamma, \Pi \Rightarrow \Box\Lambda, \Delta}{\Diamond\varphi, \Gamma, \Box\Pi \Rightarrow \Lambda, \Diamond\Delta}\ \mathrm{B}\Diamond$$

$$\frac{\Box\Gamma \Rightarrow \Diamond\Delta, \varphi}{\Box\Gamma \Rightarrow \Diamond\Delta, \Box\varphi}\ 4\Box \qquad \frac{\varphi, \Box\Gamma \Rightarrow \Diamond\Delta}{\Diamond\varphi, \Box\Gamma \Rightarrow \Diamond\Delta}\ 4\Diamond$$

$$\frac{\Box\Gamma, \Diamond\Pi \Rightarrow \Box\Delta, \Diamond\Lambda, \varphi}{\Box\Gamma, \Diamond\Pi \Rightarrow \Box\Delta, \Diamond\Lambda, \Box\varphi}\ 5\Box \qquad \frac{\varphi, \Diamond\Gamma, \Box\Pi \Rightarrow \Diamond\Delta, \Box\Lambda}{\Diamond\varphi, \Diamond\Gamma, \Box\Pi \Rightarrow \Diamond\Delta, \Box\Lambda}\ 5\Diamond$$

Table 55.1: More modal rules.

nml:seq:mru:tab:more-rules

Logic	R is ...	Rules
T = **KT**	reflexive	$\Box$, T$\Box$, T$\Diamond$
D = **KD**	serial	$\Box$, D
K4	transitive	$\Box$, 4$\Box$, 4$\Diamond$
B = **KTB**	reflexive, symmetric	$\Box$, T$\Box$, T$\Diamond$ B$\Box$, B$\Diamond$
S4 = **KT4**	reflexive, transitive	$\Box$, T$\Box$, T$\Diamond$ 4$\Box$, 4$\Diamond$
S5 = **KT5**	reflexive, transitive, euclidean	$\Box$, T$\Box$, T$\Diamond$ 5$\Box$, 5$\Diamond$

Table 55.2: Sequent rules for various modal logics.

nml:seq:mru:tab:logics-rules

55.4 Rules for Other Accessibility Relations

nml:seq:mru:sec

In order to deal with logics determined by special accessibility relations, we consider the additional rules in Table 55.1.

Adding these rules results in systems that are sound and complete for the logics given in Table 55.2.

Example 55.3. We give a sequent derivation that shows **K4** $\vdash$ 4, i.e., $\Box\varphi \to \Box\Box\varphi$.

$$\dfrac{\dfrac{\Box\varphi \Rightarrow \Box\varphi}{\Box\varphi \Rightarrow \Box\Box\varphi}\,4\Box}{\Rightarrow \Box\varphi \to \Box\Box\varphi}\,{\to}\mathrm{R}$$

Example 55.4. We give a sequent derivation that shows $\mathbf{S5} \vdash 5$, i.e., $\Diamond\varphi \to \Box\Diamond\varphi$.

$$\dfrac{\dfrac{\Diamond\varphi \Rightarrow \Diamond\varphi}{\Diamond\varphi \Rightarrow \Box\Diamond\varphi}\,5\Box}{\Rightarrow \Diamond\varphi \to \Box\Diamond\varphi}\,{\to}\mathrm{R}$$

Example 55.5. The sequent calculus for **S5** is not complete without the Cut rule; e.g., $\Diamond\Box\varphi \to \varphi$, which is valid in **S5**, has no proof without Cut. Here is a derivation using Cut:

$$\dfrac{\dfrac{\dfrac{\Box\varphi \Rightarrow \Box\varphi}{\Diamond\Box\varphi \Rightarrow \Box\varphi}\,5\Diamond \qquad \dfrac{\varphi \Rightarrow \varphi}{\Box\varphi \Rightarrow \varphi}\,\mathrm{T}\Box}{\Diamond\Box\varphi \Rightarrow \varphi}\,\mathrm{Cut}}{\Rightarrow \Diamond\Box\varphi \to \varphi}\,{\to}\mathrm{R}$$

Problem 55.2. Give sequent derivations that show the following:

1. **KT5** $\vdash$ B;
2. **KT5** $\vdash$ 4;
3. **KDB4** $\vdash$ T;
4. **KB4** $\vdash$ 5;
5. **KB5** $\vdash$ 4;
6. **KT** $\vdash$ D.

Part XII

Applied Modal Logic

This part contains experimental draft material on some applications of modal logic, such as temporal and epistemic logics.

Chapter 56

Temporal Logics

This chapter covers temporal logics.

content/applied-modal-logic/temporal-logic/introduction.tex

56.1 Introduction

aml:tl:int:sec

Temporal logics deal with claims about things that will or have been the case. Arthur Prior is credited as the originator of temporal logic, which he called *tense logic*. Our treatment of temporal logic here will largely follow Prior's original modal treatment of introducing temporal operators into the basic framework of propositional logic, which treats claims as generally lacking in tense.

For example, in propositional logic, I might talk about a dog, Beezie, who sometimes sits and sometimes doesn't sit, as dogs are wont to do. It would be contradictory in classical logic to claim that Beezie is sitting and also that Beezie is not sitting. But obviously both can be true, just not at the same time; adding temporal operators to the language can allow us to express that claim relatively easily. The addition of temporal operators also allows us to account for the validity of inferences like the one from "Beezie will get a treat or a ball" to "Beezie will get a treat or Beezie will get a ball."

However, a lot of philosophical issues arise with temporal logic that might lead us to adopt one framework of temporal logic over another. For example, a future contingent is a statement about the future that is neither necessary nor impossible. If we say "Richard will go to the grocery store tomorrow," we are expressing a claim about something that has not yet happened, and whose truth value is contestable. In fact, it is contestable whether that claim can even be *assigned* a truth value in the first place. If we are strict determinists, then perhaps we can be comfortable with the idea that this sentence is in fact true or false, even before the event in question is supposed to take place—it just may be that we do not know its truth value yet. In contrast, we might believe in a genuinely open future, in which the truth values of future contingents are undetermined.

As it turns out, a lot of these commitments about the structure and nature of time are built in to our choices of models and frameworks of temporal logics. For example, we might ask ourselves whether we should construct models in which time is linear, branching or even circular. We might have to make decisions about whether our temporal models will have beginning and end points, and whether time is to be represented using discrete instants or as a continuum.

---

content/applied-modal-logic/temporal-logic/temporal-logic-semantics.tex

56.2 Semantics for Temporal Logic

aml:tl:sem:sec

Definition 56.1. The basic language of temporal logic contains

1. The propositional constant for falsity $\bot$.
2. The propositional constant for truth $\top$.
3. A denumerable set of propositional variables: p_0, p_1, p_2, ...
4. The propositional connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (conditional), $\leftrightarrow$ (biconditional).
5. Past operators P and H.
6. Future operators F and G.

Later on, we will discuss the potential addition of other kinds of modal operators.

Definition 56.2. *Formulas* of the temporal language are inductively defined as follows:

1. $\bot$ is an atomic formula.
2. $\top$ is an atomic formula.

3. Every propositional variable p_i is an (atomic) formula.
4. If φ is a formula, then $\neg\varphi$ is a formula.
5. If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.
6. If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.
7. If φ and ψ are formulas, then $(\varphi \to \psi)$ is a formula.
8. If φ and ψ are formulas, then $(\varphi \leftrightarrow \psi)$ is a formula.
9. If φ is a formula, then $\mathsf{P}\varphi$, $\mathsf{H}\varphi$, $F\varphi$, $\mathsf{G}\varphi$ are all formulas.
10. Nothing else is a formula.

The semantics of temporal logics are given in terms of relational models, as with other kinds of intensional logics.

Definition 56.3. A *model* for temporal language is a triple $\mathfrak{M} = \langle T, \prec, V\rangle$, where

1. T is a nonempty set, interpreted as points in time.
2. $\prec$ is a binary relation on T.
3. V is a function assigning to each propositional variable p a set $V(p)$ of points in time.

When $t \prec t'$ holds, we say that t *precedes* t'. When $t \in V(p)$ we say p is *true at* t.

For now, you will notice that we do not impose any conditions on our precedence relation $\prec$. This means that at present, there are no restrictions on the structure of our temporal models, so we could have models in which time is linear, branching, circular, or has any structure whatsoever.

Just as with normal modal logic, every temporal model determines which formulas count as true at which points in it. We use the same notation "model $\mathfrak{M}$ makes formula φ true at point t" for the basic notion of relational semantics. The relation is defined inductively and is identical to the normal modal case for all non-modal operators.

aml:tl:sem:defn:tmodels

Definition 56.4. *Truth of a formula φ at t* in a $\mathfrak{M}$, in symbols: $\mathfrak{M}, t \Vdash \varphi$, is defined inductively as follows:

1. $\varphi \equiv \bot$: Never $\mathfrak{M}, t \Vdash \bot$.
2. $\varphi \equiv \top$: Always $\mathfrak{M}, t \Vdash \top$.
3. $\mathfrak{M}, t \Vdash p$ iff $t \in V(p)$
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t \nVdash \psi$.

5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t \Vdash \psi$ and $\mathfrak{M}, t \Vdash \chi$.

6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t \Vdash \psi$ or $\mathfrak{M}, t \Vdash \chi$ (or both).

7. $\varphi \equiv (\psi \to \chi)$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t \nVdash \psi$ or $\mathfrak{M}, t \Vdash \chi$.

8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, t \Vdash \varphi$ iff either both $\mathfrak{M}, t \Vdash \psi$ and $\mathfrak{M}, t \Vdash \chi$ or neither $\mathfrak{M}, t \Vdash \psi$ nor $\mathfrak{M}, t \Vdash \chi$.

9. $\varphi \equiv \mathsf{P}\psi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for some $t' \in T$ with $t' \prec t$ aml:tl:sem:defn:sub:mmodels-p

10. $\varphi \equiv \mathsf{H}\psi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for every $t' \in T$ with $t' \prec t$ aml:tl:sem:defn:sub:mmodels-h

11. $\varphi \equiv \mathsf{F}\psi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for some $t' \in T$ with $t \prec t'$ aml:tl:sem:defn:sub:mmodels-f

12. $\varphi \equiv \mathsf{G}\psi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for every $t' \in T$ with $t \prec t'$ aml:tl:sem:defn:sub:mmodels-g

Based on the semantics, you might be able to see that the operators P and H are duals, as well as the operators F and G, such that we could define $\mathsf{H}\varphi$ as $\neg\mathsf{P}\neg\varphi$, and the same with G and F.

---

content/applied-modal-logic/temporal-logic/properties-accessibility.tex

56.3 Properties of Temporal Frames

Given that our temporal models do not impose any conditions on the relation $\prec$, the only one of our familiar axioms that holds in all models is K, or its analogues K_{G} and K_{H}: aml:tl:acc:sec

$$\mathsf{G}(p \to q) \to (\mathsf{G}p \to \mathsf{G}q) \qquad (K_{\mathsf{G}})$$
$$\mathsf{H}(p \to q) \to (\mathsf{H}p \to \mathsf{H}q) \qquad (K_{\mathsf{H}})$$

However, if we want our models to impose stricter conditions on how time is represented, for instance by ensuring that $\prec$ is a linear order, then we will end up with other validities in our models.

Several of the properties from Table 56.1 might seem like desirable features for a model that is intended to represent time. However, it is worth noting that, even though we can impose whichever conditions we like on the $\prec$ relation, not all conditions correspond to formulas that can be expressed in the language of temporal logic. For example, irreflexivity, or the idea that $\forall w \neg(w \prec w)$, does not have a corresponding formula in temporal logic.

---

content/applied-modal-logic/temporal-logic/extra-temporal-operators.tex

If $\prec$ is ...	*then ... is true in $\mathfrak{M}$:*
transitive: $\forall u \forall v \forall w((u \prec v \wedge v \prec w) \to u \prec w)$	$\mathsf{F}\mathsf{F}p \to \mathsf{F}p$
linear: $\forall w \forall v(w \prec v \vee w = v \vee v \prec w)$	$(\mathsf{F}\mathsf{P}p \vee \mathsf{P}\mathsf{F}p) \to (\mathsf{P}p \vee p \vee \mathsf{F}p)$
dense: $\forall w \forall v(w \prec v \to \exists u(w \prec u \wedge u \prec v))$	$\mathsf{F}p \to \mathsf{F}\mathsf{F}p$
unbounded (past): $\forall w \exists v(v \prec w)$	$\mathsf{H}p \to \mathsf{P}p$
unbounded (future): $\forall w \exists v(w \prec v)$	$\mathsf{G}p \to \mathsf{F}p$

Table 56.1: Some temporal frame correspondence properties.

aml:tl:acc:
tab:correspondence

56.4 Additional Operators for Temporal Logic

aml:tl:ext:
sec

In addition to the unary operators for past and future, temporal logics also sometimes include binary operators S and U, intended to symbolize "since" and "until". This means adding S and U into the language of temporal logic and adding the following clause into the definition of a temporal formula:

> If φ and ψ are formulas, then $(\mathsf{S}\varphi\psi)$ and $(\mathsf{U}\varphi\psi)$ are both formulas.

The semantics for these operators are then given as follows:

aml:tl:ext:
defn:since-until

Definition 56.5. *Truth of a formula φ at t* in a $\mathfrak{M}$:

aml:tl:ext:
defn:sub:mmodels-since

1. $\varphi \equiv \mathsf{S}\psi\chi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for some $t' \in T$ with $t' \prec t$, and for all s with $t' \prec s \prec t$, $\mathfrak{M}, s \Vdash \chi$

aml:tl:ext:
defn:sub:mmodels-until

2. $\varphi \equiv \mathsf{U}\psi\chi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for some $t' \in T$ with $t \prec t'$, and for all s with $t \prec s \prec t'$, $\mathfrak{M}, s \Vdash \chi$

The intuitive reading of $\mathsf{S}\psi\chi$ is "Since ψ was the case, χ has been the case." And the intuitive reading of $\mathsf{U}\psi\chi$ is "Until ψ will be the case, χ will be the case."

---

content/applied-modal-logic/temporal-logic/possible-histories.tex

56.5 Possible Histories

aml:tl:poss:
sec

The relational models of temporal logic that we have been using are extremely flexible, since we do not have to place any restrictions on the accessibility relation. This means that temporal models can branch in the past and in the future, but we might want to consider a more "modal" conception of branching, in which we consider sequences of events as possible histories. This does not

necessarily require changing our language, though we might also add our "ordinary" modal operators $\Box$ and $\Diamond$, and we could also consider adding epistemic accessibility relations to represent changes in agents' knowledge over time.

Definition 56.6. A *possible histories model* for the temporal language is a triple $\mathfrak{M} = \langle T, C, V \rangle$, where

1. T is a nonempty set, interpreted as states in time.

2. C is a set of computational paths, or *possible histories* of a system. In other words, C is a set of sequences σ of states s_1, s_2, s_3, ..., where every $s_i \in T$.

3. V is a function assigning to each propositional variable p a set $V(p)$ of points in time.

To make things simpler, we will also generally assume that when a history is in C, then so are all of its suffixes. For example, if s_1, s_2, s_3 is a sequence in C, then so are s_2, s_3 and s_3. Also, when two states s_i and s_j appear in a sequence σ, we say that $s_i \prec_\sigma s_j$ when $i < j$. When $t \in V(p)$ we say p is *true at* t.

The one relevant change is that when we evaluate the truth of a formula at a point in time t in a model $\mathfrak{M}$, we do so relative to a history σ, in which t appears as a state. We do not need to change any of the semantics for propositional variables or for truth-functional connectives, though. All of those are exactly as they were in Definition 56.4, since none of those will make reference to σ. However, we now redefine our future operator F and add our $\Diamond$ operator with respect to these histories.

Definition 56.7. *Truth of a formula φ at t, σ in $\mathfrak{M}$*, in symbols: $\mathfrak{M}, t, \sigma \Vdash \varphi$: aml:tl:poss:defn:phmodels

1. $\varphi \equiv \mathsf{F}\psi$: $\mathfrak{M}, t, \sigma \Vdash \varphi$ iff $\mathfrak{M}, t', \sigma \Vdash \psi$ for some $t' \in T$ such that $t \prec_\sigma t'$. aml:tl:poss:defn:sub:phmodels-f

2. $\varphi \equiv \Diamond\psi$: $\mathfrak{M}, t, \sigma \Vdash \varphi$ iff $\mathfrak{M}, t, \sigma' \Vdash \psi$ for some $\sigma' \in C$ in which t occurs. aml:tl:poss:defn:sub:phmodels-diamond

Other temporal and modal operators can be defined similarly. However, we can now represent claims that combine tense and modality. For example, we might symbolize "p will not occur, but it might have occurred" using the formula $\neg\mathsf{F}p \wedge \Diamond\mathsf{F}p$. This would hold at a point and a history at which p does

not become true at a successor state, but there is an alternative history at which p will become true.

Chapter 57

Epistemic Logics

This chapter covers the metatheory of epistemic logics. It is structured in a similar way to Aldo Antonelli's notes on classical basic modal logic, but has been rewritten by Audrey Yap in order to add material on bisimulation and dynamic epistemic logics.

---

content/applied-modal-logic/epistemic-logic/introduction.tex

57.1 Introduction

aml:el:int:sec

Just as modal logic deals with *modal propositions* and the entailment relations among them, epistemic logic deals with *epistemic propositions* and the entailment relations among them. Rather than interpreting the modal operators as representing possibility and necessity, the unary connectives are interpreted in epistemic or doxastic ways, to model knowledge and belief. For example, we might want to express claims like the following:

1. Richard knows that Calgary is in Alberta.
2. Audrey thinks it is possible that a dog is on the couch.
3. Richard knows that Audrey knows that her class is on Tuesdays.
4. Everyone knows that a year has 12 months.

Contemporary epistemic logic is often traced to Jaako Hintikka's *Knowledge and Belief*, from 1962, and it was written at a time when possible worlds semantics were becoming increasingly more used in logic. In fact, epistemic logics use most of the same semantic tools as other modal logics, but will interpret them differently. The main change is in what we take the *accessibility*

relation to represent. In epistemic logics, they represent some form of *epistemic possibility*. We'll see that the epistemic notion that we're modelling will affect the constraints that we want to place on the accessibility relation. And we'll also see what happens to correspondence theory when it is given an epistemic interpretation. You'll notice that the examples above mention two agents: Richard and Audrey, and the relationship between the things that each one knows. The epistemic logics we'll consider will be multi-agent logics, in which such things can be expressed. In contrast, a single-agent epistemic logic would only talk about what one individual knows or believes.

---

content/applied-modal-logic/epistemic-logic/language-epistemic-logic.tex

57.2 The Language of Epistemic Logic

aml:el:lan:sec

Definition 57.1. Let G be a set of agent-symbols. The basic language of multi-agent epistemic logic contains

1. The propositional constant for falsity $\bot$.
2. The propositional constant for truth $\top$.
3. A denumerable set of propositional variables: p_0, p_1, p_2, ...
4. The propositional connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (conditional), $\leftrightarrow$ (biconditional).
5. The knowledge operator K_a where $a \in G$.

If we are only concerned with the knowledge of a single agent in our system, we can drop the reference to the set G, and individual agents. In that case, we only have the basic operator K.

Definition 57.2. *Formulas* of the epistemic language are inductively defined as follows:

1. $\bot$ is an atomic formula.
2. $\top$ is an atomic formula.
3. Every propositional variable p_i is an (atomic) formula.
4. If φ is a formula, then $\neg\varphi$ is a formula.
5. If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.
6. If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.
7. If φ and ψ are formulas, then $(\varphi \rightarrow \psi)$ is a formula.
8. If φ and ψ are formulas, then $(\varphi \leftrightarrow \psi)$ is a formula.

9. If φ is a formula and $a \in G$, then $\mathsf{K}_a\varphi$ is a formula.

10. Nothing else is a formula.

If a formula φ does not contain K_a, we say it is *modal-free*.

Definition 57.3. While the K operator is intended to symbolize individual knowledge, E, often read as "everybody knows," symbolizes group knowledge. Where $G' \subseteq G$, we define $\mathsf{E}_{G'}\varphi$ as an abbreviation for

$$\bigwedge_{b \in G'} \mathsf{K}_b\varphi.$$

We can also define an even stronger sense of knowledge, namely *common knowledge* among a group of agents G. When a piece of information is common knowledge among a group of agents, it means that for every combination of agents in that group, they all know that each other knows that each other knows ... ad infinitum. This is significantly stronger than group knowledge, and it is easy to come up with relational models in which a formula is group knowledge, but not common knowledge. We will use $\mathsf{C}_G\varphi$ to symbolize "it is common knowledge among G that φ."

---

content/applied-modal-logic/epistemic-logic/relational-models.tex

57.3 Relational Models

aml:el:rel:sec

The basic semantic concept for epistemic logics is the same as that of ordinary modal logics. Relational models still consist of a set of worlds, and an assignment that determines which propositional variables count as "true" at which worlds. And if we are only dealing with a single agent, we have a single accessibility relation as usual. However, if we have a multi-agent epistemic logic, then our single accessibility relation becomes a set of accessibility relations, one for each a in our set of agent symbols G.

A *relational model* consists of a set of worlds, which are related by binary accessibility relations—one for each agent—together with an assignment which determines which propositional variables are true at which worlds.

Definition 57.4. A *model* for the multi-agent epistemic language is a triple $\mathfrak{M} = \langle W, R, V\rangle$, where

1. W is a nonempty set of "worlds,"

2. For each $a \in G$, R_a is a binary accessibility relation on W, and

3. V is a function assigning to each propositional variable p a set $V(p)$ of possible worlds.

When R_aww' holds, we say that w' is *accessible by a from w*. When $w \in V(p)$ we say p is *true at w*.

The mechanics are just like the mechanics for normal modal logic, just with more accessibility relations added in. For a given agent, we will generally interpret their accessibility relation as representing something about their informational states. For example, we often treat $R_a ww'$, as expressing that w' is consistent with a's information at w. Or to put it another way, at w, they cannot tell the difference between world w and world w'.

---

content/applied-modal-logic/epistemic-logic/truth-at-w.tex

57.4 Truth at a World

Just as with normal modal logic, every epistemic model determines which formulas count as true at which worlds in it. We use the same notation "model $\mathfrak{M}$ makes formula φ true at world w" for the basic notion of relational semantics. The relation is defined inductively and is identical to the normal modal case for all non-modal operators. aml:el:trw:sec

Definition 57.5. *Truth of a formula φ at w* in a $\mathfrak{M}$, in symbols: $\mathfrak{M}, w \Vdash \varphi$, is defined inductively as follows: aml:el:trw:defn:mmodels

1. $\varphi \equiv \bot$: Never $\mathfrak{M}, w \Vdash \bot$.
2. $\varphi \equiv \top$: Always $\mathfrak{M}, w \Vdash \top$.
3. $\mathfrak{M}, w \Vdash p$ iff $w \in V(p)$
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$ (or both).
7. $\varphi \equiv (\psi \to \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$ or $\mathfrak{M}, w \Vdash \chi$.
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff either both $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$ or neither $\mathfrak{M}, w \Vdash \psi$ nor $\mathfrak{M}, w \Vdash \chi$.
9. $\varphi \equiv \mathsf{K}_a\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w' \Vdash \psi$ for all $w' \in W$ with $R_a ww'$ aml:el:trw:defn:sub:mmodels-box

Here's where we need to think about restrictions on our accessibility relations, though. After all, by clause (9), a formula $\mathsf{K}_a\psi$ is true at w whenever there are no w' with $R_a ww'$. This is the same clause as in normal modal logic; when a world has no successors, all $\Box$-formulas are vacuously true there. This seems extremely counterintuitive if we think about K as representing *knowledge*. After all, we tend to think that there are *no* circumstances under which an agent might know both φ and $\neg\varphi$ at the same time.

One solution is to ensure that our accessibility relation in epistemic logic will always be *reflexive*. This roughly corresponds to the idea that the actual world is consistent with an agent's information. In fact, epistemic logics typically use

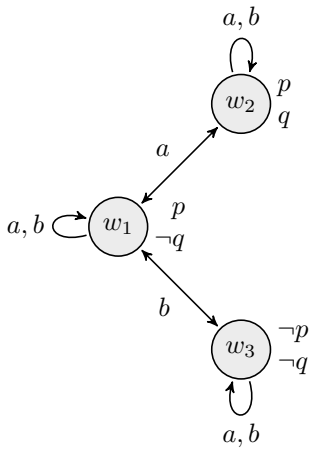


Figure 57.1: A simple epistemic model.

aml:el:trw:
fig:simple

S5, but others might use weaker systems depending on what exactly they want the K_a relation to represent.

Problem 57.1. Consider which of the following hold in Figure 57.1:

1. $\mathfrak{M}, w_1 \Vdash \neg q$;
2. $\mathfrak{M}, w_1 \Vdash \mathsf{K}_a \neg q$;
3. $\mathfrak{M}, w_1 \Vdash \mathsf{K}_b \neg q$;
4. $\mathfrak{M}, w_2 \Vdash \mathsf{K}_b q \vee \mathsf{K}_b \neg q$;
5. $\mathfrak{M}, w_2 \Vdash \mathsf{K}_a(\mathsf{K}_b q \vee \mathsf{K}_b \neg q)$;
6. $\mathfrak{M}, w_3 \Vdash \mathsf{E}_{\{a,b\}} \neg q$;

Now that we have given our basic definition of truth at a world, the other semantic concepts from normal modal logic, such as modal validity and entailment, simply carry over, applied to this new way of thinking about the interpretation for the modal operators.

We are now also in a position to give truth conditions for the common knowledge operator C_G. Recall from section 2.8 that the *transitive closure* R^+ of a relation R is defined as

$$R^+ = \bigcup_{n \in \mathbb{N}} R^n,$$

If R is ...	*then ... is true in $\mathfrak{M}$:*
	$\mathsf{K}(p \to q) \to (\mathsf{K}p \to \mathsf{K}q)$ (Closure)
reflexive: $\forall w Rww$	$\mathsf{K}p \to p$ (Veridicality)
transitive: $\forall u \forall v \forall w((Ruv \wedge Rvw) \to Ruw)$	$\mathsf{K}p \to \mathsf{K}\mathsf{K}p$ (Positive Introspection)
euclidean: $\forall w \forall u \forall v((Rwu \wedge Rwv) \to Ruv)$	$\neg\mathsf{K}p \to \mathsf{K}\neg\mathsf{K}p$ (Negative Introspection)

Table 57.1: Four epistemic principles.

aml:el:acc:tab:four

where

$$R^0 = R \text{ and}$$
$$R^{n+1} = \{\langle x, z\rangle : \exists y(R^n xy \wedge Ryz)\}.$$

Then, where G is a group of agents, we define $R_G = (\bigcup_{b\in G} R_b)^+$ to be the transitive closure of the union of all agents' accessibility relations.

Definition 57.6. If $G' \subseteq G$, we let $\mathfrak{M}, w \Vdash \mathsf{C}_{G'}\varphi$ iff for every w' such that $R_{G'}ww'$, $\mathfrak{M}, w' \Vdash \varphi$.

---

content/applied-modal-logic/epistemic-logic/properties-accessibility.tex

57.5 Accessibility Relations and Epistemic Principles

aml:el:acc:sec

Given what we already know about frame correspondence in normal modal logics, we might want to see what the characteristic formulas look like given epistemic interpretations. We have already said that epistemic logics are typically interpreted in S5. So let's take a look at how various epistemic principles are represented, and consider how they correspond to various frame conditions.

Recall from normal modal logic, that different modal formulas characterized different properties of accessibility relations. This table picks out a few that correspond to particular epistemic principles.

Veridicality, corresponding to the T axiom, is often treated as the most uncontroversial of these principles, as it represents that claim that if a formula is known, then it must be true. Closure, as well as Positive and Negative Introspection are much more contested.

Closure, corresponding to the K axiom, represents the idea that an agent's knowledge is closed under implication. This might seem plausible to us in some cases. For instance, I might know that if I am in Victoria, then I am on Vancouver Island. Barring odd skeptical scenarios, I do know that I am in Victoria, and this should also suggest that I know I am on Vancouver Island. So

in this case, the logical closure of my knowledge might seem relatively intuitive. On the other hand, we do not always think through the consequences of our knowledge, and so this might lead to less intuitive results in other cases.

Positive Introspection, sometimes known as the KK-principle, is sometimes articulated as the statement that if I know something, then I know that I know. It is the epistemic counterpart of the 4 axiom. Correspondingly, negative introspection is articulated as the statement that if I *don't* know something, then I know that I don't know it, which is the counterpart of the 5 axiom. Both of these seem to admit of relatively ordinary counterexamples, in which I am unsure whether or not I know something that I do in fact know.

---

content/applied-modal-logic/epistemic-logic/bisimulations.tex

57.6 Bisimulations

aml:el:bsd:sec

One remaining question that we might have about the expressive power of our epistemic language has to do with the relationship between models and the formulas that hold in them. We have seen from our frame correspondence results that when certain formulas are valid in a frame, they will also ensure that those frames satisfy certain properties. But does our modal language, for example, allow us to distinguish between a world at which there is a reflexive arrow, and an infinite chain of worlds, each of which leads to the next? That is, is there any formula A that might hold at only one of these two worlds?

Bisimulation is a relationship that we can define between relational models to say that they have effectively the same structure. And as we will see, it will capture a sense of equivalence between models that can be captured in our epistemic language.

Definition 57.7 (Bisimulation). Let $M_1 = \langle W_1, R_1, V_1\rangle$ and $M_2 = \langle W_2, R_2, V_2\rangle$ be two relational models. And let $\mathcal{R} \subseteq W_1 \times W_2$ be a binary relation. We say that $\mathcal{R}$ is a *bisimulation* when for every $\langle w_1, w_2\rangle \in \mathcal{R}$, we have:

1. $w_1 \in V_1(p)$ iff $w_2 \in V_2(p)$ for all propositional variables p.
2. For all agents $a \in A$ and worlds $v_1 \in W_1$, if $R_{1_a} w_1 v_1$ then there is some $v_2 \in W_2$ such that $R_{2_a} w_2 v_2$, and $\langle v_1, v_2\rangle \in \mathcal{R}$.
3. For all agents $a \in A$ and worlds $v_2 \in W_2$, if $R_{2_a} w_2 v_2$ then there is some $v_1 \in W_1$ such that $R_{1_a} w_1 v_1$, and $\langle v_1, v_2\rangle \in \mathcal{R}$.

When there is a bisimulation between M_1 and M_2 that links worlds w_1 and w_2, we can also write $\langle M_1, w_1\rangle \leftrightarrows \langle M_2, w_2\rangle$, and call $\langle M_1, w_1\rangle$ and $\langle M_2, w_2\rangle$ *bisimilar*.

The different clauses in the bisimulation relation ensure different things. Clause 1 ensures that bisimilar worlds will satisfy the same modal-free formulas, since it ensures agreement on all propositional variables. The other two clauses,

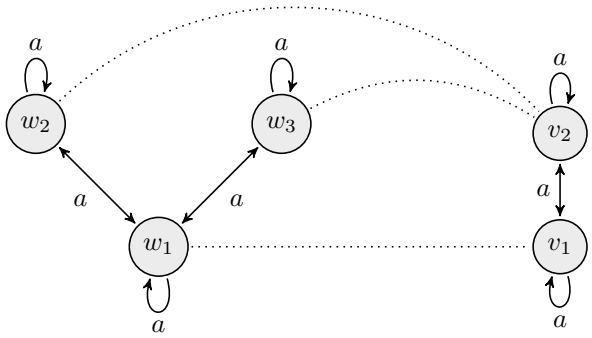


Figure 57.2: Two bisimilar models.

aml:el:bsd:fig:bisimilar

sometimes referred to as "forth" and "back," respectively, ensure that the accessibility relations will have the same structure.

Theorem 57.8. *If* $\langle M_1, w_1\rangle \leftrightarrows \langle M_2, w_2\rangle$*, then for every formula* φ*, we have that* $\mathfrak{M}_1, w_1 \Vdash \varphi$ *iff* $\mathfrak{M}_2, w_2 \Vdash \varphi$*.*

Even though the two models pictured in Figure 57.2 aren't quite the same as each other, there is a bisimulation linking worlds w_1 and v_1. This bisimulation will also link both w_2 and w_3 to v_2, with the idea being that there is nothing expressible in our modal language that can really distinguish between them. The situation would be different if w_2 and w_3 satisfied different propositional variables, however.

---

content/applied-modal-logic/epistemic-logic/public-announcement-logic-lang.tex

57.7 Public Announcement Logic

aml:el:pal:sec

Dynamic epistemic logics allow us to represent the ways in which agents' knowledge changes over time, or as they gain new information. Many of these represent changes in knowlege using informational *events* or *updates*. The most basic kind of update is a public announcement in which some formula is truthfully announced and all of the agents witness this taking place together. To do this, we expand the language as follows

Definition 57.9. Let G be a set of agent-symbols. The basic language of multi-agent epistemic logic with public announcements contains

1. The propositional constant for falsity $\bot$.
2. The propositional constant for truth $\top$.

3. A denumerable set of propositional variables: p_0, p_1, p_2, ...

4. The propositional connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (conditional)

5. The knowledge operator K_a where $a \in G$.

6. The public announcement operator $[\psi]$ where ψ is a formula.

The public announcement operator functions as a box operator, and our inductive definition of the language is given accordingly:

Definition 57.10. *Formulas* of the epistemic language are inductively defined as follows:

1. $\bot$ is an atomic formula.

2. $\top$ is an atomic formula.

3. Every propositional variable p_i is an (atomic) formula.

4. If φ is a formula, then $\neg\varphi$ is a formula.

5. If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.

6. If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.

7. If φ and ψ are formulas, then $(\varphi \rightarrow \psi)$ is a formula.

8. If φ and ψ are formulas, then $(\varphi \leftrightarrow \psi)$ is a formula.

9. If φ is a formula and $a \in G$, then $\mathsf{K}_a\varphi$ is a formula.

10. If φ and ψ are formulas, then $[\varphi]\psi$ is a formula.

11. Nothing else is a formula.

The intended reading of the formula $[\varphi]\psi$ is "After φ is truthfully announced, ψ holds. It will sometimes also be useful to talk about common knowledge in the context of public announcements, so the language may also include the common knowledge operator $\mathsf{C}_G\varphi$.

---

content/applied-modal-logic/epistemic-logic/public-announcement-logic-semantics.tex

57.8 Semantics of Public Announcement Logic

Relational models for public announcement logics are the same as they were in epistemic logics. However, the semantics for the public announcement operator are something new.

aml:el:psm:sec

Definition 57.11. *Truth of a formula φ at w* in a $\mathfrak{M} = \langle W, R, V\rangle$, in symbols: $\mathfrak{M}, w \Vdash \varphi$, is defined inductively as follows:

aml:el:psm:defn:mmodels

1. $\varphi \equiv \bot$: Never $\mathfrak{M}, w \Vdash \bot$.
2. $\varphi \equiv \top$: Always $\mathfrak{M}, w \Vdash \top$.
3. $\mathfrak{M}, w \Vdash p$ iff $w \in V(p)$
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$ (or both).
7. $\varphi \equiv (\psi \to \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$ or $\mathfrak{M}, w \Vdash \chi$.
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff either both $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$ or neither $\mathfrak{M}, w \Vdash \psi$ nor $\mathfrak{M}, w \Vdash \chi$.
9. $\varphi \equiv \mathsf{K}_a\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w' \Vdash \psi$ for all $w' \in W$ with $R_a w w'$ aml:el:psm:defn:sub:mmodels-box
10. $\varphi \equiv [\psi]\chi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ implies $\mathfrak{M} \mid \psi, w \Vdash \chi$ aml:el:psm:defn:sub:mmodels-pal

    Where $\mathfrak{M} \mid \psi = \langle W', R', V'\rangle$ is defined as follows:

    a) $W' = \{u \in W : \mathfrak{M}, u \Vdash \psi\}$. So the worlds of $\mathfrak{M} \mid \psi$ are the worlds in $\mathfrak{M}$ at which ψ holds.

    b) $R'_a = R_a \cap (W' \times W')$. Each agent's accessibility relation is simply restricted to the worlds that remain in W'.

    c) $V'(p) = \{u \in W' : u \in V(p)\}$. Similarly, the propositional valuations at worlds remain the same, representing the idea that informational events will not change the truth value of propositional variables.

What is distinctive, then, about public announcement logics, is that the truth of a formula at $\mathfrak{M}$ can sometimes only be decided by referring to a model other than $\mathfrak{M}$ itself.

Notice also that our semantics treats the announcement operator as a $\Box$ operator, and so if a formula φ cannot be truthfully announced at a world, then $[\varphi]B$ will hold there trivially, just as all $\Box$ formulas hold at endpoints.

We can see the public announcement of a formula as shrinking a model, or restricting it to the worlds at which the formula was true. Figure 57.3 gives an example of the effects of publicly announcing p. One notable thing about

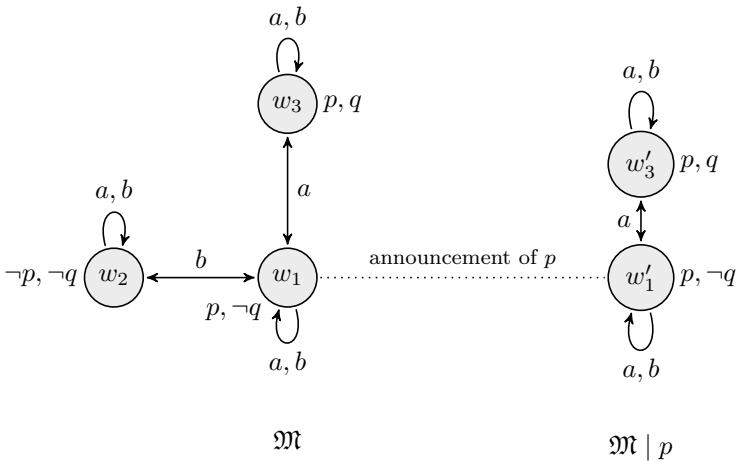


Figure 57.3: Before and after the public announcement of p.

aml:el:psm:
fig:announcement-example

that model is that agent b learns that p as a result of the announcement, while agent a does not (since a already knew that p was true).

More formally, we have $\mathfrak{M}, w_1 \Vdash \neg \mathsf{K}_b p$ but $\mathfrak{M} \mid p, w_1' \Vdash \mathsf{K}_b p$. This implies that $\mathfrak{M}, w_1 \Vdash [p]\mathsf{K}_b p$. But we have some even stronger claims that we can make about the result of the announcement. In fact, it is the case that $\mathfrak{M}, w_1 \Vdash [p]\mathsf{C}_{\{a,b\}} p$. In other words, after p is announced, it becomes *common knowledge*.

We might wonder, though, whether this holds in the general case, and whether a truthful announcement of φ will *always* result in φ becoming common knowledge. It may be surprising that the answer is no. And in fact, it is possible to truthfully announce formulas that will no longer be true once they are announced. For example, consider the effects of announcing $p \wedge \neg \mathsf{K}_b p$ at w_1 in Figure 57.3. In fact, $\mathfrak{M} \mid p$ and $\mathfrak{M} \mid (p \wedge \neg \mathsf{K}_b p)$ are the same model. However, as we have already noted, $\mathfrak{M} \mid p, w_1' \Vdash \mathsf{K}_b p$. Therefore, $\mathfrak{M} \mid (p \wedge \neg \mathsf{K}_b p), w_1' \Vdash \neg(p \wedge \neg \mathsf{K}_b p)$, so this is a formula that becomes false once it has been announced.

Part XIII

Intuitionistic Logic

Chapter 58

Introduction

---

content/intuitionistic-logic/introduction/constructive-reasoning.tex

58.1 Constructive Reasoning

In contrast to extensions of classical logic by modal operators or second-order quantifiers, intuitionistic logic is "non-classical" in that it restricts classical logic. Classical logic is *non-constructive* in various ways. Intuitionistic logic is intended to capture a more "constructive" kind of reasoning characteristic of a kind of constructive mathematics. The following examples may serve to illustrate some of the underlying motivations.

Suppose someone claimed that they had determined a natural number n with the property that if n is even, the Riemann hypothesis is true, and if n is odd, the Riemann hypothesis is false. Great news! Whether the Riemann hypothesis is true or not is one of the big open questions of mathematics, and they seem to have reduced the problem to one of calculation, that is, to the determination of whether a specific number is even or not.

What is the magic value of n? They describe it as follows: n is the natural number that is equal to 2 if the Riemann hypothesis is true, and 3 otherwise.

Angrily, you demand your money back. From a classical point of view, the description above does in fact determine a unique value of n; but what you really want is a value of n that is given *explicitly*.

To take another, perhaps less contrived example, consider the following question. We know that it is possible to raise an irrational number to a rational power, and get a rational result. For example, $\sqrt{2}^2 = 2$. What is less clear is whether or not it is possible to raise an irrational number to an *irrational*

power, and get a rational result. The following theorem answers this in the affirmative:

Theorem 58.1. *There are irrational numbers a and b such that a^b is rational.*

Proof. Consider $\sqrt{2}^{\sqrt{2}}$. If this is rational, we are done: we can let $a = b = \sqrt{2}$. Otherwise, it is irrational. Then we have

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\cdot\sqrt{2}} = \sqrt{2}^2 = 2,$$

which is rational. So, in this case, let a be $\sqrt{2}^{\sqrt{2}}$, and let b be $\sqrt{2}$. □

Does this constitute a valid proof? Most mathematicians feel that it does. But again, there is something a little bit unsatisfying here: we have proved the existence of a pair of real numbers with a certain property, without being able to say *which* pair of numbers it is. It is possible to prove the same result, but in such a way that the pair a, b *is* given in the proof: take $a = \sqrt{3}$ and $b = \log_3 4$. Then

$$a^b = \sqrt{3}^{\log_3 4} = 3^{1/2\cdot\log_3 4} = (3^{\log_3 4})^{1/2} = 4^{1/2} = 2,$$

since $3^{\log_3 x} = x$.

Intuitionistic logic is designed to capture a kind of reasoning where moves like the one in the first proof are disallowed. Proving the existence of an x satisfying $\varphi(x)$ means that you have to give a specific x, and a proof that it satisfies φ, like in the second proof. Proving that φ or ψ holds requires that you can prove one or the other.

Formally speaking, intuitionistic logic is what you get if you restrict a derivation system for classical logic in a certain way. From the mathematical point of view, these are just formal deductive systems, but, as already noted, they are intended to capture a kind of mathematical reasoning. One can take this to be the kind of reasoning that is justified on a certain philosophical view of mathematics (such as Brouwer's intuitionism); one can take it to be a kind of mathematical reasoning which is more "concrete" and satisfying (along the lines of Bishop's constructivism); and one can argue about whether or not the formal description captures the informal motivation. But whatever philosophical positions we may hold, we can study intuitionistic logic as a formally presented logic; and for whatever reasons, many mathematical logicians find it interesting to do so.

---

content/intuitionistic-logic/introduction/syntax.tex

58.2 Syntax of Intuitionistic Logic

int:int:syn:sec

The syntax of intuitionistic logic is the same as that for propositional logic. In classical propositional logic it is possible to define connectives by others, e.g.,

one can define $\varphi \to \psi$ by $\neg\varphi \vee \psi$, or $\varphi \vee \psi$ by $\neg(\neg\varphi \wedge \neg\psi)$. Thus, presentations of classical logic often introduce some connectives as abbreviations for these definitions. This is not so in intuitionistic logic, with two exceptions: $\neg\varphi$ can be—and often is—defined as an abbreviation for $\varphi \to \bot$. Then, of course, $\bot$ must not itself be defined! Also, $\varphi \leftrightarrow \psi$ can be defined, as in classical logic, as $(\varphi \to \psi) \wedge (\psi \to \varphi)$.

Formulas of propositional intuitionistic logic are built up from *propositional variables* and the propositional constant $\bot$ using *logical connectives*. We have:

1. A denumerable set At_0 of propositional variables p_0, p_1, ...
2. The propositional constant for falsity $\bot$.
3. The logical connectives: $\wedge$ (conjunction), $\vee$ (disjunction), $\to$ (conditional)
4. Punctuation marks: (,), and the comma.

Definition 58.2 (Formula). The set $\mathrm{Frm}(\mathcal{L}_0)$ of *formulas* of propositional intuitionistic logic is defined inductively as follows: int:int:syn:defn:formulas

1. $\bot$ is an atomic formula.
2. Every propositional variable p_i is an atomic formula.
3. If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.
4. If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.
5. If φ and ψ are formulas, then $(\varphi \to \psi)$ is a formula.
6. Nothing else is a formula.

In addition to the primitive connectives introduced above, we also use the following *defined* symbols: $\neg$ (negation) and $\leftrightarrow$ (biconditional). Formulas constructed using the defined operators are to be understood as follows:

1. $\neg\varphi$ abbreviates $\varphi \to \bot$.
2. $\varphi \leftrightarrow \psi$ abbreviates $(\varphi \to \psi) \wedge (\psi \to \varphi)$.

Although $\neg$ is officially treated as an abbreviation, we will sometimes give explicit rules and clauses in definitions for $\neg$ as if it were primitive. This is mostly so we can state practice problems.

---

content/intuitionistic-logic/introduction/bhk-interpretation.tex

58.3 The Brouwer–Heyting–Kolmogorov Interpretation

int:int:bhk:sec

Proofs of validity of intuitionistic propositions using the BHK interpretation are confusing; they have to be explained better.

There is an informal constructive interpretation of the intuitionist connectives, usually known as the Brouwer–Heyting–Kolmogorov interpretation. It uses the notion of a "construction," which you may think of as a constructive proof. (We don't use "proof" in the BHK interpretation so as not to get confused with the notion of a derivation in a formal derivation system.) Based on this intuitive notion, the BHK interpretation explains the meanings of the intuitionistic connectives.

1. We assume that we know what constitutes a construction of an atomic statement.
2. A construction of $\varphi_1 \wedge \varphi_2$ is a pair $\langle M_1, M_2 \rangle$ where M_1 is a construction of φ_1 and M_2 is a construction of φ_2.
3. A construction of $\varphi_1 \vee \varphi_2$ is a pair $\langle s, M \rangle$ where s is 1 and M is a construction of φ_1, or s is 2 and M is a construction of φ_2.
4. A construction of $\varphi \to \psi$ is a function that converts a construction of φ into a construction of ψ.
5. There is no construction for $\bot$ (absurdity).
6. $\neg\varphi$ is defined as synonym for $\varphi \to \bot$. That is, a construction of $\neg\varphi$ is a function converting a construction of φ into a construction of $\bot$.

Example 58.3. Take $\neg\bot$ for example. A construction of it is a function which, given any construction of $\bot$ as input, provides a construction of $\bot$ as output. Obviously, the identity function Id is such a construction: given a construction M of $\bot$, $\mathrm{Id}(M) = M$ yields a construction of $\bot$.

Generally speaking, $\neg\varphi$ means "A construction of φ is impossible".

Example 58.4. Let us prove $\varphi \to \neg\neg\varphi$ for any proposition φ, which is $\varphi \to ((\varphi \to \bot) \to \bot)$. The construction should be a function f that, given a construction M of φ, returns a construction $f(M)$ of $(\varphi \to \bot) \to \bot$. Here is how f constructs the construction of $(\varphi \to \bot) \to \bot$: We have to define a function g which, when given a construction h of $\varphi \to \bot$ as input, outputs a construction of $\bot$. We can define g as follows: apply the input h to the construction M of φ (that we received earlier). Since the output $h(M)$ of h is a construction of $\bot$, $f(M)(h) = h(M)$ is a construction of $\bot$ if M is a construction of φ.

Example 58.5. Let us give a construction for $\neg(\varphi\wedge\neg\varphi)$, i.e., $(\varphi\wedge(\varphi\to\bot))\to\bot$. This is a function f which, given as input a construction M of $\varphi\wedge(\varphi\to\bot)$, yields a construction of $\bot$. A construction of a conjunction $\psi_1\wedge\psi_2$ is a pair $\langle N_1, N_2\rangle$ where N_1 is a construction of ψ_1 and N_2 is a construction of ψ_2. We can define functions p_1 and p_2 which recover from a construction of $\psi_1\wedge\psi_2$ the constructions of ψ_1 and ψ_2, respectively:

$$p_1(\langle N_1, N_2\rangle) = N_1$$
$$p_2(\langle N_1, N_2\rangle) = N_2$$

Here is what f does: First it applies p_1 to its input M. That yields a construction of φ. Then it applies p_2 to M, yielding a construction of $\varphi\to\bot$. Such a construction, in turn, is a function $p_2(M)$ which, if given as input a construction of φ, yields a construction of $\bot$. In other words, if we apply $p_2(M)$ to $p_1(M)$, we get a construction of $\bot$. Thus, we can define $f(M) = p_2(M)(p_1(M))$.

Example 58.6. Let us give a construction of $((\varphi\wedge\psi)\to\chi)\to(\varphi\to(\psi\to\chi))$, i.e., a function f which turns a construction g of $(\varphi\wedge\psi)\to\chi$ into a construction of $(\varphi\to(\psi\to\chi))$. The construction g is itself a function (from constructions of $\varphi\wedge\psi$ to constructions of C). And the output $f(g)$ is a function h_g from constructions of φ to functions from constructions of ψ to constructions of χ.

Ok, this is confusing. We have to construct a certain function h_g, which will be the output of f for input g. The input of h_g is a construction M of φ. The output of $h_g(M)$ should be a function k_M from constructions N of ψ to constructions of χ. Let $k_{g,M}(N) = g(\langle M, N\rangle)$. Remember that $\langle M, N\rangle$ is a construction of $\varphi\wedge\psi$. So $k_{g,M}$ is a construction of $\psi\to\chi$: it maps constructions N of ψ to constructions of χ. Now let $h_g(M) = k_{g,M}$. That's a function that maps constructions M of φ to constructions $k_{g,M}$ of $\psi\to\chi$. Now let $f(g) = h_g$. That's a function that maps constructions g of $(\varphi\wedge\psi)\to\chi$ to constructions of $\varphi\to(\psi\to\chi)$. Whew!

The statement $\varphi\vee\neg\varphi$ is called the Law of Excluded Middle. We can prove it for some specific φ (e.g., $\bot\vee\neg\bot$), but not in general. This is because the intuitionistic disjunction requires a construction of one of the disjuncts, but there are statements which currently can neither be proved nor refuted (say, Goldbach's conjecture). However, you can't refute the law of excluded middle either: that is, $\neg\neg(\varphi\vee\neg\varphi)$ holds.

Example 58.7. To prove $\neg\neg(\varphi\vee\neg\varphi)$, we need a function f that transforms a construction of $\neg(\varphi\vee\neg\varphi)$, i.e., of $(\varphi\vee(\varphi\to\bot))\to\bot$, into a construction of $\bot$. In other words, we need a function f such that $f(g)$ is a construction of $\bot$ if g is a construction of $\neg(\varphi\vee\neg\varphi)$.

Suppose g is a construction of $\neg(\varphi\vee\neg\varphi)$, i.e., a function that transforms a construction of $\varphi\vee\neg\varphi$ into a construction of $\bot$. A construction of $\varphi\vee\neg\varphi$ is a pair $\langle s, M\rangle$ where either $s = 1$ and M is a construction of φ, or $s = 2$ and M is a construction of $\neg\varphi$. Let h_1 be the function mapping a construction M_1 of φ to a construction of $\varphi\vee\neg\varphi$: it maps M_1 to $\langle 1, M_2\rangle$. And let h_2 be the function

mapping a construction M_2 of $\neg\varphi$ to a construction of $\varphi \vee \neg\varphi$: it maps M_2 to $\langle 2, M_2\rangle$.

Let k be $g \circ h_1$: it is a function which, if given a construction of φ, returns a construction of $\bot$, i.e., it is a construction of $\varphi \to \bot$ or $\neg\varphi$. Now let l be $g \circ h_2$. It is a function which, given a construction of $\neg\varphi$, provides a construction of $\bot$. Since k is a construction of $\neg\varphi$, $l(k)$ is a construction of $\bot$.

Together, what we've done is describe how we can turn a construction g of $\neg(\varphi \vee \neg\varphi)$ into a construction of $\bot$, i.e., the function f mapping a construction g of $\neg(\varphi \vee \neg\varphi)$ to the construction $l(k)$ of $\bot$ is a construction of $\neg\neg(\varphi \vee \neg\varphi)$.

As you can see, using the BHK interpretation to show the intuitionistic validity of formulas quickly becomes cumbersome and confusing. Luckily, there are better derivation systems for intuitionistic logic, and more precise semantic interpretations.

---

content/intuitionistic-logic/introduction/natural-deduction.tex

58.4 Natural Deduction

int:int:ntd:sec

Natural deduction without the $\bot_C$ rules is a standard derivation system for intuitionistic logic. We repeat the rules here and indicate the motivation using the BHK interpretation. In each case, we can think of a rule which allows us to conclude that if the premises have constructions, so does the conclusion.

Since natural deduction derivations have undischarged assumptions, we should consider such a derivation, say, of φ from undischarged assumptions Γ, as a function that turns constructions of all $\psi \in \Gamma$ into a construction of φ. If there is a derivation of φ from no undischarged assumptions, then there is a construction of φ in the sense of the BHK interpretation. For the purpose of the discussion, however, we'll suppress the Γ when not needed.

An assumption φ by itself is a derivation of φ from the undischarged assumption φ. This agrees with the BHK-interpretation: the identity function on constructions turns any construction of φ into a construction of φ.

Conjunction

$$\frac{\varphi \qquad \psi}{\varphi \wedge \psi}\,\wedge\text{Intro} \qquad \frac{\varphi \wedge \psi}{\varphi}\,\wedge\text{Elim} \qquad \frac{\varphi \wedge \psi}{\psi}\,\wedge\text{Elim}$$

Suppose we have constructions N_1, N_2 of φ_1 and φ_2, respectively. Then we also have a construction $\varphi_1 \wedge \varphi_2$, namely the pair $\langle N_1, N_2\rangle$.

A construction of $\varphi_1 \wedge \varphi_1$ on the BHK interpretation is a pair $\langle N_1, N_2 \rangle$. So assume we have such a pair. Then we also have a construction of each conjunct: N_1 is a construction of φ_1 and N_2 is a construction of φ_2.

Conditional

$$\dfrac{\begin{array}{c}[\varphi]^u \\ \vdots \\ \psi\end{array}}{\varphi \to \psi}\, u \to\text{Intro} \qquad\qquad \dfrac{\varphi \to \psi \qquad \varphi}{\psi} \to\text{Elim}$$

If we have a derivation of ψ from undischarged assumption φ, then there is a function f that turns constructions of φ into constructions of ψ. That same function is a construction of $\varphi \to \psi$. So, if the premise of $\to$Intro has a construction conditional on a construction of φ, the conclusion $\varphi \to \psi$ has a construction.

On the other hand, suppose there are constructions N of φ and f of $\varphi \to \psi$. A construction of $\varphi \to \psi$ is a function that turns constructions of φ into constructions of ψ. So, $f(N)$ is a construction of ψ, i.e., the conclusion of $\to$Elim has a construction.

Disjunction

$$\dfrac{\varphi}{\varphi \vee \psi} \vee\text{Intro} \qquad \dfrac{\psi}{\varphi \vee \psi} \vee\text{Intro}$$

$$\dfrac{\varphi \vee \psi \qquad \begin{array}{c}[\varphi]^n \\ \vdots \\ \chi\end{array} \qquad \begin{array}{c}[\psi]^n \\ \vdots \\ \chi\end{array}}{\chi}\, n \vee\text{Elim}$$

If we have a construction N_i of φ_i we can turn it into a construction $\langle i, N_i \rangle$ of $\varphi_1 \vee \varphi_2$. On the other hand, suppose we have a construction of $\varphi_1 \vee \varphi_2$, i.e., a pair $\langle i, N_i \rangle$ where N_i is a construction of φ_i, and also functions f_1, f_2, which turn constructions of φ_1, φ_2, respectively, into constructions of χ. Then $f_i(N_i)$ is a construction of χ, the conclusion of $\vee$Elim.

Absurdity

$$\dfrac{\bot}{\varphi}\, \bot_I$$

If we have a derivation of $\bot$ from undischarged assumptions $\psi_1, \ldots, \psi_n$, then there is a function $f(M_1, \ldots, M_n)$ that turns constructions of $\psi_1, \ldots, \psi_n$ into a construction of $\bot$. Since $\bot$ has no construction, there cannot be any constructions of all of $\psi_1, \ldots, \psi_n$ either. Hence, f also has the property that *if* $M_1, \ldots, M_n$ are constructions of $\psi_1, \ldots, \psi_n$, respectively, *then* $f(M_1, \ldots, M_n)$ is a construction of φ.

Rules for $\neg$

Since $\neg\varphi$ is defined as $\varphi \to \bot$, we strictly speaking do not need rules for $\neg$. But if we did, this is what they'd look like:

$$\dfrac{\begin{array}{c}[\varphi]^n \\ \vdots \\ \bot\end{array}}{\neg\varphi}\ n\ \neg\text{Intro} \qquad \dfrac{\neg\varphi \qquad \varphi}{\bot}\ \neg\text{Elim}$$

Examples of Derivations

1. $\vdash \varphi \to (\neg\varphi \to \bot)$, i.e., $\vdash \varphi \to ((\varphi \to \bot) \to \bot)$

$$\dfrac{\dfrac{\dfrac{[\varphi]^2 \qquad [\varphi \to \bot]^1}{\bot}\to\text{Elim}}{(\varphi \to \bot) \to \bot}\ 1\ \to\text{Intro}}{\varphi \to (\varphi \to \bot) \to \bot}\ 2\ \to\text{Intro}$$

2. $\vdash ((\varphi \land \psi) \to \chi) \to (\varphi \to (\psi \to \chi))$

$$\dfrac{\dfrac{\dfrac{\dfrac{[(\varphi \land \psi) \to \chi]^3 \qquad \dfrac{[\varphi]^2 \qquad [\psi]^1}{\varphi \land \psi}\land\text{Intro}}{\chi}\to\text{Elim}}{\psi \to \chi}\ 1\ \to\text{Intro}}{\varphi \to (\psi \to \chi)}\ 2\ \to\text{Intro}}{((\varphi \land \psi) \to \chi) \to (\varphi \to (\psi \to \chi))}\ 3\ \to\text{Intro}$$

3. $\vdash \neg(\varphi \land \neg\varphi)$, i.e., $\vdash (\varphi \land (\varphi \to \bot)) \to \bot$

$$\dfrac{\dfrac{\dfrac{[\varphi \land (\varphi \to \bot)]^1}{\varphi \to \bot}\land\text{Elim} \qquad \dfrac{[\varphi \land (\varphi \to \bot)]^1}{\varphi}\land\text{Elim}}{\bot}\to\text{Elim}}{(\varphi \land (\varphi \to \bot)) \to \bot}\ 1\ \to\text{Intro}$$

4. $\vdash \neg\neg(\varphi \lor \neg\varphi)$, i.e., $\vdash ((\varphi \lor (\varphi \to \bot)) \to \bot) \to \bot$

$$\dfrac{\dfrac{\dfrac{[(\varphi \vee (\varphi \to \bot)) \to \bot]^2 \qquad \dfrac{[\varphi]^1}{\varphi \vee (\varphi \to \bot)} \vee\text{Intro}}{\dfrac{\bot}{\varphi \to \bot}\,1 \to\text{Intro}} \to\text{Elim} \qquad \dfrac{[(\varphi \vee (\varphi \to \bot)) \to \bot]^2}{\varphi \vee (\varphi \to \bot)} \vee\text{Intro}}{\bot} \to\text{Elim}}{((\varphi \vee (\varphi \to \bot)) \to \bot) \to \bot}\,2 \to\text{Intro}$$

Proposition 58.8. *If $\Gamma \vdash \varphi$ in natural deduction for intuitionistic logic, $\Gamma \vdash \varphi$ in classical logic. In particular, if φ is an intuitionistic theorem, it is also a classical theorem.*

Proof. Every natural deduction rule is also a rule in classical natural deduction, so every derivation in intuitionistic logic is also a derivation in classical logic. □

Problem 58.1. Give derivations in intuitionistic logic of the following formulas:

1. $(\neg\varphi \vee \psi) \to (\varphi \to \psi)$
2. $\neg\neg\neg\varphi \to \neg\varphi$
3. $\neg\neg(\varphi \wedge \psi) \leftrightarrow (\neg\neg\varphi \wedge \neg\neg\psi)$
4. $\neg(\varphi \vee \psi) \leftrightarrow (\neg\varphi \wedge \neg\psi)$
5. $(\neg\varphi \vee \neg\psi) \to \neg(\varphi \wedge \psi)$
6. $\neg\neg(\varphi \wedge \psi) \to (\neg\neg\varphi \vee \neg\neg\psi)$

---

content/intuitionistic-logic/introduction/axiomatic-derivations.tex

58.5 Axiomatic Derivations

int:int:axd:sec

Axiomatic derivations for intuitionistic propositional logic are the conceptually simplest, and historically first, derivation systems. They work just as in classical propositional logic.

Definition 58.9 (Derivability). If Γ is a set of formulas of $\mathcal{L}$ then a *derivation* from Γ is a finite sequence $\varphi_1, \ldots, \varphi_n$ of formulas where for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. φ_i follows from some φ_j and φ_k with $j < i$ and $k < i$ by modus ponens, i.e., $\varphi_k \equiv \varphi_j \to \varphi_i$.

Definition 58.10 (Axioms). The set of Ax_0 of *axioms* for the intuitionistic propositional logic are all formulas of the following forms:

int:int:axd:
$$(\varphi \wedge \psi) \to \varphi \tag{58.1}$$
ax:land1
int:int:axd:
$$(\varphi \wedge \psi) \to \psi \tag{58.2}$$
ax:land2
int:int:axd:
$$\varphi \to (\psi \to (\varphi \wedge \psi)) \tag{58.3}$$
ax:land3
int:int:axd:
$$\varphi \to (\varphi \vee \psi) \tag{58.4}$$
ax:lor1
int:int:axd:
$$\varphi \to (\psi \vee \varphi) \tag{58.5}$$
ax:lor2
int:int:axd:
$$(\varphi \to \chi) \to ((\psi \to \chi) \to ((\varphi \vee \psi) \to \chi)) \tag{58.6}$$
ax:lor3
int:int:axd:
$$\varphi \to (\psi \to \varphi) \tag{58.7}$$
ax:lif1
int:int:axd:
$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \tag{58.8}$$
ax:lif2
int:int:axd:
$$\bot \to \varphi \tag{58.9}$$
ax:lfalse1

Definition 58.11 (Derivability). A formula φ is *derivable* from Γ, written $\Gamma \vdash \varphi$, if there is a derivation from Γ ending in φ.

Definition 58.12 (Theorems). A formula φ is a *theorem* if there is a derivation of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Proposition 58.13. *If $\Gamma \vdash \varphi$ in the axiomatic derivation system for intuitionistic logic, $\Gamma \vdash \varphi$ also in classical logic. In particular, if φ is an intuitionistic theorem, it is also a classical theorem.*

Proof. Every intuitionistic axiom is also a classical axiom, so every derivation in intuitionistic logic is also a derivation in classical logic. □

Chapter 59

Semantics

This chapter collects definitions for semantics for intuitionistic logic. So far only Kripke and topological semantics are covered. There are no examples yet, either of how models make formulas true or of proofs that formulas are valid.

content/intuitionistic-logic/semantics/introduction.tex

59.1 Introduction

int:sem:int:sec

No logic is satisfactorily described without a semantics, and intuitionistic logic is no exception. Whereas for classical logic, the semantics based on valuations is canonical, there are several competing semantics for intuitionistic logic. None of them are completely satisfactory in the sense that they give an intuitionistically acceptable account of the meanings of the connectives.

The semantics based on relational models, similar to the semantics for modal logics, is perhaps the most popular one. In this semantics, propositional variables are assigned to worlds, and these worlds are related by an accessibility relation. That relation is always a partial order, i.e., it is reflexive, antisymmetric, and transitive.

Intuitively, you might think of these worlds as states of knowledge or "evidentiary situations." A state w' is accessible from w iff, for all we know, w' is a possible (future) state of knowledge, i.e., one that is compatible with what's known at w. Once a proposition is known, it can't become un-known, i.e., whenever φ is known at w and Rww', φ is known at w' as well. So "knowledge" is monotonic with respect to the accessibility relation.

If we define "φ is known" as in epistemic logic as "true in all epistemic alternatives," then $\varphi \wedge \psi$ is known at w if in all epistemic alternatives, both φ and ψ are known. But since knowledge is monotonic and R is reflexive, that means that $\varphi \wedge \psi$ is known at w iff φ and ψ are known at w. For the same reason, $\varphi \vee \psi$ is known at w iff at least one of them is known. So for $\wedge$ and $\vee$, the truth conditions of the connectives coincide with those in classical logic.

The truth conditions for the conditional, however, differ from classical logic. $\varphi \to \psi$ is known at w iff at no w' with Rww', φ is known without ψ also being known. This is not the same as the condition that φ is unknown or ψ is known at w. For if we know neither φ nor ψ at w, there might be a future epistemic state w' with Rww' such that at w', φ is known without also coming to know ψ.

We know $\neg\varphi$ only if there is no possible future epistemic state in which we know φ. Here the idea is that if φ were knowable, then in some possible future epistemic state φ becomes known. Since we can't know $\bot$, in that future epistemic state, we would know φ but not know $\bot$.

On this interpretation the principle of excluded middle fails. For there are some φ which we don't yet know, but which we might come to know. For such a formula φ, both φ and $\neg\varphi$ are unknown, so $\varphi \vee \neg\varphi$ is not known. But we do know, e.g., that $\neg(\varphi \wedge \neg\varphi)$. For no future state in which we know both φ and $\neg\varphi$ is possible, and we know this independently of whether or not we know φ or $\neg\varphi$.

Relational models are not the only available semantics for intuitionistic logic. The topological semantics is another: here propositions are interpreted

as open sets in a topological space, and the connectives are interpreted as operations on these sets (e.g., $\wedge$ corresponds to intersection).

---

content/intuitionistic-logic/semantics/relational-models.tex

59.2 Relational models

int:sem:rel: sec

In order to give a precise semantics for intuitionistic propositional logic, we have to give a definition of what counts as a model relative to which we can evaluate formulas. On the basis of such a definition it is then also possible to define semantics notions such as validity and entailment. One such semantics is given by relational models.

Definition 59.1. A relational model for intuitionistic propositional logic is a triple $\mathfrak{M} = \langle W, R, V \rangle$, where

1. W is a non-empty set,
2. R is a partial order (i.e., a reflexive, antisymmetric, and transitive binary relation) on W, and
3. V is a function assigning to each propositional variable p a subset of W, such that
4. V is monotone with respect to R, i.e., if $w \in V(p)$ and Rww', then $w' \in V(p)$.

int:sem:rel: defn:true-at-w

Definition 59.2. We define the notion of φ *being true at* w *in* $\mathfrak{M}$, $\mathfrak{M}, w \Vdash \varphi$, inductively as follows:

1. $\varphi \equiv p$: $\mathfrak{M}, w \Vdash \varphi$ iff $w \in V(p)$.
2. $\varphi \equiv \bot$: not $\mathfrak{M}, w \Vdash \varphi$.
3. $\varphi \equiv \neg\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff for no w' such that Rww', $\mathfrak{M}, w' \Vdash \psi$.
4. $\varphi \equiv \psi \wedge \chi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$.
5. $\varphi \equiv \psi \vee \chi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$ (or both).
6. $\varphi \equiv \psi \to \chi$: $\mathfrak{M}, w \Vdash \varphi$ iff for every w' such that Rww', not $\mathfrak{M}, w' \Vdash \psi$ or $\mathfrak{M}, w' \Vdash \chi$ (or both).

We write $\mathfrak{M}, w \nVdash \varphi$ if not $\mathfrak{M}, w \Vdash \varphi$. If Γ is a set of formulas, $\mathfrak{M}, w \Vdash \Gamma$ means $\mathfrak{M}, w \Vdash \psi$ for all $\psi \in \Gamma$.

Problem 59.1. Show that according to Definition 59.2, $\mathfrak{M}, w \Vdash \neg\varphi$ iff $\mathfrak{M}, w \Vdash \varphi \to \bot$.

int:sem:rel: prop:true-monotonic

Proposition 59.3. *Truth at worlds is monotonic with respect to R, i.e., if $\mathfrak{M}, w \Vdash \varphi$ and Rww', then $\mathfrak{M}, w' \Vdash \varphi$.*

Proof. Exercise. □

Problem 59.2. Prove Proposition 59.3.

---

content/intuitionistic-logic/semantics/semantic-notions.tex

59.3 Semantic Notions

int:sem:sem:sec

Definition 59.4. We say φ is *true in the model* $\mathfrak{M} = \langle W, R, V\rangle$, $\mathfrak{M} \Vdash \varphi$, iff $\mathfrak{M}, w \Vdash \varphi$ for all $w \in W$. φ is *valid*, $\vDash \varphi$, iff it is true in all models. We say a set of formulas Γ *entails* φ, $\Gamma \vDash \varphi$, iff for every model $\mathfrak{M}$ and every w such that $\mathfrak{M}, w \Vdash \Gamma$, $\mathfrak{M}, w \Vdash \varphi$.

Proposition 59.5.

int:sem:sem:prop:sat-entails

1. *If* $\mathfrak{M}, w \Vdash \Gamma$ *and* $\Gamma \vDash \varphi$, *then* $\mathfrak{M}, w \Vdash \varphi$. int:sem:sem:prop:sat-entails1
2. *If* $\mathfrak{M} \Vdash \Gamma$ *and* $\Gamma \vDash \varphi$, *then* $\mathfrak{M} \Vdash \varphi$. int:sem:sem:prop:sat-entails2

Proof. 1. Suppose $\mathfrak{M} \Vdash \Gamma$. Since $\Gamma \vDash \varphi$, we know that if $\mathfrak{M}, w \Vdash \Gamma$, then $\mathfrak{M}, w \Vdash \varphi$. Since $\mathfrak{M}, u \Vdash \Gamma$ for all every $u \in W$, $\mathfrak{M}, w \Vdash \Gamma$. Hence $\mathfrak{M}, w \Vdash \varphi$.

2. Follows immediately from (1). □

int:sem:sem:defn:restrict

Definition 59.6. Suppose $\mathfrak{M}$ is a relational model and $w \in W$. The *restriction* $\mathfrak{M}_w = \langle W_w, R_w, V_w\rangle$ of $\mathfrak{M}$ to w is given by:

$$\begin{aligned} W_w &= \{u \in W : Rwu\},\\ R_w &= R \cap (W_w)^2, \text{ and}\\ V_w(p) &= V(p) \cap W_w. \end{aligned}$$

Proposition 59.7. $\mathfrak{M}, w \Vdash \varphi$ *iff* $\mathfrak{M}_w \Vdash \varphi$. int:sem:sem:prop:restrict

Problem 59.3. Prove Proposition 59.7.

Proposition 59.8. *Suppose for every model* $\mathfrak{M}$ *such that* $\mathfrak{M} \Vdash \Gamma$, $\mathfrak{M} \Vdash \varphi$. *Then* $\Gamma \vDash \varphi$.

Proof. Suppose that $\mathfrak{M}, w \Vdash \Gamma$. By the Proposition 59.7 applied to every $\psi \in \Gamma$, we have $\mathfrak{M}_w \Vdash \Gamma$. By the assumption, we have $\mathfrak{M}_w \Vdash \varphi$. By Proposition 59.7 again, we get $\mathfrak{M}, w \Vdash \varphi$. □

---

content/intuitionistic-logic/semantics/topological-semantics.tex

59.4 Topological Semantics

int:sem:top:sec

Another way to provide a semantics for intuitionistic logic is using the mathematical concept of a topology.

Definition 59.9. Let X be a set. A *topology on* X is a set $\mathcal{O} \subseteq \wp(X)$ that satisfies the properties below. The elements of $\mathcal{O}$ are called the *open sets* of the topology. The set X together with $\mathcal{O}$ is called a *topological space*.

1. The empty set and the entire space are open: $\emptyset$, $X \in \mathcal{O}$.
2. Open sets are closed under finite intersections: if U, $V \in \mathcal{O}$ then $U \cap V \in \mathcal{O}$
3. Open sets are closed under arbitrary unions: if $U_i \in \mathcal{O}$ for all $i \in I$, then $\bigcup\{U_i : i \in I\} \in \mathcal{O}$.

We may write X for a topology if the collection of open sets can be inferred from the context; note that, still, only after X is endowed with open sets can it be called a topology.

Definition 59.10. A *topological model* of intuitionistic propositional logic is a triple $\mathfrak{X} = \langle X, \mathcal{O}, V\rangle$ where $\mathcal{O}$ is a topology on X and V is a function assigning an open set in $\mathcal{O}$ to each propositional variable.

Given a topological model $\mathfrak{X}$, we can define $[\![\varphi]\!]_{\mathfrak{X}}$ inductively as follows:

1. $[\![\bot]\!]_{\mathfrak{X}} = \emptyset$
2. $[\![p]\!]_{\mathfrak{X}} = V(p)$
3. $[\![\varphi \wedge \psi]\!]_{\mathfrak{X}} = [\![\varphi]\!]_{\mathfrak{X}} \cap [\![\psi]\!]_{\mathfrak{X}}$
4. $[\![\varphi \vee \psi]\!]_{\mathfrak{X}} = [\![\varphi]\!]_{\mathfrak{X}} \cup [\![\psi]\!]_{\mathfrak{X}}$
5. $[\![\varphi \to \psi]\!]_{\mathfrak{X}} = \mathrm{Int}((X \setminus [\![\varphi]\!]_{\mathfrak{X}}) \cup [\![\psi]\!]_{\mathfrak{X}})$

Here, $\mathrm{Int}(V)$ is the function that maps a set $V \subseteq X$ to its *interior*, that is, the union of all open sets it contains. In other words,

$$\mathrm{Int}(V) = \bigcup\{U : U \subseteq V \text{ and } U \in \mathcal{O}\}.$$

Note that the interior of any set is always open, since it is a union of open sets. Thus, $[\![\varphi]\!]_{\mathfrak{X}}$ is always an open set.

Although topological semantics is highly abstract, there are ways to think about it that might motivate it. Suppose that the elements, or "points," of X are points at which statements can be evaluated. The set of all points where φ is true is the proposition expressed by φ. Not every set of points is a potential proposition; only the elements of $\mathcal{O}$ are. $\varphi \vDash \psi$ iff ψ is true at every point at which φ is true, i.e., $[\![\varphi]\!]_{\mathfrak{X}} \subseteq [\![\psi]\!]_{\mathfrak{X}}$, for all X. The absurd statement $\bot$ is never true, so $[\![\bot]\!]_{\mathfrak{X}} = \emptyset$.

How must the propositions expressed by $\psi \wedge \chi$, $\psi \vee \chi$, and $\psi \to \chi$ be related to those expressed by ψ and χ for the intuitionistically valid laws to hold, i.e., so that $\varphi \vdash \psi$ iff $[\varphi]_{\mathfrak{X}} \subset [\psi]_{\mathfrak{X}}$? We require $\bot \vdash \varphi$ for any φ, which is satisfied because $\emptyset \subseteq U$ for all U. Since $\psi \wedge \chi \vdash \psi$, we require that $[\psi \wedge \chi]_{\mathfrak{X}} \subseteq [\psi]_{\mathfrak{X}}$, and similarly $[\psi \wedge \chi]_{\mathfrak{X}} \subseteq [\chi]_{\mathfrak{X}}$. The largest set satisfying $W \subseteq U$ and $W \subseteq V$ is $U \cap V$. Conversely, $\psi \vdash \psi \vee \chi$ and $\chi \vdash \psi \vee \chi$, and so we require that $[\psi]_{\mathfrak{X}} \subseteq [\psi \vee \chi]_{\mathfrak{X}}$ and $[\chi]_{\mathfrak{X}} \subseteq [\psi \vee \chi]_{\mathfrak{X}}$. The smallest set W such that $U \subseteq W$ and $V \subseteq W$ is $U \cup V$.

The definition for $\to$ is tricky: $\varphi \to \psi$ expresses the weakest proposition that, combined with φ, entails ψ. That $\varphi \to \psi$ combined with φ entails ψ is clear from $(\varphi \to \psi) \wedge \varphi \vdash \psi$. So $[\varphi \to \psi]_{\mathfrak{X}}$ should be the greatest open set such that $[\varphi \to \psi]_{\mathfrak{X}} \cap [\varphi]_{\mathfrak{X}} \subset [\psi]_{\mathfrak{X}}$, leading to our definition.

Chapter 60

Soundness and Completeness

This chapter collects soundness and completeness results for propositional intuitionistic logic. It needs an introduction. The completeness proof makes use of facts about provability that should be stated and proved explicitly somewhere.

---

content/intuitionistic-logic/soundness-completeness/soundness-axd.tex

60.1 Soundness of Axiomatic Derivations

int:sc:sax:sec

The soundness proof relies on the fact that all axioms are intuitionistically valid; this still needs to be proved, e.g., in the Semantics chapter.

Theorem 60.1 (Soundness). *If $\Gamma \vdash \varphi$, then $\Gamma \vDash \varphi$.*

int:sc:sax:thm:soundness

Proof. We prove that if $\Gamma \vdash \varphi$, then $\Gamma \vDash \varphi$. The proof is by induction on the number n of formulas in the derivation of φ from Γ. We show that if φ_1, ..., $\varphi_n = \varphi$ is a derivation from Γ, then $\Gamma \vDash \varphi_n$. Note that if φ_1, ..., φ_n is a derivation, so is φ_1, ..., φ_k for any $k < n$.

There are no derivations of length 0, so for $n = 0$ the claim holds vacuously. So the claim holds for all derivations of length $< n$. We distinguish cases according to the justification of φ_n.

1. φ_n is an axiom. All axioms are valid, so $\Gamma \vDash \varphi_n$ for any Γ.

2. $\varphi_n \in \Gamma$. Then for any $\mathfrak{M}$ and w, if $\mathfrak{M}, w \Vdash \Gamma$, obviously $\mathfrak{M} \Vdash \Gamma\varphi_n[w]$, i.e., $\Gamma \vDash \varphi$.

3. φ_n follows by MP from φ_i and $\varphi_j \equiv \varphi_i \to \varphi_n$. φ_1, ..., φ_i and φ_1, ..., φ_j are derivations from Γ, so by inductive hypothesis, $\Gamma \vDash \varphi_i$ and $\Gamma \vDash \varphi_i \to \varphi_n$.

   Suppose $\mathfrak{M}, w \Vdash \Gamma$. Since $\mathfrak{M}, w \Vdash \Gamma$ and $\Gamma \vDash \varphi_i \to \varphi_n$, $\mathfrak{M}, w \Vdash \varphi_i \to \varphi_n$. By definition, this means that for all w' such that Rww', if $\mathfrak{M}, w' \Vdash \varphi_i$ then $\mathfrak{M}, w' \Vdash \varphi_n$. Since R is reflexive, w is among the w' such that Rww', i.e., we have that if $\mathfrak{M}, w \Vdash \varphi_i$ then $\mathfrak{M}, w \Vdash \varphi_n$. Since $\Gamma \vDash \varphi_i$, $\mathfrak{M}, w \Vdash \varphi_i$. So, $\mathfrak{M}, w \Vdash \varphi_n$, as we wanted to show. □

---

content/intuitionistic-logic/soundness-completeness/soundness-nd.tex

60.2 Soundness of Natural Deduction

int:sc:snd:sec

We will now prove soundness of natural deduction with regards to the relational semantics, that is, showing that if a formula is derivable from a set of assumptions then the set of assumptions entails the formula.

int:sc:snd:thm:soundness

Theorem 60.2 (Soundness). *If $\Gamma \vdash \varphi$, then $\Gamma \vDash \varphi$.*

Proof. We prove that if $\Gamma \vdash \varphi$, then $\Gamma \vDash \varphi$. The proof is by induction on the derivation of φ from Γ.

1. If the derivation consists of just the assumption φ, we have $\varphi \vdash \varphi$, and want to show that $\varphi \vDash \varphi$. Suppose that $\mathfrak{M}, w \Vdash \varphi$. Then trivially $\mathfrak{M}, w \Vdash \varphi$.

2. The derivation ends in ∧Intro: The derivations of the premises ψ from undischarged assumptions Γ and of χ from undischarged assumptions Δ show that $\Gamma \vdash \psi$ and $\Delta \vdash \chi$. By induction hypothesis we have that $\Gamma \vDash \psi$ and $\Delta \vDash \chi$. We have to show that $\Gamma \cup \Delta \vDash \varphi \wedge \psi$, since the undischarged assumptions of the entire derivation are Γ together with Δ. So suppose $\mathfrak{M}, w \Vdash \Gamma \cup \Delta$. Then also $\mathfrak{M}, w \Vdash \Gamma$. Since $\Gamma \vDash \psi$, $\mathfrak{M}, w \Vdash \psi$. Similarly, $\mathfrak{M}, w \Vdash \chi$. So $\mathfrak{M}, w \Vdash \psi \wedge \chi$.

3. The derivation ends in $\wedge$Elim: The derivation of the premise $\psi \wedge \chi$ from undischarged assumptions Γ shows that $\Gamma \vdash \psi \wedge \chi$. By induction hypothesis, $\Gamma \vDash \psi \wedge \chi$. We have to show that $\Gamma \vDash \psi$. So suppose $\mathfrak{M}, w \Vdash \Gamma$. Since $\Gamma \vDash \psi \wedge \chi$, $\mathfrak{M}, w \Vdash \psi \wedge \chi$. Then also $\mathfrak{M}, w \Vdash \psi$. Similarly if $\wedge$Elim ends in χ, then $\Gamma \vDash \chi$.

4. The derivation ends in $\vee$Intro: Suppose the premise is ψ, and the undischarged assumptions of the derivation ending in ψ are Γ. Then we have $\Gamma \vdash \psi$ and by inductive hypothesis, $\Gamma \vDash \psi$. We have to show that $\Gamma \vDash \psi \vee \chi$. Suppose $\mathfrak{M}, w \Vdash \Gamma$. Since $\Gamma \vDash \psi$, $\mathfrak{M}, w \Vdash \psi$. But then also $\mathfrak{M}, w \Vdash \psi \vee \chi$. Similarly, if the premise is χ, we have that $\Gamma \vDash \chi$.

5. The derivation ends in $\vee$Elim: The derivations ending in the premises are of $\psi \vee \chi$ from undischarged assumptions Γ, of θ from undischarged assumptions $\Delta_1 \cup \{\psi\}$, and of θ from undischarged assumptions $\Delta_2 \cup \{\chi\}$. So we have $\Gamma \vdash \psi \vee \chi$, $\Delta_1 \cup \{\psi\} \vdash \theta$, and $\Delta_2 \cup \{\chi\} \vdash \theta$. By induction hypothesis, $\Gamma \vDash \psi \vee \chi$, $\Delta_1 \cup \{\psi\} \vDash \theta$, and $\Delta_2 \cup \{\chi\} \vDash \theta$. We have to prove that $\Gamma \cup \Delta_1 \cup \Delta_2 \vDash \theta$.

   Suppose $\mathfrak{M}, w \Vdash \Gamma \cup \Delta_1 \cup \Delta_2$. Then $\mathfrak{M}, w \Vdash \Gamma$ and since $\Gamma \vDash \psi \vee \chi$, $\mathfrak{M}, w \Vdash \psi \vee \chi$. By definition of $\mathfrak{M} \Vdash$, either $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$. So we distinguish cases: (a) $\mathfrak{M} \Vdash \psi$[w]. Then $\mathfrak{M}, w \Vdash \Delta_1 \cup \{\psi\}$. Since $\Delta_1 \cup \psi \vDash \theta$, we have $\mathfrak{M}, w \Vdash \theta$. (b) $\mathfrak{M}, w \Vdash \chi$. Then $\mathfrak{M}, w \Vdash \Delta_2 \cup \{\chi\}$. Since $\Delta_2 \cup \chi \vDash \theta$, we have $\mathfrak{M}, w \Vdash \theta$. So in either case, $\mathfrak{M}, w \Vdash \theta$, as we wanted to show.

6. The derivation ends with $\rightarrow$Intro concluding $\psi \rightarrow \chi$. Then the premise is χ, and the derivation ending in the premise has undischarged assumptions $\Gamma \cup \{\psi\}$. So we have that $\Gamma \cup \{\psi\} \vdash \chi$, and by induction hypothesis that $\Gamma \cup \{\psi\} \vDash \chi$. We have to show that $\Gamma \vDash \psi \rightarrow \chi$.

   Suppose $\mathfrak{M}, w \Vdash \Gamma$. We want to show that for all w' such that Rww', if $\mathfrak{M}, w' \Vdash \psi$, then $\mathfrak{M}, w' \Vdash \chi$. So assume that Rww' and $\mathfrak{M}, w' \Vdash \psi$. By Proposition 59.3, $\mathfrak{M}, w' \Vdash \Gamma$. Since $\Gamma \cup \{\psi\} \vDash \chi$, $\mathfrak{M}, w' \Vdash \chi$, which is what we wanted to show.

7. The derivation ends in $\rightarrow$Elim and conclusion χ. The premises are $\psi \rightarrow \chi$ and ψ, with derivations from undischarged assumptions Γ, Δ. So we have $\Gamma \vdash \psi \rightarrow \chi$ and $\Delta \vdash \psi$. By inductive hypothesis, $\Gamma \vDash \psi \rightarrow \chi$ and $\Delta \vDash \psi$. We have to show that $\Gamma \cup \Delta \vDash \chi$.

   Suppose $\mathfrak{M}, w \Vdash \Gamma \cup \Delta$. Since $\mathfrak{M}, w \Vdash \Gamma$ and $\Gamma \vDash \psi \rightarrow \chi$, $\mathfrak{M}, w \Vdash \psi \rightarrow \chi$. By definition, this means that for all w' such that Rww', if $\mathfrak{M}, w' \Vdash \psi$ then $\mathfrak{M}, w' \Vdash \chi$. Since R is reflexive, w is among the w' such that Rww', i.e., we have that if $\mathfrak{M}, w \Vdash \psi$ then $\mathfrak{M}, w \Vdash \chi$. Since $\mathfrak{M}, w \Vdash \Delta$ and $\Delta \vDash \psi$, $\mathfrak{M}, w \Vdash \psi$. So, $\mathfrak{M}, w \Vdash \chi$, as we wanted to show.

8. The derivation ends in $\bot_I$, concluding φ. The premise is $\bot$ and the undischarged assumptions of the derivation of the premise are Γ. Then $\Gamma \vdash \bot$. By inductive hypothesis, $\Gamma \vDash \bot$. We have to show $\Gamma \vDash \varphi$.

We proceed indirectly. If $\Gamma \nvdash \varphi$ there is a model $\mathfrak{M}$ and world w such that $\mathfrak{M}, w \Vdash \Gamma$ and $\mathfrak{M}, w \nVdash \varphi$. Since $\Gamma \vDash \bot$, $\mathfrak{M}, w \Vdash \bot$. But that's impossible, since by definition, $\mathfrak{M}, w \nVdash \bot$. So $\Gamma \vDash \varphi$.

9. The derivation ends in ¬Intro: Exercise.

10. The derivation ends in ¬Elim: Exercise. □

Problem 60.1. Complete the proof of Theorem 60.2. For the cases for ¬Intro and ¬Elim, use the definition of $\mathfrak{M}, w \Vdash \neg\varphi$ in Definition 59.2, i.e., don't treat $\neg\varphi$ as defined by $\varphi \to \bot$.

Problem 60.2. Show that the following formulas are not derivable in intuitionistic logic:

1. $(\varphi \to \psi) \vee (\psi \to \varphi)$
2. $(\neg\neg\varphi \to \varphi) \to (\varphi \vee \neg\varphi)$
3. $(\varphi \to \psi \vee \chi) \to \big((\varphi \to \psi) \vee (\varphi \to \chi)\big)$

---

content/intuitionistic-logic/soundness-completeness/lindenbaum.tex

60.3 Lindenbaum's Lemma

int:sc:lin:sec

The completeness theorem for intuitionistic logic is proved by assuming $\Gamma \nvdash \varphi$ and constructing a model $\mathfrak{M} \Vdash \Gamma$ and $\mathfrak{M} \nVdash \varphi$.

In classical logic the relation of derivability can be reduced to the notion of consistency since a formula φ is derivable from a set of formulas iff the set together with the negation of φ is inconsistent. This is not possible in intuitionistic logic. In intuitionistic logic, if $\neg\varphi$ is inconsistent, we only get that $\vdash \neg\neg\varphi$. Since $\neg\neg\varphi \to \varphi$ does not hold intuitionistically in general, we cannot conclude that $\vdash \varphi$.

Thus, when constructing the model $\mathfrak{M}$, we will need to keep track of the non-derivability of the formula φ and thus we will not be able to use a complete set $\Gamma^* \supseteq \Gamma$ to build the model $\mathfrak{M}$, as in every complete set Γ^*, we have $\Gamma^* \vdash \varphi \vee \neg\varphi$.

Instead of using a complete set Γ^*, we will us the notion of a prime set of formulas:

int:sc:lin:defn:prime

Definition 60.3. A set of formulas Γ is *prime* iff

int:sc:lin:defn:prime1
1. Γ is consistent, i.e., $\Gamma \nvdash \bot$;

int:sc:lin:defn:prime2
2. if $\Gamma \vdash \varphi$ then $\varphi \in \Gamma$; and

int:sc:lin:defn:prime3
3. if $\varphi \vee \psi \in \Gamma$ then $\varphi \in \Gamma$ or $\psi \in \Gamma$.

Lemma 60.4 (Lindenbaum's Lemma). *If $\Gamma \nvdash \varphi$, there is a $\Gamma^* \supseteq \Gamma$ such that Γ^* is prime and $\Gamma^* \nvdash \varphi$.* int:sc:lin: lem:lindenbaum

Proof. Let $\psi_1 \vee \chi_1$, $\psi_2 \vee \chi_2$, ..., be an enumeration of all formulas of the form $\psi \vee \chi$. We'll define an increasing sequence of sets of formulas Γ_n, where each Γ_{n+1} is defined as Γ_n together with one new formula. Γ^* will be the union of all Γ_n. The new formulas are selected so as to ensure that Γ^* is prime and still $\Gamma^* \nvdash \varphi$. This means that at each step we should find the first disjunction $\psi_i \vee \chi_i$ such that:

1. $\Gamma_n \vdash \psi_i \vee \chi_i$ int:sc:lin: gamma-1
2. $\psi_i \notin \Gamma_n$ and $\chi_i \notin \Gamma_n$ int:sc:lin: gamma-2

We add to Γ_n either ψ_i if $\Gamma_n \cup \{\psi_i\} \nvdash \varphi$, or χ_i otherwise. We'll have to show that this works. For now, let's define $i(n)$ as the least i such that (1) and (2) hold.

Define $\Gamma_0 = \Gamma$ and

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\psi_{i(n)}\} & \text{if } \Gamma_n \cup \{\psi_{i(n)}\} \nvdash \varphi \\ \Gamma_n \cup \{\chi_{i(n)}\} & \text{otherwise} \end{cases}$$

If $i(n)$ is undefined, i.e., whenever $\Gamma_n \vdash \psi \vee \chi$, either $\psi \in \Gamma_n$ or $\chi \in \Gamma_n$, we let $\Gamma_{n+1} = \Gamma_n$. Now let $\Gamma^* = \bigcup_{n=0}^{\infty} \Gamma_n$

First we show that for all n, $\Gamma_n \nvdash \varphi$. We proceed by induction on n. For $n = 0$ the claim holds by the hypothesis of the theorem, i.e., $\Gamma \nvdash \varphi$. If $n > 0$, we have to show that if $\Gamma_n \nvdash \varphi$ then $\Gamma_{n+1} \nvdash \varphi$. If $i(n)$ is undefined, $\Gamma_{n+1} = \Gamma_n$ and there is nothing to prove. So suppose $i(n)$ is defined. For simplicity, let $i = i(n)$.

We'll prove the contrapositive of the claim. Suppose $\Gamma_{n+1} \vdash \varphi$. By construction, $\Gamma_{n+1} = \Gamma_n \cup \{\psi_i\}$ if $\Gamma_n \cup \{\psi_i\} \nvdash \varphi$, or else $\Gamma_{n+1} = \Gamma_n \cup \{\chi_i\}$. It clearly can't be the first, since then $\Gamma_{n+1} \nvdash \varphi$. Hence, $\Gamma_n \cup \{\psi_i\} \vdash \varphi$ and $\Gamma_{n+1} = \Gamma_n \cup \{\chi_i\}$. By definition of $i(n)$, we have that $\Gamma_n \vdash \psi_i \vee \chi_i$. We have $\Gamma_n \cup \{\psi_i\} \vdash \varphi$. We also have $\Gamma_{n+1} = \Gamma_n \cup \{\chi_i\} \vdash \varphi$. Hence, $\Gamma_n \vdash \varphi$, which is what we wanted to show.

If $\Gamma^* \vdash \varphi$, there would be some finite subset $\Gamma' \subseteq \Gamma^*$ such that $\Gamma' \vdash \varphi$. Each $\theta \in \Gamma'$ must be in Γ_i for some i. Let n be the largest of these. Since $\Gamma_i \subseteq \Gamma_n$ if $i \leq n$, $\Gamma' \subseteq \Gamma_n$. But then $\Gamma_n \vdash \varphi$, contrary to our proof above that $\Gamma_n \nvdash \varphi$.

Lastly, we show that Γ^* is prime, i.e., satisfies conditions (1), (2), and (3) of Definition 60.3.

First, $\Gamma^* \nvdash \varphi$, so Γ^* is consistent, so (1) holds.

We now show that if $\Gamma^* \vdash \psi \vee \chi$, then either $\psi \in \Gamma^*$ or $\chi \in \Gamma^*$. This proves (3), since if $\psi \vee \chi \in \Gamma^*$ then also $\Gamma^* \vdash \psi \vee \chi$. So assume $\Gamma^* \vdash \psi \vee \chi$ but $\psi \notin \Gamma^*$ and $\chi \notin \Gamma^*$. Since $\Gamma^* \vdash \psi \vee \chi$, $\Gamma_n \vdash \psi \vee \chi$ for some n. $\psi \vee \chi$ appears on the enumeration of all disjunctions, say, as $\psi_j \vee \chi_j$. $\psi_j \vee \chi_j$ satisfies the properties in the definition of $i(n)$, namely we have $\Gamma_n \vdash \psi_j \vee \chi_j$, while $\psi_j \notin \Gamma_n$

and $\chi_j \notin \Gamma_n$. At each stage, at least one fewer disjunction $\psi_i \vee \chi_i$ satisfies the conditions (since at each stage we add either ψ_i or χ_i), so at some stage m we will have $j = i(m)$. But then either $\psi \in \Gamma_{m+1}$ or $\chi \in \Gamma_{m+1}$, contrary to the assumption that $\psi \notin \Gamma^*$ and $\chi \notin \Gamma^*$.

Now suppose $\Gamma^* \vdash \psi$. Then $\Gamma^* \vdash \psi \vee \psi$. But we've just proved that if $\Gamma^* \vdash \psi \vee \psi$ then $\psi \in \Gamma^*$. Hence, Γ^* satisfies (2) of Definition 60.3. □

Problem 60.3. Show that if $\Gamma \nvdash \bot$ then Γ is consistent in classical logic, i.e., there is a valuation making all formulas in Γ true.

---

content/intuitionistic-logic/soundness-completeness/canonical-model.tex

60.4 The Canonical Model

int:sc:mod:sec

The worlds in our model will be finite sequences σ of natural numbers, i.e., $\sigma \in \mathbb{N}^*$. Note that $\mathbb{N}^*$ is inductively defined by:

1. $\Lambda \in \mathbb{N}^*$.
2. If $\sigma \in \mathbb{N}^*$ and $n \in \mathbb{N}$, then $\sigma.n \in \mathbb{N}^*$ (where $\sigma.n$ is $\sigma \frown \langle n \rangle$ and $\sigma \frown \sigma'$ is the concatenation if σ and σ').
3. Nothing else is in $\mathbb{N}^*$.

So we can use $\mathbb{N}^*$ to give inductive definitions.

Let $\langle \psi_1, \chi_1 \rangle$, $\langle \psi_2, \chi_2 \rangle$, ..., be an enumeration of all pairs of formulas. Given a set of formulas Δ, define $\Delta(\sigma)$ by induction as follows:

1. $\Delta(\Lambda) = \Delta$
2. $\Delta(\sigma.n) =$
$$\begin{cases} (\Delta(\sigma) \cup \{\psi_n\})^* & \text{if } \Delta(\sigma) \cup \{\psi_n\} \nvdash \chi_n \\ \Delta(\sigma) & \text{otherwise} \end{cases}$$

Here by $(\Delta(\sigma) \cup \{\psi_n\})^*$ we mean the prime set of formulas which exists by Lemma 60.4 applied to the set $\Delta(\sigma) \cup \{\psi_n\}$ and the formula χ_n. Note that by this definition, if $\Delta(\sigma) \cup \{\psi_n\} \nvdash \chi_n$, then $\Delta(\sigma.n) \vdash \psi_n$ and $\Delta(\sigma.n) \nvdash \chi_n$. Note also that $\Delta(\sigma) \subseteq \Delta(\sigma.n)$ for any n. If Δ is prime, then $\Delta(\sigma)$ is prime for all σ.

int:sc:mod:defn:canonical-model

Definition 60.5. Suppose Δ is prime. Then the *canonical model* $\mathfrak{M}(\Delta)$ for Δ is defined by:

1. $W = \mathbb{N}^*$, the set of finite sequences of natural numbers.
2. R is the partial order according to which $R\sigma\sigma'$ iff σ is an initial segment of σ' (i.e., $\sigma' = \sigma \frown \sigma''$ for some sequence σ'').
3. $V(p) = \{\sigma : p \in \Delta(\sigma)\}$.

It is easy to verify that R is indeed a partial order. Also, the monotonicity condition on V is satisfied. Since $\Delta(\sigma) \subseteq \Delta(\sigma.n)$ we get $\Delta(\sigma) \subseteq \Delta(\sigma')$ whenever $R\sigma\sigma'$ by induction on σ.

---

content/intuitionistic-logic/soundness-completeness/truth-lemma.tex

60.5 The Truth Lemma

The following lemma connects satisfaction in the canonical model with which formulas are elements of the prime set Δ. int:sc:truth-sec

Lemma 60.6. *If Δ is prime, then $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$ iff $\Delta(\sigma) \vdash \varphi$.* int:sc:truth-lem:truth

Proof. By induction on φ.

1. $\varphi \equiv \bot$: Since $\Delta(\sigma)$ is prime, it is consistent, so $\Delta(\sigma) \nvdash \varphi$. By definition, $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

2. $\varphi \equiv p$: By definition of $\Vdash$, $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$ iff $\sigma \in V(p)$, i.e., $\Delta(\sigma) \vdash \varphi$.

3. $\varphi \equiv \neg\psi$: exercise.

4. $\varphi \equiv \psi \wedge \chi$: $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$ iff $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ and $\mathfrak{M}(\Delta), \sigma \Vdash \chi$. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ iff $\Delta(\sigma) \vdash \psi$, and similarly for χ. But $\Delta(\sigma) \vdash \psi$ and $\Delta(\sigma) \vdash \chi$ iff $\Delta(\sigma) \vdash \varphi$.

5. $\varphi \equiv \psi \vee \chi$: $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$ iff $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ or $\mathfrak{M}(\Delta), \sigma \Vdash \chi$. By induction hypothesis, this holds iff $\Delta(\sigma) \vdash \psi$ or $\Delta(\sigma) \vdash \chi$. We have to show that this in turn holds iff $\Delta(\sigma) \vdash \varphi$. The left-to-right direction is clear. The right-to-left direction follows since $\Delta(\sigma)$ is prime.

6. $\varphi \equiv \psi \to \chi$: First the contrapositive of the left-to-right direction: Assume $\Delta(\sigma) \nvdash \psi \to \chi$. Then also $\Delta(\sigma) \cup \{\psi\} \nvdash \chi$. Since $\langle \psi, \chi \rangle$ is $\langle \psi_n, \chi_n \rangle$ for some n, we have $\Delta(\sigma.n) = (\Delta(\sigma) \cup \{\psi\})^*$, and $\Delta(\sigma.n) \vdash \psi$ but $\Delta(\sigma.n) \nvdash \chi$. By inductive hypothesis, $\mathfrak{M}(\Delta), \sigma.n \Vdash \psi$ and $\mathfrak{M}(\Delta), \sigma.n \nVdash \chi$. Since $R\sigma(\sigma.n)$, this means that $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

   Now assume $\Delta(\sigma) \vdash \psi \to \chi$, and let $R\sigma\sigma'$. Since $\Delta(\sigma) \subseteq \Delta(\sigma')$, we have: if $\Delta(\sigma') \vdash \psi$, then $\Delta(\sigma') \vdash \chi$. In other words, for every σ' such that $R\sigma\sigma'$, either $\Delta(\sigma') \nvdash \psi$ or $\Delta(\sigma') \vdash \chi$. By induction hypothesis, this means that whenever $R\sigma\sigma'$, either $\mathfrak{M}(\Delta), \sigma' \nVdash \psi$ or $\mathfrak{M}(\Delta), \sigma' \Vdash \chi$, i.e., $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$. □

---

content/intuitionistic-logic/soundness-completeness/completeness-thm.tex

60.6 The Completeness Theorem

int:sc:cpl:sec
int:sc:cpl:thm:completeness

Theorem 60.7. *If $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$.*

Proof. We prove the contrapositive: Suppose $\Gamma \nvdash \varphi$. Then by Lemma 60.4, there is a prime set $\Gamma^* \supseteq \Gamma$ such that $\Gamma^* \nvdash \varphi$. Consider the canonical model $\mathfrak{M}(\Gamma^*)$ for Γ^* as defined in Definition 60.5. For any $\psi \in \Gamma$, $\Gamma^* \vdash \psi$. Note that $\Gamma^*(\Lambda) = \Gamma^*$. By the Truth Lemma (Lemma 60.6), we have $\mathfrak{M}(\Gamma^*), \Lambda \Vdash \psi$ for all $\psi \in \Gamma$ and $\mathfrak{M}(\Gamma^*), \Lambda \nVdash \varphi$. This shows that $\Gamma \nvDash \varphi$. □

Problem 60.4. Show that if φ only contains propositional variables, $\vee$, and $\wedge$, then $\nvDash \varphi$. Use this to conclude that $\to$ is not definable in intuitionistic logic from $\vee$ and $\wedge$.

Problem 60.5. By using the completeness theorem prove that if $\vdash \varphi \vee \psi$ then $\vdash \varphi$ or $\vdash \psi$. (Hint: Assume $\mathfrak{M}_1 \nVdash \varphi$ and $\mathfrak{M}_2 \nVdash \psi$ and construct a new model $\mathfrak{M}$ such that $\mathfrak{M} \nVdash \varphi \vee \psi$.)

Problem 60.6. Show that if $\mathfrak{M}$ is a relational model using a linear order then $\mathfrak{M} \Vdash (\varphi \to \psi) \vee (\psi \to \varphi)$.

---

content/intuitionistic-logic/soundness-completeness/decidability.tex

60.7 Decidability

int:sc:dec:sec

Observe that the proof of the completeness theorem gives us for every $\Gamma \nvdash \varphi$ a model with an infinite number of worlds witnessing the fact that $\Gamma \nvDash \varphi$. The following proposition shows that to prove $\vDash \varphi$ it is enough to prove that $\mathfrak{M} \Vdash \varphi$ for all finite models (i.e., models with a finite set of worlds).

int:sc:dec:thm:decidability

Theorem 60.8. *If $\nvDash \varphi$ then there is a finite model $\mathfrak{M}' \nVdash \varphi$.*

Proof. Assume $\mathfrak{M} = \langle W, R, V\rangle$ is such that $\mathfrak{M} \nVdash \varphi$ and P is the set of propositional variables occurring in φ. Define $\mathfrak{M}' = \langle W', R', V'\rangle$ by letting $W' = \{[w] : w \in W\}$ where $[w] = \{p \in P : w \in V(p)\}$, R' be the subset relation, and $V'(p) = \{[w] : p \in [w]\}$. It should be clear that W' is a finite set and that $\mathfrak{M}'$ is a relational model.

It can be shown, by induction on φ, that

$$\mathfrak{M}, w \Vdash \varphi \text{ iff } \mathfrak{M}', [w] \Vdash \varphi$$

for all formulas φ with only propositional variables from P. This is left as an exercise for the reader. □

Problem 60.7. Finish the proof of Theorem 60.8 by showing that $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}', [w] \Vdash \varphi$ for all formulas φ with only propositional variables from P.

From Theorem 60.8 it follows that there is an algorithm to decide whether $\vDash \varphi$.

Chapter 61

Intuitionistic Tableaux

Draft chapter on prefixed tableaux for intuitionistic logic. Needs more examples, completeness proofs, and discussion of how one can find countermodels from unsuccessful searches for closed tableaux.

---

content/intuitionistic-logic/tableaux/introduction.tex

61.1 Introduction

int:tab:int:sec

Tableaux are certain (downward-branching) trees of signed formulas, i.e., pairs consisting of a truth value sign ($\mathbb{T}$ or $\mathbb{F}$) and a sentence

$$\mathbb{T}\varphi \text{ or } \mathbb{F}\varphi.$$

A tableau begins with a number of *assumptions*. Each further signed formula is generated by applying one of the inference rules. Some inference rules add one or more signed formulas to a tip of the tree; others add two new tips, resulting in two branches. Rules result in signed formulas where the formula is less complex than that of the signed formula to which it was applied. When a branch contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, we say the branch is *closed*. If every branch in a tableau is closed, the entire tableau is closed. A closed tableau constitutes a derivation that shows that the set of signed formulas which were used to begin the tableau are unsatisfiable. This can be used to define a $\vdash$ relation: $\Gamma \vdash \varphi$ iff there is some finite set $\Gamma_0 = \{\psi_1, \ldots, \psi_n\} \subseteq \Gamma$ such that there is a closed tableau for the assumptions

$$\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \ldots, \mathbb{T}\psi_n\}.$$

For intuitionistic logic, we have to both extend the notion of signed formula and adjust the rules for the connectives. In addition to a sign($\mathbb{T}$ or $\mathbb{F}$), formulas

$\dfrac{\sigma\,\mathbb{T}\,\varphi\wedge\psi}{\begin{array}{l}\sigma\,\mathbb{T}\,\varphi\\ \sigma\,\mathbb{T}\,\psi\end{array}}\wedge\mathbb{T}$	$\dfrac{\sigma\,\mathbb{F}\,\varphi\wedge\psi}{\sigma\,\mathbb{F}\,\varphi \quad\mid\quad \sigma\,\mathbb{F}\,\psi}\wedge\mathbb{F}$
$\dfrac{\sigma\,\mathbb{T}\,\varphi\vee\psi}{\sigma\,\mathbb{T}\,\varphi \quad\mid\quad \sigma\,\mathbb{T}\,\psi}\vee\mathbb{T}$	$\dfrac{\sigma\,\mathbb{F}\,\varphi\vee\psi}{\begin{array}{l}\sigma\,\mathbb{F}\,\varphi\\ \sigma\,\mathbb{F}\,\psi\end{array}}\vee\mathbb{F}$

Table 61.1: Prefixed tableau rules for $\wedge$ and $\vee$

int:tab:rul:tab:prop-rules

in modal tableaux also have *prefixes* σ. The prefixes are non-empty sequences of positive integers, i.e., $\sigma \in (\mathbb{Z}^+)^* \setminus \{\Lambda\}$. When we write such prefixes without the surrounding $\langle\ \rangle$, and separate the individual elements by .'s instead of ,'s. If σ is a prefix, then $\sigma.n$ is $\sigma \frown \langle n\rangle$; e.g., if $\sigma = 1.2.1$, then $\sigma.3$ is 1.2.1.3. So for instance,

$$1.2\,\mathbb{T}\,\varphi \to (\psi \to \chi)$$

is a *prefixed signed formula* (or just a *prefixed formula* for short).

Intuitively, the prefix names a world in a model that might satisfy the formulas on a branch of a tableau, and if σ names some world, then $\sigma.n$ names a world accessible from (the world named by) σ.

In intuitionistic models, the accessibility relation is reflexive and transitive. In terms of prefixes, this means that σ is accessible from σ itself, and so is any prefix that extends σ, i.e., any prefix of the form $\sigma.n_1.\cdots.n_k$. Let's introduce the notation $\sigma.*$ to indicate σ itself and any extension of it. In other words, the prefixes $\sigma.*$ are all and only the prefixes accessible from σ.

---

content/intuitionistic-logic/tableaux/rules.tex

61.2 Rules for Intuitionistic Logic

int:tab:rul:sec

The rules for the connectives $\wedge$ and $\vee$ are the same as for regular propositional signed tableaux, just with prefixes added. In each case, the rule applied to a signed formula $\sigma\,S\,\varphi$ produces new formulas that are also prefixed by σ. This should be intuitively clear: e.g., if $\varphi\wedge\psi$ is true at (a world named by) σ, then φ and ψ are true at σ (and not at any other world). We collect the rules for $\wedge$ and $\vee$ in Table 61.1.

The closure condition is similar to that for ordinary tableaux, although we require that not just the formulas, but also that the prefixes must match. In fact, we can be somewhat more liberal: Since in intuitionistic models, formulas, once true, remain true, it is impossible that φ is true at σ but false at any accessible prefix $\sigma.*$. So a branch is closed if it contains both

$$\sigma\,\mathbb{T}\,\varphi \quad\text{and}\quad \sigma.*\,\mathbb{F}\,\varphi$$

$\dfrac{\sigma\,\mathbb{T}\,\neg\varphi}{\sigma.*\,\mathbb{F}\,\varphi}\ \neg\mathbb{T}$ $\sigma.*$ is used	$\dfrac{\sigma\,\mathbb{F}\,\neg\varphi}{\sigma.n\,\mathbb{T}\,\varphi}\ \neg\mathbb{F}$ $\sigma.n$ is new
$\dfrac{\sigma\,\mathbb{T}\,\varphi\to\psi}{\sigma.*\,\mathbb{F}\,\varphi \quad\mid\quad \sigma.*\,\mathbb{T}\,\psi}\ \to\mathbb{T}$ $\sigma.*$ is used	$\dfrac{\sigma\,\mathbb{F}\,\varphi\to\psi}{\begin{array}{l}\sigma.n\,\mathbb{T}\,\varphi\\ \sigma.n\,\mathbb{F}\,\psi\end{array}}\ \to\mathbb{F}$ $\sigma.n$ is new

Table 61.2: Prefixed tableau rules for $\neg$ and $\to$

int:tab:rul:
tab:rules-lif-lnot

for some prefix σ and formula φ. Note that if the signs are reversed, i.e., if it contains

$$\sigma\,\mathbb{F}\,\varphi \quad \text{and} \quad \sigma.*\,\mathbb{T}\,\varphi$$

the branch is closed only if $*$ is the empty sequence.

In addition, a branch is closed if it contains $\sigma\,\mathbb{T}\,\bot$.

The rules for setting up assumptions is also as for ordinary tableaux, except that for assumptions we always use the prefix 1. (It does not matter which prefix we use, as long as it's the same for all assumptions.) So, e.g., we say that

$$\psi_1, \ldots, \psi_n \vdash \varphi$$

iff there is a closed tableau for the assumptions

$$1\,\mathbb{T}\,\psi_1, \ldots, 1\,\mathbb{T}\,\psi_n, 1\,\mathbb{F}\,\varphi.$$

For the conditional $\to$, the rules differ from the classical and modal cases. The $\mathbb{T}\to$ rule extends a branch containing $\sigma\,\mathbb{T}\,\varphi\to\psi$ by $\sigma.*\,\mathbb{T}\,\varphi$ and $\sigma.*\,\mathbb{F}\,\psi$ on two different branches. It can only be applied for a prefix $\sigma.*$ which *already* occurs on the branch in which it is applied. Let's call such a prefix "used" (on the branch). (Since $\sigma.*$ includes σ itself, the rule can always be applied by adding the prefixed signed formulas $\sigma\,\mathbb{T}\,\varphi$ and $\sigma\,\mathbb{F}\,\psi$ on separate branches.)

The $\mathbb{F}\to$ rule extends a branch containing $\sigma\,\mathbb{F}\,\varphi\to\psi$ by both $\sigma.n\,\mathbb{T}\,\varphi$ and $\sigma.n\,\mathbb{F}\,\psi$ on the same branch, with $\sigma.n$ a prefix new to the branch.

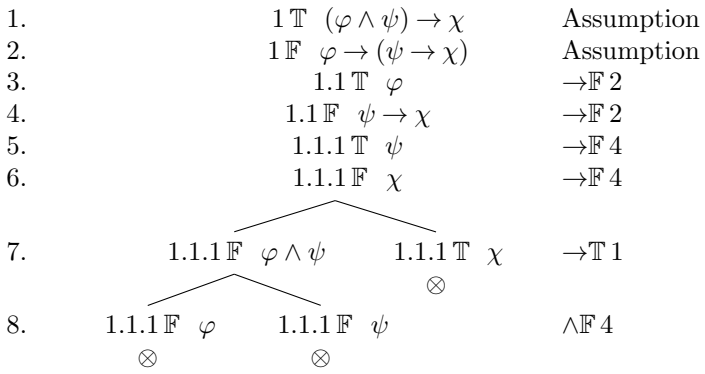
The rules for $\neg$ are defined analogously (using the definition of $\neg\varphi$ as $\varphi\to\bot$).

The rules are given in Table 61.2.

61.3 Tableaux for Intuitionistic Logic

int:tab:prf:sec

Example 61.1. We give a closed tableau that shows $(\varphi \wedge \psi) \to \chi \vdash \varphi \to (\psi \to \chi)$.



Problem 61.1. Find closed intuitionistic tableaux to show the following:

1. $\vdash \varphi \to (\psi \to \varphi)$
2. $\vdash \neg(\varphi \wedge \neg\varphi)$
3. $\varphi \to (\psi \to \chi) \vdash (\varphi \wedge \psi) \to \chi$
4. $\neg\varphi \vee \neg\psi \vdash \neg(\varphi \wedge \psi)$

---

content/intuitionistic-logic/tableaux/soundness.tex

61.4 Soundness for Intuitionistic Tableaux

int:tab:sou:sec

In order to show that intuitionistic tableaux are sound, we have to show that if explanation

$$1\,\mathbb{T}\,\psi_1, \ldots, 1\,\mathbb{T}\,\psi_n, 1\,\mathbb{F}\,\varphi$$

has a closed tableau then $\psi_1, \ldots, \psi_n \vDash \varphi$. It is easier to prove the contrapositive: if for some $\mathfrak{M}$ and world w, $\mathfrak{M}, w \Vdash \psi_i$ for all $i = 1, \ldots, n$ but $\mathfrak{M}, w \nVdash \varphi$, then no tableau can close. Such a countermodel shows that the initial assumptions of the tableau are satisfiable. The strategy of the proof is to show that whenever all the prefixed formulas on a tableau branch are satisfiable, any application of a rule results in at least one extended branch that is also satisfiable. Since closed branches are unsatisfiable, any tableau for a satisfiable set of prefixed formulas must have at least one open branch.

In order to apply this strategy in the modal case, we have to extend our definition of "satisfiable" to relational and prefixes. With that in hand, however, the proof is straightforward.

Definition 61.2. Let P be some set of prefixes, i.e., $P \subseteq (\mathbb{Z}^+)^* \setminus \{\Lambda\}$ and let $\mathfrak{M}$ be a model. A function $f\colon P \to W$ is an *interpretation of* P in $\mathfrak{M}$ if, whenever σ and $\sigma.n$ are both in P, then $Rf(\sigma)f(\sigma.n)$.

Relative to an interpretation of prefixes P we can define:

1. $\mathfrak{M}$ satisfies $\sigma\,\mathbb{T}\,\varphi$ iff $\mathfrak{M}, f(\sigma) \Vdash \varphi$.
2. $\mathfrak{M}$ satisfies $\sigma\,\mathbb{F}\,\varphi$ iff $\mathfrak{M}, f(\sigma) \nVdash \varphi$.

Note that since R is reflexive and transitive and $\sigma.*$ denotes, σ, $\sigma.n_1$, $\sigma.n_1.n_2$, ..., we also have that $Rf(\sigma)f(\sigma.*)$.

Definition 61.3. Let Γ be a set of prefixed formulas, and let $P(\Gamma)$ be the set of prefixes that occur in it. If f is an interpretation of $P(\Gamma)$ in $\mathfrak{M}$, we say that $\mathfrak{M}$ satisfies Γ with respect to f, $\mathfrak{M}, f \Vdash \Gamma$, if $\mathfrak{M}$ satisfies every prefixed formula in Γ with respect to f. Γ is *satisfiable* iff there is a model $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that $\mathfrak{M}, f \Vdash \Gamma$.

Proposition 61.4. *If Γ contains both $\sigma\,\mathbb{T}\,\varphi$ and $\sigma.*\,\mathbb{F}\,\varphi$ for some formula φ and prefix σ, or it contains $\sigma\,\mathbb{T}\,\bot$, then Γ is unsatisfiable.*

Proof. Since always $\mathfrak{M}, f(\sigma) \nVdash \bot$, a Γ that contains $\mathbb{T}\,\bot$ is unsatisfiable.

There also cannot be a model $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that both If $\mathfrak{M}, f(\sigma) \Vdash \varphi$, then by Proposition 59.3, since $Rf(\sigma)(\sigma.*)$, $\mathfrak{M}, f(\sigma) \Vdash \varphi$. So we cannot have both $\mathfrak{M}, f(\sigma) \Vdash \varphi$ and $\mathfrak{M}, f(\sigma.*) \nVdash \varphi$. □

Theorem 61.5 (Soundness). *If Γ has a closed tableau, Γ is unsatisfiable.* *int:tab:sou: thm:tableau-soundness*

Proof. We call a branch of a tableau satisfiable iff the set of signed formulas on it is satisfiable, and let's call a tableau satisfiable if it contains at least one satisfiable branch.

We show the following: Extending a satisfiable tableau by one of the rules of inference always results in a satisfiable tableau. This will prove the theorem: any closed tableau results by applying rules of inference to the tableau consisting only of assumptions from Γ. So if Γ were satisfiable, any tableau for it would be satisfiable. A closed tableau, however, is clearly not satisfiable, since all its branches are closed and closed branches are unsatisfiable.

Suppose we have a satisfiable tableau, i.e., a tableau with at least one satisfiable branch. Applying a rule of inference either adds signed formulas to a branch, or splits a branch in two. If the tableau has a satisfiable branch which is not extended by the rule application in question, it remains a satisfiable branch in the extended tableau, so the extended tableau is satisfiable. So we only have to consider the case where a rule is applied to a satisfiable branch.

Let Γ be the set of signed formulas on that branch, and let $\sigma\,S\,\varphi \in \Gamma$ be the signed formula to which the rule is applied. If the rule does not result in a split branch, we have to show that the extended branch, i.e., Γ together with the conclusions of the rule, is still satisfiable. If the rule results in split branch, we have to show that at least one of the two resulting branches is satisfiable.

1. The branch is expanded by applying $\neg\mathbb{T}$ to $\sigma\,\mathbb{T}\neg\psi \in \Gamma$. Then the extended branch contains the signed formulas $\Gamma \cup \{\sigma.*\,\mathbb{F}\,\psi\}$. Suppose $\mathfrak{M}, f \Vdash \Gamma$. In particular, $\mathfrak{M}, f(\sigma) \Vdash \neg\psi$. Thus, $\mathfrak{M}, w \nVdash \psi$ for any w such that $Rf(\sigma)w$, and that includes $f(\sigma.*)$. So, $\mathfrak{M}$ satisfies $\sigma.*\,\mathbb{F}\,\psi$ with respect to f.

2. The branch is expanded by applying $\neg\mathbb{F}$ to $\sigma\,\mathbb{F}\,\neg\psi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\wedge\mathbb{T}$ to $\sigma\,\mathbb{T}\,\psi \wedge \chi \in \Gamma$, which results in two new signed formulas on the branch: $\sigma\,\mathbb{T}\,\psi$ and $\sigma\,\mathbb{T}\,\chi$. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \Vdash \psi \wedge \chi$. Then $\mathfrak{M}, f(\sigma) \Vdash \psi$ and $\mathfrak{M}, f(\sigma) \Vdash \chi$. This means that $\mathfrak{M}$ satisfies both $\sigma\,\mathbb{T}\,\psi$ and $\sigma\,\mathbb{T}\,\chi$ with respect to f.

4. The branch is expanded by applying $\vee\mathbb{F}$ to $\mathbb{F}\,\psi \vee \chi \in \Gamma$: Exercise.

5. The branch is expanded by applying $\to\mathbb{F}$ to $\sigma\,\mathbb{F}\,\psi \to \chi \in \Gamma$: This results in two new signed formulas on the branch: $\sigma.n\,\mathbb{T}\,\psi$ and $\sigma.n\,\mathbb{F}\,\chi$, where $\sigma.n$ is a new prefix on the branch, i.e., $\sigma.n \notin P(\Gamma)$.

   Since Γ is satisfiable, there is a $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \nVdash \psi \to \chi$. We have to show that $\Gamma \cup \{\sigma.n\,\mathbb{F}\,\psi \to \chi\}$ is satisfiable. To do this, we define an interpretation of $P(\Gamma) \cup \{\sigma.n\}$ as follows:

   Since $\mathfrak{M}, f(\sigma) \nVdash \psi \to \chi$, there is a $w \in W$ such that $Rf(\sigma)w$ such that $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \nVdash \chi$. Let f' be like f, except that $f'(\sigma.n) = w$. Since $f'(\sigma) = f(\sigma)$ and $Rf(\sigma)w$, we have $Rf'(\sigma)f'(\sigma.n)$, so f' is an interpretation of $P(\Gamma) \cup \{\sigma.n\}$. Obviously $\mathfrak{M}, f'(\sigma.n) \Vdash \psi$ and $\mathfrak{M}, f'(\sigma.n) \nVdash \chi$. Since $f(\sigma') = f'(\sigma')$ for all prefixes $\sigma' \in P(\Gamma)$, $\mathfrak{M}, f' \Vdash \Gamma$. So, $\mathfrak{M}, f'$ satisfies $\Gamma \cup \{\sigma.n\,\mathbb{F}\,\psi \to \chi\}$

Now let's consider the possible inferences with two premises.

1. The branch is expanded by applying $\wedge\mathbb{F}$ to $\sigma\,\mathbb{F}\,\psi \wedge \chi \in \Gamma$, which results in two branches, a left one continuing through $\sigma\,\mathbb{F}\,\psi$ and a right one through $\sigma\,\mathbb{F}\,\chi$. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \nVdash \psi \wedge \chi$. Then $\mathfrak{M}, f(\sigma) \nVdash \psi$ or $\mathfrak{M}, f(\sigma) \nVdash \chi$. In the former case, $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{F}\,\psi$, i.e., the left branch is satisfiable. In the latter, $\mathfrak{M}, f$ satisfies $\sigma\,\mathbb{F}\,\chi$, i.e., the right branch is satisfiable.

2. The branch is expanded by applying $\vee\mathbb{T}$ to $\sigma\,\mathbb{T}\,\psi \vee \chi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\to\mathbb{T}$ to $\sigma\,\mathbb{T}\,\psi \to \chi \in \Gamma$: Exercise. □

Problem 61.2. Complete the proof of Theorem 61.5.

int:tab:sou: cor:entailment-soundness

Corollary 61.6. *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.*

Proof. If $\Gamma \vdash \varphi$ then for some $\psi_1, \ldots, \psi_n \in \Gamma$, $\Delta = \{1\,\mathbb{F}\,\varphi, 1\,\mathbb{T}\,\psi_1, \ldots, 1\,\mathbb{T}\,\psi_n\}$ has a closed tableau. We want to show that $\Gamma \vDash \varphi$. Suppose not, so for some $\mathfrak{M}$ and w, $\mathfrak{M}, w \Vdash \psi_i$ for $i = 1, \ldots, n$, but $\mathfrak{M}, w \nVdash \varphi$. Let $f(1) = w$; then f is an interpretation of $P(\Delta)$ into $\mathfrak{M}$, and $\mathfrak{M}$ satisfies Δ with respect to f. But by Theorem 61.5, Δ is unsatisfiable since it has a closed tableau, a contradiction. So we must have $\Gamma \vdash \varphi$ after all. □

Corollary 61.7. *If $\vdash \varphi$ then φ is true in all models.*

int:tab:sou: cor:weak-soundness

Part XIV

Counterfactuals

Chapter 62

Introduction

content/counterfactuals/introduction/material-conditional.tex

62.1 The Material Conditional

cnt:int:mat:sec

In its simplest form in English, a conditional is a sentence of the form "If ... then ...," where the ... are themselves sentences, such as "If the butler did it, then the gardener is innocent." In introductory logic courses, we earn to symbolize conditionals using the $\to$ connective: symbolize the parts indicated by ..., e.g., by formulas φ and ψ, and the entire conditional is symbolized by $\varphi \to \psi$.

The connective $\to$ is *truth-functional*, i.e., the truth value—$\mathbb{T}$ or $\mathbb{F}$—of $\varphi \to \psi$ is determined by the truth values of φ and ψ: $\varphi \to \psi$ is true iff φ is false or ψ is true, and false otherwise. Relative to a truth value assignment $\mathfrak{v}$, we define $\mathfrak{v} \vDash \varphi \to \psi$ iff $\mathfrak{v} \nvDash \varphi$ or $\mathfrak{v} \vDash \psi$. The connective $\to$ with this semantics is called the *material conditional*.

This definition results in a number of elementary logical facts. First of all, the deduction theorem holds for the material conditional:

$$\text{If } \Gamma, \varphi \vDash \psi \text{ then } \Gamma \vDash \varphi \to \psi \tag{62.1}$$

It is truth-functional: $\varphi \to \psi$ and $\neg\varphi \vee \psi$ are equivalent:

$$\varphi \to \psi \vDash \neg\varphi \vee \psi \tag{62.2}$$

$$\neg\varphi \vee \psi \vDash \varphi \to \psi \tag{62.3}$$

A material conditional is entailed by its consequent and by the negation of its antecedent:

$$\psi \vDash \varphi \to \psi \tag{62.4}$$
$$\neg\varphi \vDash \varphi \to \psi \tag{62.5}$$

A false material conditional is equivalent to the conjunction of its antecedent and the negation of its consequent: if $\varphi \to \psi$ is false, $\varphi \wedge \neg\psi$ is true, and vice versa:

$$\neg(\varphi \to \psi) \vDash \varphi \wedge \neg\psi \tag{62.6}$$
$$\varphi \wedge \neg\psi \vDash \neg(\varphi \to \psi) \tag{62.7}$$

The material conditional supports modus ponens:

$$\varphi, \varphi \to \psi \vDash \psi \tag{62.8}$$

The material conditional agglomerates:

$$\varphi \to \psi, \varphi \to \chi \vDash \varphi \to (\psi \wedge \chi) \tag{62.9}$$

We can always strengthen the antecedent, i.e., the conditional is *monotonic*:

$$\varphi \to \psi \vDash (\varphi \wedge \chi) \to \psi \tag{62.10}$$

The material conditional is transitive, i.e., the chain rule is valid:

$$\varphi \to \psi, \psi \to \chi \vDash \varphi \to \chi \tag{62.11}$$

The material conditional is equivalent to its contrapositive:

$$\varphi \to \psi \vDash \neg\psi \to \neg\varphi \tag{62.12}$$
$$\neg\psi \to \neg\varphi \vDash \varphi \to \psi \tag{62.13}$$

These are all useful and unproblematic inferences in mathematical reasoning. However, the philosophical and linguistic literature is replete with purported counterexamples to the equivalent inferences in non-mathematical contexts. These suggest that the material conditional $\to$ is not—or at least not always—the appropriate connective to use when symbolizing English "if ... then ..." statements.

---

content/counterfactuals/introduction/paradoxes-material.tex

62.2 Paradoxes of the Material Conditional

cnt:int:par: sec

One of the first to criticize the use of $\varphi \to \psi$ as a way to symbolize "if ... then ..." statements of English was C. I. Lewis. Lewis was criticizing the use of the material condition in Whitehead and Russell's *Principia Mathematica*, who pronounced $\to$ as "implies." Lewis rightly complained that if $\to$ meant "implies," then any false proposition p implies that p implies q, since $p \to (p \to q)$ is true if p is false, and that any true proposition q implies that p implies q, since $q \to (p \to q)$ is true if q is true.

Logicians of course know that *implication*, i.e., logical entailment, is not a connective but a relation between formulas or statements. So we should just not read $\to$ as "implies" to avoid confusion.[1] As long as we don't, the particular worry that Lewis had simply does not arise: p does not "imply" q even if we think of p as standing for a false English sentence. To determine if $p \vDash q$ we must consider *all* valuations, and $p \nvDash q$ even when we use p to symbolize a sentence which happens to be false.

But there is still something odd about "if ... then..." statements such as Lewis's

> If the moon is made of green cheese, then $2 + 2 = 4$.

and about the inferences

> The moon is not made of green cheese. Therefore, if the moon is made of green cheese, then $2 + 2 = 4$.
>
> $2 + 2 = 4$. Therefore, if the moon is made of green cheese, then $2 + 2 = 4$.

Yet, if "if ... then ..." were just $\to$, the sentence would be unproblematically true, and the inferences unproblematically valid.

Another example of concerns the tautology $(\varphi \to \psi) \vee (\psi \to \varphi)$. This would suggest that if you take two indicative sentences S and T from the newspaper at random, the sentence "If S then T, or if T then S" should be true.

---

content/counterfactuals/introduction/strict-conditional.tex

62.3 The Strict Conditional

cnt:int:str: sec

Lewis introduced the *strict conditional* $\varthickspace\neg\!\!\!\supset$ and argued that it, not the material conditional, corresponds to implication. In alethic modal logic, $\varphi \varthickspace\neg\!\!\!\supset \psi$ can be defined as $\Box(\varphi \to \psi)$. A strict conditional is thus true (at a world) iff the corresponding material conditional is necessary.

How does the strict conditional fare vis-a-vis the paradoxes of the material conditional? A strict conditional with a false antecedent and one with a true

[1] Reading "$\to$" as "implies" is still widely practised by mathematicians and computer scientists, although philosophers try to avoid the confusions Lewis highlighted by pronouncing it as "only if."

consequent, may be true, or it may be false. Moreover, $(\varphi \dashv\!\vDash \psi) \vee (\psi \dashv\!\vDash \varphi)$ is not valid. The strict conditional $\varphi \dashv\!\vDash \psi$ is also not equivalent to $\neg\varphi \vee \psi$, so it is not truth functional.

We have:

$$\varphi \dashv\!\vDash \psi \vDash \neg\varphi \vee \psi \text{ but:} \tag{62.14}$$
$$\neg\varphi \vee \psi \nvDash \varphi \dashv\!\vDash \psi \tag{62.15}$$
$$\psi \nvDash \varphi \dashv\!\vDash \psi \tag{62.16}$$
$$\neg\varphi \nvDash \varphi \dashv\!\vDash \psi \tag{62.17}$$
$$\neg(\varphi \to \psi) \nvDash \varphi \wedge \neg\psi \text{ but:} \tag{62.18}$$
$$\varphi \wedge \neg\psi \vDash \neg(\varphi \dashv\!\vDash \psi) \tag{62.19}$$

However, the strict conditional still supports modus ponens:

$$\varphi, \varphi \dashv\!\vDash \psi \vDash \psi \tag{62.20}$$

The strict conditional agglomerates:

$$\varphi \dashv\!\vDash \psi, \varphi \dashv\!\vDash \chi \vDash \varphi \dashv\!\vDash (\psi \wedge \chi) \tag{62.21}$$

Antecedent strengthening holds for the strict conditional:

$$\varphi \dashv\!\vDash \psi \vDash (\varphi \wedge \chi) \dashv\!\vDash \psi \tag{62.22}$$

The strict conditional is also transitive:

$$\varphi \dashv\!\vDash \psi, \psi \dashv\!\vDash \chi \vDash \varphi \dashv\!\vDash \chi \tag{62.23}$$

Finally, the strict conditional is equivalent to its contrapositive:

$$\varphi \dashv\!\vDash \psi \vDash \neg\psi \dashv\!\vDash \neg\varphi \tag{62.24}$$
$$\neg\psi \dashv\!\vDash \neg\varphi \vDash \varphi \dashv\!\vDash \psi \tag{62.25}$$

Problem 62.1. Give **S5**-counterexamples to the entailment relations which do not hold for the strict conditional, i.e., for:

1. $\neg p \nvDash \Box(p \to q)$
2. $q \nvDash \Box(p \to q)$
3. $\neg\Box(p \to q) \nvDash p \wedge \neg q$
4. $\nvDash \Box(p \to q) \vee \Box(q \to p)$

Problem 62.2. Show that the valid entailment relations hold for the strict conditional by giving **S5**-proofs of:

1. $\Box(\varphi \to \psi) \vDash \neg\varphi \vee \psi$
2. $\varphi \wedge \neg\psi \vDash \neg\Box(\varphi \to \psi)$
3. $\varphi, \Box(\varphi \to \psi) \vDash \psi$
4. $\Box(\varphi \to \psi), \Box(\varphi \to \chi) \vDash \Box(\varphi \to (\psi \wedge \chi))$
5. $\Box(\varphi \to \psi) \vDash \Box((\varphi \wedge \chi) \to \psi)$
6. $\Box(\varphi \to \psi), \Box(\psi \to \chi) \vDash \Box(\varphi \to \chi)$
7. $\Box(\varphi \to \psi) \vDash \Box(\neg\psi \to \neg\varphi)$
8. $\Box(\neg\psi \to \neg\varphi) \vDash \Box(\varphi \to \psi)$

However, the strict conditional still has its own "paradoxes." Just as a material conditional with a false antecedent or a true consequent is true, a strict conditional with a *necessarily* false antecedent or a necessarily true consequent is true. Moreover, any true strict conditional is necessarily true, and any false strict conditional is necessarily false. In other words, we have

$$\Box\neg\varphi \vDash \varphi \mathrel{-\!\!\!\vartriangleright} \psi \tag{62.26}$$
$$\Box\psi \vDash \varphi \mathrel{-\!\!\!\vartriangleright} \psi \tag{62.27}$$
$$\varphi \mathrel{-\!\!\!\vartriangleright} \psi \vDash \Box(\varphi \mathrel{-\!\!\!\vartriangleright} \psi) \tag{62.28}$$
$$\neg(\varphi \mathrel{-\!\!\!\vartriangleright} \psi) \vDash \Box\neg(\varphi \mathrel{-\!\!\!\vartriangleright} \psi) \tag{62.29}$$

These are not problems if you think of $\mathrel{-\!\!\!\vartriangleright}$ as "implies." Logical entailment relationships are, after all, mathematical facts and so can't be contingent. But they do raise issues if you want to use $\mathrel{-\!\!\!\vartriangleright}$ as a logical connective that is supposed to capture "if ... then ...," especially the last two. For surely there are "if ... then ..." statements that are contingently true or contingently false—in fact, they generally are neither necessary nor impossible.

Problem 62.3. Give proofs in **S5** of:

1. $\Box\neg\varphi \vDash \varphi \mathrel{-\!\!\!\vartriangleright} \psi$
2. $\varphi \mathrel{-\!\!\!\vartriangleright} \psi \vDash \Box(\varphi \mathrel{-\!\!\!\vartriangleright} \psi)$
3. $\neg(\varphi \mathrel{-\!\!\!\vartriangleright} \psi) \vDash \Box\neg(\varphi \mathrel{-\!\!\!\vartriangleright} \psi)$

Use the definition of $\mathrel{-\!\!\!\vartriangleright}$ to do so.

62.4 Counterfactuals

cnt:int:cnt:sec

A very common and important form of "if ... then ..." constructions in English are built using the past subjunctive form of *to be*: "if it were the case that ... then it would be the case that ..." Because usually the antecedent of such a conditional is false, i.e., counter to fact, they are called *counterfactual conditionals* (and because they use the subjunctive form of *to be*, also *subjunctive conditionals*. They are distinguished from *indicative* conditionals which take the form of "if it is the case that ... then it is the case that ..." Counterfactual and indicative conditionals differ in truth conditions. Consider Adams's famous example:

> If Oswald didn't kill Kennedy, someone else did.
>
> If Oswald hadn't killed Kennedy, someone else would have.

The first is indicative, the second counterfactual. The first is clearly true: we know President John F. Kennedy was killed by *someone*, and if that someone wasn't (contrary to the Warren Report) Lee Harvey Oswald, then someone else killed Kennedy. The second one says something different. It claims that if Oswald hadn't killed Kennedy, i.e., if the Dallas shooting had been avoided or had been unsuccessful, history would have subsequently unfolded in such a way that another assassination would have been successful. In order for it to be true, it would have to be the case that powerful forces had conspired to ensure JFK's death (as many JFK conspiracy theorists believe).

It is a live debate whether the *indicative* conditional is correctly captured by the material conditional, in particular, whether the paradoxes of the material conditional can be "explained" in a way that is compatible with it giving the truth conditions for English indicative conditionals. By contrast, it is uncontroversial that counterfactual conditionals cannot be symbolized correctly by the material conditionals. That is clear because, even though generally the antecedents of counterfactuals are false, not all counterfactuals with false antecedents are true—for instance, if you believe the Warren Report, and there was no conspiracy to assassinate JFK, then Adams's counterfactual conditional is an example.

Counterfactual conditionals play an important role in causal reasoning: a prime example of the use of counterfactuals is to express causal relationships. E.g., striking a match causes it to light, and you can express this by saying "if this match were struck, it would light." Material, and generally indicative conditionals, cannot be used to express this: "the match is struck $\rightarrow$ the match lights" is true if the match is never struck, regardless of what would happen if it were. Even worse, "the match is struck $\rightarrow$ the match turns into a bouquet of flowers" is also true if it is never struck, but the match would certainly not turn into a bouquet of flowers if it were struck.

It is still debated What exactly the correct logic of counterfactuals is. An influential analysis of counterfactuals was given by Stalnaker and Lewis. According to them, a counterfactual "if it were the case that S then it would be

the case that T" is true iff T is true in the counterfactual situation ("possible world") that is closest to the way the actual world is and where S is true. This is called an "ontic" analysis, since it makes reference to an ontology of possible worlds. Other analyses make use of conditional probabilities or theories of belief revision. There is a proliferation of different proposed logics of counterfactuals. There isn't even a single Lewis–Stalnaker logic of counterfactuals: even though Stalnaker and Lewis proposed accounts along similar lines with reference to closest possible worlds, the assumptions they made result in different valid inferences.

Chapter 63

Minimal Change Semantics

---

content/counterfactuals/minimal-change-semantics/introduction.tex

63.1 Introduction

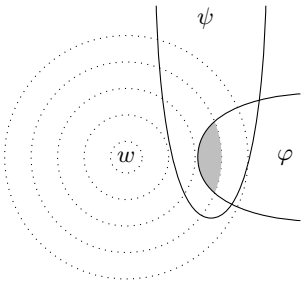
cnt:min:int:sec

Stalnaker and Lewis proposed accounts of counterfactual conditionals such as "If the match were struck, it would light." Their accounts were proposals for how to properly understand the truth conditions for such sentences. The idea behind both proposals is this: to evaluate whether a counterfactual conditional is true, we have to consider those possible worlds which are minimally different from the way the world actually is to make the antecedent true. If the consequent is true in these possible worlds, then the counterfactual is true. For instance, suppose I hold a match and a matchbook in my hand. In the actual world I only look at them and ponder what would happen if I were to strike the match. The minimal change from the actual world where I strike the match is that where I decide to act and strike the match. It is minimal in that nothing else changes: I don't also jump in the air, striking the match doesn't also light my hair on fire, I don't suddenly lose all strength in my fingers, I am not simultaneously doused with water in a SuperSoaker ambush, etc. In that alternative possibility, the match lights. Hence, it's true that if I were to strike the match, it would light.

This intuitive account can be paired with formal semantics for logics of counterfactuals. Lewis introduced the symbol "$\boxminus\!\!\rightarrow$" for the counterfactual

while Stalnaker used the symbol ">". We'll use $\boxrightarrow$, and add it as a binary connective to propositional logic. So, we have, in addition to formulas of the form $\varphi \to \psi$ also formulas of the form $\varphi \boxrightarrow \psi$. The formal semantics, like the relational semantics for modal logic, is based on models in which formulas are evaluated at worlds, and the satisfaction condition defining $\mathfrak{M}, w \Vdash \varphi \boxrightarrow \psi$ is given in terms of $\mathfrak{M}, w' \Vdash \varphi$ and $\mathfrak{M}, w' \Vdash \psi$ for some (other) worlds w'. Which w'? Intuitively, the one(s) closest to w for which it holds that $\mathfrak{M}, w' \Vdash \varphi$. This requires that a relation of "closeness" has to be included in the model as well.

Lewis introduced an instructive way of representing counterfactual situations graphically. Each possible world is at the center of a set of nested spheres containing other worlds—we draw these spheres as concentric circles. The worlds between two spheres are equally close to the world at the center as each other, those contained in a nested sphere are closer, and those in a surrounding sphere further away.



The closest φ-worlds are those worlds w' where φ is satisfied which lie in the smallest sphere around the center world w (the gray area). Intuitively, $\varphi \boxrightarrow \psi$ is satisfied at w if ψ is true at all closest φ-worlds.

---

content/counterfactuals/minimal-change-semantics/sphere-models.tex

63.2 Sphere Models

con:min:sph:sec

One way of providing a formal semantics for counterfactuals is to turn Lewis's informal account into a mathematical structure. The spheres around a world w then are sets of worlds. Since the spheres are nested, the sets of worlds around w have to be linearly ordered by the subset relation.

Definition 63.1. A *sphere model* is a triple $\mathfrak{M} = \langle W, O, V \rangle$ where W is a non-empty set of worlds, $V\colon \mathrm{At}_0 \to \wp(W)$ is a valuation, and $O\colon W \to \wp(\wp(W))$ assigns to each world w a *system of spheres* O_w. For each w, O_w is a set of sets of worlds, and must satisfy:

1. O_w is *centered* on w: $\{w\} \in O_w$.
2. O_w is *nested*: whenever S_1, $S_2 \in O_w$, $S_1 \subseteq S_2$ or $S_2 \subseteq S_1$, i.e., O_w is linearly ordered by $\subseteq$.

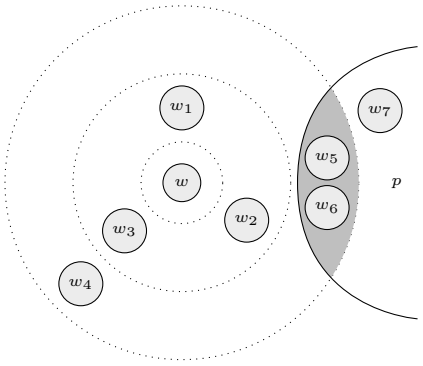


Figure 63.1: Diagram of a sphere model

con:min:sph:
fig:sphere-model

3. O_w is closed under non-empty unions.

4. O_w is closed under non-empty intersections.

The intuition behind O_w is that the worlds "around" w are stratified according to how far away they are from w. The innermost sphere is just w by itself, i.e., the set $\{w\}$: w is closer to w than the worlds in any other sphere. If $S \subsetneq S'$, then the worlds in $S' \setminus S$ are further way from w than the worlds in S: $S' \setminus S$ is the "layer" between the S and the worlds outside of S'. In particular, we have to think of the spheres as containing all the worlds within their outer surface; they are not just the individual layers.

The diagram in Figure 63.1 corresponds to the sphere model with $W = \{w, w_1, \ldots, w_7\}$, $V(p) = \{w_5, w_6, w_7\}$. The innermost sphere $S_1 = \{w\}$. The closest worlds to w are w_1, w_2, w_3, so the next larger sphere is $S_2 = \{w, w_1, w_2, w_3\}$. The worlds further out are w_4, w_5, w_6, so the outermost sphere is $S_3 = \{w, w_1, \ldots, w_6\}$. The system of spheres around w is $O_w = \{S_1, S_2, S_3\}$. The world w_7 is not in any sphere around w. The closest worlds in which p is true are w_5 and w_6, and so the smallest p-admitting sphere is S_3.

To define satisfaction of a formula φ at world w in a sphere model $\mathfrak{M}$, $\mathfrak{M}, w \Vdash \varphi$, we expand the definition for modal formulas to include a clause for $\psi \boxdot\!\!\rightarrow \chi$:

Definition 63.2. $\mathfrak{M}, w \Vdash \psi \boxdot\!\!\rightarrow \chi$ iff either

con:min:sph:
sphere-vac

1. For all $u \in \bigcup O_w$, $\mathfrak{M}, u \nVdash \psi$, or

con:min:sph:
sphere-nonvac

2. For some $S \in O_w$,

   a) $\mathfrak{M}, u \Vdash \psi$ for some $u \in S$, and

   b) for all $v \in S$, either $\mathfrak{M}, v \nVdash \psi$ or $\mathfrak{M}, v \Vdash \chi$.

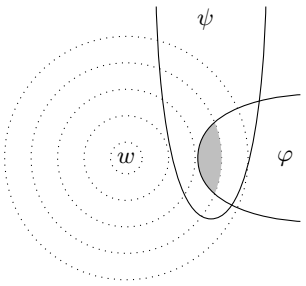


Figure 63.2: Non-vacuously true counterfactual

con:min:tf:fig:true

According to this definition, $\mathfrak{M}, w \Vdash \psi \boxdot\!\!\rightarrow \chi$ iff either the antecedent ψ is false everywhere in the spheres around w, or there is a sphere S where ψ is true, and the material conditional $\psi \rightarrow \chi$ is true at all worlds in that "ψ-admitting" sphere. Note that we didn't require in the definition that S is the *innermost* ψ-admitting sphere, contrary to what one might expect from the intuitive explanation. But if the condition in (2) is satisfied for some sphere S, then it is also satisfied for all spheres S contains, and hence in particular for the innermost sphere.

Note also that the definition of sphere models does not require that there *is* an innermost ψ-admitting sphere: we may have an infinite sequence $S_1 \supsetneq S_2 \supsetneq \cdots \supsetneq \{w\}$ of ψ-admitting spheres, and hence no innermost ψ-admitting spheres. In that case, $\mathfrak{M}, w \Vdash \psi \boxdot\!\!\rightarrow \chi$ iff $\psi \rightarrow \chi$ holds throughout the spheres S_i, S_{i+1}, ..., for some i.

---

content/counterfactuals/minimal-change-semantics/true-false.tex

63.3 Truth and Falsity of Counterfactuals

con:min:tf:sec

A counterfactual $\varphi \boxdot\!\!\rightarrow \psi$ is (non-vacuously) true if the closest φ-worlds are all ψ-worlds, as depicted in Figure 63.2. A counterfactual is also true at w if the system of spheres around w has no φ-admitting spheres at all. In that case it is *vacuously* true (see Figure 63.3).

It can be false in two ways. One way is if the closest φ-worlds are not all ψ-worlds, but some of them are. In this case, $\varphi \boxdot\!\!\rightarrow \neg\psi$ is also false (see Figure 63.4). If the closest φ-worlds do not overlap with the ψ-worlds at all, then $\varphi \boxdot\!\!\rightarrow \psi$ is false. But, in this case all the closest φ-worlds are $\neg\psi$-worlds, and so $\varphi \boxdot\!\!\rightarrow \neg\psi$ is true (see Figure 63.5).

In contrast to the strict conditional, counterfactuals may be contingent. Consider the sphere model in Figure 63.6. The φ-worlds closest to u are all ψ-worlds, so $\mathfrak{M}, u \Vdash \varphi \boxdot\!\!\rightarrow \psi$. But there are φ-worlds closest to v which are not ψ-worlds, so $\mathfrak{M}, v \nVdash \varphi \boxdot\!\!\rightarrow \psi$.

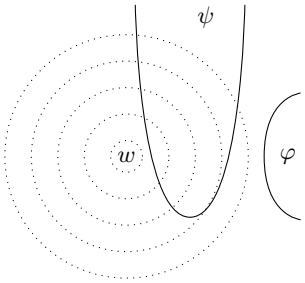


Figure 63.3: Vacuously true counterfactual

con:min:tf:
fig:vacuous

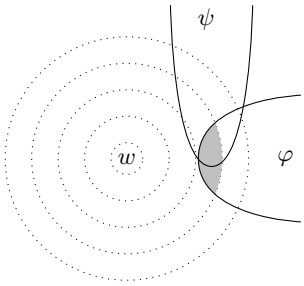


Figure 63.4: False counterfactual, false opposite

con:min:tf:
fig:false

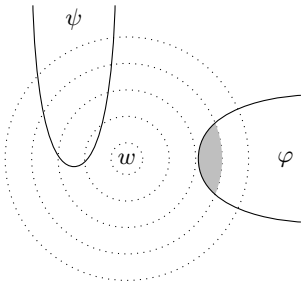


Figure 63.5: False counterfactual, true opposite

con:min:tf:
fig:false-opposite

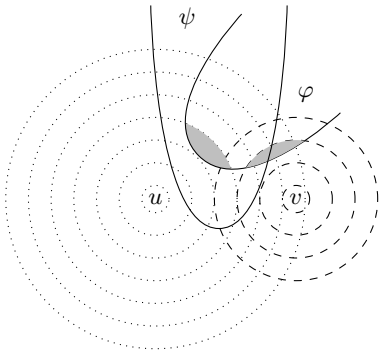


Figure 63.6: Contingent counterfactual

con:min:tf:
fig:contingent

content/counterfactuals/minimal-change-semantics/antecedent-strengthening.tex

63.4 Antecedent Strengthening

cnt:min:agg:
sec

"Strengthening the antecedent" refers to the inference $\varphi \to \chi \vDash (\varphi \wedge \psi) \to \chi$. It is valid for the material conditional, but invalid for counterfactuals. Suppose it is true that if I were to strike this match, it would light. (That means, there is nothing wrong with the match or the matchbook surface, I will not break the match, etc.) But it is not true that if I were to light this match in outer space, it would light. So the following inference is invalid:

> If the match were struck, it would light.
>
> Therefore, if the match were struck in outer space, it would light.

The Lewis–Stalnaker account of conditionals explains this: the closest world where I light the match and I do so in outer space is much further removed from the actual world than the closest world where I light the match is. So although it's true that the match lights in the latter, it is not in the former. And that is as it should be.

Example 63.3. The sphere semantics invalidates the inference, i.e., we have $p \boxdot\!\!\to r \nvDash (p \wedge q) \boxdot\!\!\to r$. Consider the model $\mathfrak{M} = \langle W, O, V\rangle$ where $W = \{w, w_1, w_2\}$, $O_w = \{\{w\}, \{w, w_1\}, \{w, w_1, w_2\}\}$, $V(p) = \{w_1, w_2\}$, $V(q) = \{w_2\}$, and $V(r) = \{w_1\}$. There is a p-admitting sphere $S = \{w, w_1\}$ and $p \to r$ is true at all worlds in it, so $\mathfrak{M}, w \Vdash p \boxdot\!\!\to r$. There is also a $(p \wedge q)$-admitting sphere $S' = \{w, w_1, w_2\}$ but $\mathfrak{M}, w_2 \nVdash (p \wedge q) \to r$, so $\mathfrak{M}, w \nVdash (p \wedge q) \boxdot\!\!\to r$ (see Figure 63.7).

content/counterfactuals/minimal-change-semantics/transitivity.tex

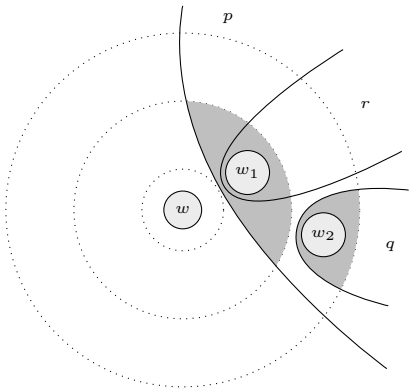


Figure 63.7: Counterexample to antecedent strengthening

cnt:min:agg:
fig:antecedent-strengthening

63.5 Transitivity

cnt:min:tra:
sec

For the material conditional, the chain rule holds: $\varphi \to \psi, \psi \to \chi \vDash \varphi \to \chi$. In other words, the material conditional is transitive. Is the same true for counterfactuals? Consider the following example due to Stalnaker.

> If J. Edgar Hoover had been born a Russian, he would have been a Communist.
>
> If J. Edgar Hoover were a Communist, he would have been be a traitor.
>
> Therefore, If J. Edgar Hoover had been born a Russian, he would have been be a traitor.

If Hoover had been born (at the same time he actually did), not in the United States, but in Russia, he would have grown up in the Soviet Union and become a Communist (let's assume). So the first premise is true. Likewise, the second premise, considered in isolation is true. The conclusion, however, is false: in all likelihood, Hoover would have been a fervent Communist if he had been born in the USSR, and not been a traitor (to his country). The intuitive assignment of truth values is borne out by the Stalnaker–Lewis account. The closest possible world to ours with the only change being Hoover's place of birth is the one where Hoover grows up to be a good citizen of the USSR. This is the closest possible world where the antecedent of the first premise and of the conclusion is true, and in that world Hoover is a loyal member of the Communist party, and so not a traitor. To evaluate the second premise, we have to look at a different world, however: the closest world where Hoover is a Communist, which is one where he was born in the United States, turned, and thus became a traitor.[1]

[1] Of course, to appreciate the force of the example we have to take on board some metaphysical and political assumptions, e.g., that it is possible that Hoover could have been born

Problem 63.1. Find a convincing, intuitive example for the failure of transitivity of counterfactuals.

Example 63.4. The sphere semantics invalidates the inference, i.e., we have $p \boxright q, q \boxright r \not\models p \boxright r$. Consider the model $\mathfrak{M} = \langle W, O, V\rangle$ where $W = \{w, w_1, w_2\}$, $O_w = \{\{w\}, \{w, w_1\}, \{w, w_1, w_2\}\}$, $V(p) = \{w_2\}$, $V(q) = \{w_1, w_2\}$, and $V(r) = \{w_1\}$. There is a p-admitting sphere $S = \{w, w_1, w_2\}$ and $p \to q$ is true at all worlds in it, so $\mathfrak{M}, w \Vdash p \boxright q$. There is also a q-admitting sphere $S' = \{w, w_1\}$ and $\mathfrak{M} \nVdash q \to r$ is true at all worlds in it, so $\mathfrak{M}, w \Vdash q \boxright r$. However, the p-admitting sphere $\{w, w_1, w_2\}$ contains a world, namely w_2, where $\mathfrak{M}, w_2 \nVdash p \to r$. cnt:min:tra:ex:trans-counterex

Problem 63.2. Draw the sphere diagram corresponding to the counterexample in Example 63.4.

Problem 63.3. In Example 63.4, world w_2 is where Hoover is born in Russia, is a communist, and not a traitor, and w_1 is the world where Hoover is born in the US, is a communist, and a traitor. In this model, w_1 is closer to w than w_2 is. Is this necessary? Can you give a counterexample that does not assume that Hoover's being born in Russia is a more remote possibility than him being a Communist?

---

content/counterfactuals/minimal-change-semantics/contraposition.tex

63.6 Contraposition

Material and strict conditionals are equivalent to their contrapositives. Counterfactuals are not. Here is an example due to Kratzer: cnt:min:cpo:sec

> If Goethe hadn't died in 1832, he would (still) be dead now.
>
> If Goethe weren't dead now, he would have died in 1832.

The first sentence is true: humans don't live hundreds of years. The second is clearly false: if Goethe weren't dead now, he would be still alive, and so couldn't have died in 1832.

Example 63.5. The sphere semantics invalidates contraposition, i.e., we have $p \boxright q \not\models \neg q \boxright \neg p$. Think of p as "Goethe didn't die in 1832" and q as "Goethe is dead now." We can capture this in a model $\mathfrak{M}_1 = \langle W, O, V\rangle$ with $W = \{w, w_1, w_2\}$, $O = \{\{w\}, \{w, w_1\}, \{w, w_1, w_2\}\}$, $V(p) = \{w_1, w_2\}$ and $V(q) = \{w, w_1\}$. So w is the actual world where Goethe died in 1832 and is still dead; w_1 is the (close) world where Goethe died in, say, 1833, and is still dead; and w_2 is a (remote) world where Goethe is still alive. There is a p-admitting sphere $S = \{w, w_1\}$ and $p \to q$ is true at all worlds in it, so $\mathfrak{M}, w \Vdash p \boxright q$. cnt:min:cpo:ex:contraposition-counterex

to Russian parents, or that Communists in the US of the 1950s were traitors to their country.

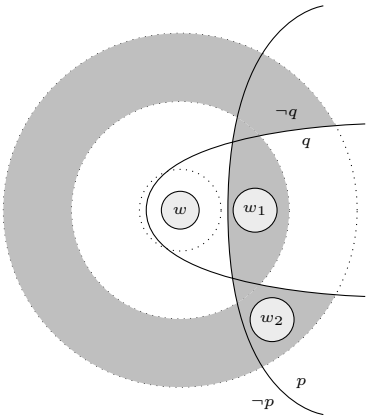


Figure 63.8: Counterexample to contraposition

cnt:min:cpo:
fig:contraposition

However, the $\neg q$-admitting sphere $\{w, w_1, w_2\}$ contains a world, namely w_2, where q is false and p is true, so $\mathfrak{M}, w_2 \nVdash \neg q \to \neg p$.

Part XV

Set Theory

Chapter 64

The Iterative Conception

---

content/set-theory/story/extensionality.tex

64.1 Extensionality

sth:story:extensionality:sec

The very first thing to say is that sets are individuated by their elements. More precisely:

Axiom (Extensionality). If sets A and B have the same elements, then A and B are the same set.

$$\forall A \forall B \left(\forall x \left(x \in A \leftrightarrow x \in B\right) \rightarrow A = B\right)$$

We assumed this throughout part I. But it bears repeating. The Axiom of Extensionality expresses the basic idea that a set is determined by its elements. (So sets might be contrasted with *concepts*, where precisely the same objects might fall under many different concepts.)

Why embrace this principle? Well, it is plausible to say that any denial of Extensionality is a decision to abandon anything which might even be called *set theory*. Set theory is no more nor less than the theory of extensional collections.

The real challenge in part XV, though, is to lay down principles which tell us *which sets exist*. And it turns out that the only truly "obvious" answer to this question is provably wrong.

---

content/set-theory/story/russells-paradox-again.tex

64.2 Russell's Paradox (again)

sth:story:rus:sec

In part I, we worked with a naïve set theory. But according to a *very* naïve conception, sets are just the extensions of predicates. This naïve thought would mandate the following principle:

> *Naïve Comprehension.* $\{x : \varphi(x)\}$ exists for any formula φ.

Tempting as this principle is, it is provably inconsistent. We saw this in section 1.6, but the result is so important, and so straightforward, that it's worth repeating. Verbatim.

Theorem 64.1 (Russell's Paradox). *There is no set* $R = \{x : x \notin x\}$

Proof. If $R = \{x : x \notin x\}$ exists, then $R \in R$ iff $R \notin R$, which is a contradiction. □

Russell discovered this result in June 1901. (He did not, though, put the paradox in quite the form we just presented it, since he was considering Frege's set theory, as outlined in *Grundgesetze*. We will return to this in section 64.6.) Russell wrote to Frege on June 16, 1902, explaining the inconsistency in Frege's system. For the correspondence, and a bit of background, see Heijenoort (1967, pp. 124–8).

It is worth emphasising that this two-line proof is a result of *pure logic*. Granted, we implicitly used a (non-logical?) axiom, Extensionality, in our notation $\{x : x \notin x\}$; for $\{x : \varphi(x)\}$ is to be *the unique* (by Extensionality) set of the φs, if one exists. But we can avoid even the hint of Extensionality, just by stating the result as follows: *there is no set whose members are exactly the non-self-membered sets*. And this has nothing much to do with sets. As Russell himself observed, exactly similar reasoning will lead you to conclude: *no man shaves exactly the men who do not shave themselves*. Or: *no pug sniffs exactly the pugs which don't sniff themselves*. And so on. Schematically, the shape of the result is just:

$$\neg\exists x\forall z(Rzx \leftrightarrow \neg Rzz).$$

And that's just a theorem (scheme) of first-order logic. Consequently, we can't avoid Russell's Paradox just by tinkering with our set theory; it arises before we even *get* to set theory. If we're going to use (classical) first-order logic, we simply have to *accept* that there is no set $R = \{x : x \notin x\}$.

The upshot is this. If you want to accept Naïve Comprehension whilst *avoiding* inconsistency, you cannot just tinker with the *set theory*. Instead, you would have to overhaul your *logic*.

Of course, set theories with non-classical logics have been presented. But they are—to say the least—non-standard. The standard approach to Russell's Paradox is to treat it as a straightforward non-existence proof, and then to try to learn how to live with it. That is the approach we will follow.

64.3 Predicative and Impredicative

sth:story:predicative:sec

The Russell set, R, was defined via $\{x : x \notin x\}$. Spelled out more fully, R would be the set which contains all and only those sets which are not non-self-membered. So in defining R, we quantify over the domain which would contain R (if it existed).

This is an *impredicative* definition. More generally, we might say that a definition is impredicative iff it quantifies over a domain which contains the object that is being defined.

In the wake of the paradoxes, Whitehead, Russell, Poincaré and Weyl rejected such impredicative definitions as "viciously circular":

> An analysis of the paradoxes to be avoided shows that they all result from a kind of vicious circle. The vicious circles in question arise from supposing that a collection of objects may contain members which can only be defined by means of the collection as a whole[. . . . ¶]
>
> The principle which enables us to avoid illegitimate totalities may be stated as follows: 'Whatever involves *all* of a collection must not be one of the collection'; or, conversely: 'If, provided a certain collection had a total, it would have members only definable in terms of that total, then the said collection has no total.' We shall call this the 'vicious-circle principle,' because it enables us to avoid the vicious circles involved in the assumption of illegitimate totalities. (Whitehead and Russell, 1910, p. 37)

If we follow them in rejecting the *vicious-circle principle*, then we might attempt to replace the disastrous Naïve Comprehension Scheme (of section 64.2) with something like this:

> *Predicative Comprehension.* For every formula φ quantifying only over sets: the set′ $\{x : \varphi(x)\}$ exists.

So long as sets′ are not sets, no contradiction will ensue.

Unfortunately, Predicative Comprehension is not very *comprehensive*. After all, it introduces us to new entities, sets′. So we will have to consider formulas which quantify over sets′. If they always yield a set′, then Russell's paradox will arise again, just by considering the set′ of all non-self-membered sets′. So, pursuing the same thought, we must say that a formula quantifying over sets′ yields a corresponding set″. And then we will need sets‴, sets⁗, etc. To prevent a rash of primes, it will be easier to think of these as sets_0, sets_1, sets_2, sets_3, sets_4,. . . . And this would give us a way into the (simple) theory of types.

There are a few obvious objections against such a theory (though it is not obvious that they are *overwhelming* objections). In brief: the resulting theory is cumbersome to use; it is profligate in postulating different kinds of objects;

and it is not clear, in the end, that impredicative definitions are even *all that bad*.

To bring out the last point, consider this remark from Ramsey:

> we may refer to a man as the tallest in a group, thus identifying him by means of a totality of which he is himself a member without there being any vicious circle. (Ramsey, 1925)

Ramsey's point is that "the tallest man in the group" *is* an impredicative definition; but it is obviously perfectly kosher.

One might respond that, in this case, we could pick out the tallest person by *predicative* means. For example, maybe we could just point at the man in question. The objection against impredicative definitions, then, would clearly need to be limited to entities which can *only* be picked out impredicatively. But even then, we would need to hear more, about why such "essential impredicativity" would be so bad.[1]

Admittedly, impredicative definitions are extremely bad news, if we want our definitions to provide us with something like a recipe for *creating* an object. For, given an impredicative definition, one would genuinely be caught in a vicious circle: to create the impredicatively specified object, one would *first* need to create all the objects (including the impredicatively specified object), since the impredicatively specified object is specified in terms of all the objects; so one would need to create the impredicatively specified object before one had created it itself. But again, this is only a serious objection against "essentially impredicatively" specified sets, if we think of sets as things that we *create*. And we (probably) don't.

As such—for better or worse—the approach which became common does not involve taking a hard line concerning (im)predicativity. Rather, it involves what is now regarded as the cumulative-iterative approach. In the end, this will allow us to stratify our sets into "stages"—a *bit* like the predicative approach stratifies entities into sets_0, sets_1, sets_2, ...—but we will not postulate any difference in kind between them.

---

content/set-theory/story/cumulative-approach.tex

64.4 The Cumulative-Iterative Approach

sth:story:approach:sec

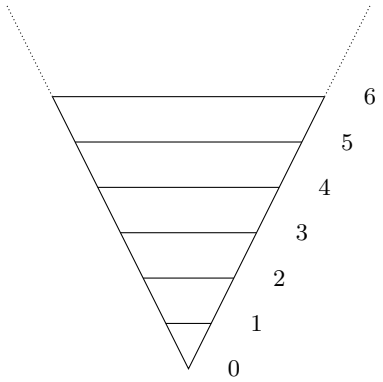
Here is a slightly fuller statement of how we will stratify sets into stages:

> Sets are formed in *stages*. For each stage S, there are certain stages which are *before* S. At stage S, each collection consisting of sets formed at stages before S is formed into a set. There are no sets other than the sets which are formed at stages. (Shoenfield, 1977, p. 323)

[1] For more, see Linnebo (2010).

This is a sketch of the *cumulative-iterative conception of set*. It will underpin the formal set theory that we present in part XV.

Let's explore this in a little more detail. As Shoenfield describes the process, at every stage, we form new sets from the sets which were available to us from earlier stages. So, on Shoenfield's picture, at the initial stage, stage 0, there are no *earlier* stages, and so *a fortiori* there are no sets available to us from earlier stages.[2] So we form only one set: the set with no elements $\emptyset$. At stage 1, exactly one set is available to us from earlier stages, so only one new set is $\{\emptyset\}$. At stage 2, two sets are available to us from earlier stages, and we form two new sets $\{\{\emptyset\}\}$ and $\{\emptyset, \{\emptyset\}\}$. At stage 3, four sets are available to us from earlier stages, so we form twelve new sets.... As such, the cumulative-iterative picture of the sets will look a bit like this (with numbers indicating stages):



So: why should we embrace this story?

One reason is that it is a nice, tractable story. Given the demise of the most obvious story, i.e., Naïve Comprehension, we are in want of something nice.

But the story is not *just* nice. We have a good reason to believe that any set theory based on this story will be *consistent*. Here is why.

Given the cumulative-iterative conception of set, we form sets at stages; and their elements must be objects which were available *already*. So, for any stage S, we can form the set

$$R_S = \{x : x \notin x \text{ and } x \text{ was available before } S\}$$

The reasoning involved in proving Russell's Paradox will now establish that R_S itself is not available before stage S. And that's not a contradiction. Moreover, if we embrace the cumulative-iterative conception of set, then we shouldn't even have *expected* to be able to form the Russell set itself. For that would be the set of all non-self-membered sets that "will ever be available". In short: the fact that we (provably) can't form the Russell set isn't *surprising*, given the cumulative-iterative story; it's what we would *predict*.

[2] Why should we assume that there *is* a first stage? See the footnote to *Stages-are-ordered* in section 65.1.

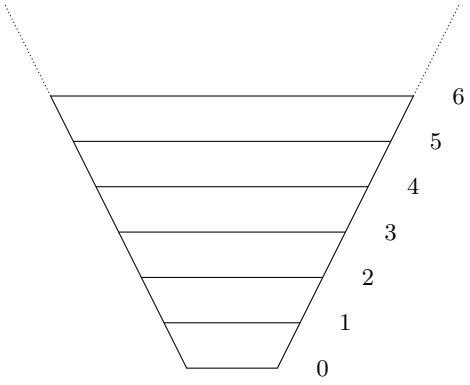
content/set-theory/story/urelements.tex

64.5 Urelements or Not?

sth:story:urelements:sec

In the next few chapters, we will try to extract axioms from the cumulative-iterative conception of set. But, before going any further, we need to say something more about *urelements*.

The picture of section 64.4 allowed us only to form new sets from old *sets*. However, we might want to allow that certain *non-sets*—cows, pigs, grains of sand, or whatever—can be elements of sets. In that case, we would start with certain basic elements, *urelements*, and then say that at each stage S we would form "all possible" sets consisting of urelements or sets formed at stages before S (in any combination). The resulting picture would look more like this:



So now we have a decision to take: *Should we allow urelements?*

Philosophically, it makes sense to include urelements in our theorising. The main reason for this is to make our set theory *applicable*. To illustrate the point, recall from chapter 4 that we say that two sets A and B have the same size, i.e., $A \approx B$, iff there is a bijection between them. Now, if the cows in the field and the pigs in the sty both form sets, we can offer a set-theoretical treatment of the claim "there are as many cows as pigs". But if we ban urelements, so that the cows and the pigs do *not* form sets, then that set-theoretical treatment will be unavailable. Indeed, we will have no straightforward ability to apply set theory to anything other than sets themselves. (For more reasons to include urelements, see Potter 2004, pp. vi, 24, 50–1.)

Mathematically, however, it is quite rare to allow urelements. In part, this is because it is *very slightly* easier to formulate set theory without urelements. But, occasionally, one finds more interesting justifications for excluding urelement from set theory:

> In accordance with the belief that set theory is the foundation of mathematics, we should be able to capture all of mathematics by

> just talking about sets, so our variable should not range over objects like cows and pigs. (Kunen, 1980, p. 8)

So: a focus on applicability would suggest *including* urelements; a focus on a reductive foundational goal (reducing mathematics to pure set theory) might suggest *excluding* them. Mild laziness, too, points in the direction of excluding urelements.

We will follow the laziest path. Partly, though, there is a pedagogical justification. Our aim is to introduce you to the elements of set theory that you would need in order to get started on the philosophy of set theory. And most of that philosophical literature discusses set theories formulated *without* urelements. So this book will, perhaps, be of more use, if it hews fairly closely to that literature.

---

content/set-theory/story/grundgesetze.tex

64.6 Appendix: Frege's Basic Law V

sth:story:blv:sec

In section 64.2, we explained that Russell's formulated his paradox as a problem for the system Frege outlined in his *Grundgesetze*. Frege's system did not include a direct formulation of Naïve Comprehension. So, in this appendix, we will very briefly explain what Frege's system *did* include, and how it relates to Naïve Comprehension and how it relates to Russell's Paradox.

Frege's system is *second-order*, and was designed to formulate the notion of an *extension of a concept*.[3] Using notation inspired by Frege, we will write $\epsilon x\, F(x)$ for *the extension of the concept* F. This is a device which takes a *predicate*, "F", and turns it into a (first-order) *term*, "$\epsilon x\, F(x)$". Using this device, Frege offered the following *definition* of membership:

$$a \in b =_{\text{df}} \exists G(b = \epsilon x\, G(x) \wedge Ga)$$

roughly: $a \in b$ iff a falls under a concept whose extension is b. (Note that the quantifier "$\exists G$" is second-order.) Frege also maintained the following principle, known as *Basic Law V*:

$$\epsilon x\, F(x) = \epsilon x\, G(x) \leftrightarrow \forall x(Fx \leftrightarrow Gx)$$

roughly: concepts have identical extensions iff they are coextensive. (Again, both "F" and "G" are in predicate position.) Now a simple principle connects membership with property-satisfaction:

Lemma 64.2 (in *Grundgesetze*). $\forall F \forall a(a \in \epsilon x\, F(x) \leftrightarrow Fa)$

sth:story:blv:lem:Fregeextensions

[3] Strictly speaking, Frege attempts to formalize a more general notion: the "value-range" of a function. Extensions of concepts are a special case of the more general notion. See Heck (2012, pp. 8–9) for the details.

Proof. Fix F and a. Now $a \in \epsilon x\, F(x)$ iff $\exists G(\epsilon x\, F(x) = \epsilon x\, G(x) \wedge Ga)$ (by the definition of membership) iff $\exists G(\forall x(Fx \leftrightarrow Gx) \wedge Ga)$ (by Basic Law V) iff Fa (by elementary second-order logic). □

And this yields Naïve Comprehension almost immediately:

Lemma 64.3 (in *Grundgesetze*.). $\forall F \exists s \forall a(a \in s \leftrightarrow Fa)$

Proof. Fix F; now Lemma 64.2 yields $\forall a(a \in \epsilon x\, F(x) \leftrightarrow Fa)$; so $\exists s \forall a(a \in s \leftrightarrow Fa)$ by existential generalisation. The result follows since F was arbitrary. □

Russell's Paradox follows by taking F as given by $\forall x(Fx \leftrightarrow x \notin x)$.

Chapter 65

Steps towards Z

---

content/set-theory/z/story.tex

65.1 The Story in More Detail

sth:z:story:sec

In section 64.4, we quoted Schoenfield's description of the process of set-formation. We now want to write down a few more principles, to make this story a bit more precise. Here they are:

> *Stages-are-key.* Every set is formed at some stage.
>
> *Stages-are-ordered.* Stages are ordered: some come *before* others.[1]
>
> *Stages-accumulate.* For any stage S, and for any sets which were formed *before* stage S: a set is formed at stage S whose members are exactly those sets. Nothing else is formed at stage S.

[1] We will actually assume—tacitly—that the stages are *well-ordered*. What this amounts to is explained in chapter 66. This is a substantial assumption. In fact, using a very clever technique due to Scott (1974), this assumption can be *avoided* and then *derived*. (This will also explain why we should think that there is an initial stage.) We cannot go into that here; for more, see Button (2021).

These are informal principles, but we will be able to use them to vindicate several of the axioms of Zermelo's set theory.

(We should offer a word of caution. Although we will be presenting some completely standard axioms, with completely standard names, the italicized principles we have just presented have no particular names in the literature. We simply monikers which we hope are helpful.)

---

content/set-theory/z/separation.tex

65.2 Separation

We start with a principle to replace Naïve Comprehension: sth:z:sep:sec

Axiom (Scheme of Separation). For every formula $\varphi(x)$, this is an axiom: for any A, the set $\{x \in A : \varphi(x)\}$ exists.

Note that this is not a single axiom. It is a *scheme* of axioms. There are *infinitely many* Separation axioms; one for every formula $\varphi(x)$. The scheme can equally well be (and normally is) written down as follows:

> For any formula $\varphi(x)$ which does not contain "S", this is an axiom:
>
> $$\forall A \exists S \forall x (x \in S \leftrightarrow (\varphi(x) \wedge x \in A)).$$

In keeping with the convention noted at the start of part XV, the formulas φ in the Separation axioms may have parameters.[2]

Separation is immediately justified by our cumulative-iterative conception of sets we have been telling. To see why, let A be a set. So A is formed by some stage S (by *Stages-are-key*). Since A was formed at stage S, all of A's members were formed before stage S (by *Stages-accumulate*). Now in particular, consider all the sets which are members of A and which also satisfy φ; clearly all of these sets, too, were formed before stage S. So they are formed into a set $\{x \in A : \varphi(x)\}$ at stage S too (by *Stages-accumulate*).

Unlike Naïve Comprehension, this avoid Russell's Paradox. For we cannot simply assert the existence of the set $\{x : x \notin x\}$. Rather, *given* some set A, we can assert the existence of the set $R_A = \{x \in A : x \notin x\}$. But all this proves is that $R_A \notin R_A$ and $R_A \notin A$, none of which is very worrying.

However, Separation has an immediate and striking consequence:

Theorem 65.1. *There is no* universal *set, i.e.,* $\{x : x = x\}$ *does not exist.* sth:z:sep:thm:NoUniversalSet

Proof. For reductio, suppose V is a universal set. Then by Separation, $R = \{x \in V : x \notin x\} = \{x : x \notin x\}$ exists, contradicting Russell's Paradox. □

[2] For an explanation of what this means, see the discussion immediately after Corollary 6.7.

The absence of a universal set—indeed, the open-endedness of the hierarchy of sets—is one of the most fundamental ideas behind the cumulative-iterative conception. So it is worth seeing that, intuitively, we could reach it via a different route. A universal set must be an element of itself. But, on our cumulative-iterative conception, every set appears (for the first time) in the hierarchy at the first stage immediately after all of its elements. But this entails that *no* set is self-membered. For any self-membered set would have to first occur immediately after the stage at which it first occurred, which is absurd. (We will see in Definition 67.15 how to make this explanation more rigorous, by using the notion of the "rank" of a set. However, we will need to have a few more axioms in place to do this.)

Here are a few more consequences of Separation and Extensionality.

sth:z:sep: prop:emptyexists **Proposition 65.2.** *If any set exists, then $\emptyset$ exists.*

Proof. If A is a set, $\emptyset = \{x \in A : x \neq x\}$ exists by Separation. □

Proposition 65.3. *$A \setminus B$ exists for any sets A and B*

Proof. $A \setminus B = \{x \in A : x \notin B\}$ exists by Separation. □

It also turns out that (almost) arbitrary intersections exist:

sth:z:sep: prop:intersectionsexist **Proposition 65.4.** *If $A \neq \emptyset$, then $\bigcap A = \{x : (\forall y \in A) x \in y\}$ exists.*

Proof. Let $A \neq \emptyset$, so there is some $c \in A$. Then $\bigcap A = \{x : (\forall y \in A) x \in y\} = \{x \in c : (\forall y \in A) x \in y\}$, which exists by Separation. □

Note the condition that $A \neq \emptyset$, though; for $\bigcap \emptyset$ would be the universal set, vacuously, contradicting Theorem 65.1.

---

content/set-theory/z/union.tex

65.3 Union

sth:z:union: sec Proposition 65.4 gave us intersections. But if we want arbitrary unions to exist, we need to lay down another axiom:

Axiom (Union). For any set A, the set $\bigcup A = \{x : (\exists b \in A) x \in b\}$ exists.

$$\forall A \exists U \forall x (x \in U \leftrightarrow (\exists b \in A) x \in b)$$

This axiom is also justified by the cumulative-iterative conception. Let A be a set, so A is formed at some stage S (by *Stages-are-key*). Every member of A was formed *before* S (by *Stages-accumulate*); so, reasoning similarly, every member of every member of A was formed before S. Thus all of *those* sets are available before S, to be formed into a set at S. And that set is just $\bigcup A$.

---

content/set-theory/z/pairs.tex

65.4 Pairs

The next axiom to consider is the following: sth:z:pairs:sec

Axiom (Pairs). For any sets a, b, the set $\{a, b\}$ exists.

$$\forall a \forall b \exists P \forall x (x \in P \leftrightarrow (x = a \vee x = b))$$

Here is how to justify this axiom, using the iterative conception. Suppose a is available at stage S, and b is available at stage T. Let M be whichever of stages S and T comes later. Then since a and b are both available at stage M, the set $\{a, b\}$ is a possible collection available at any stage after M (whichever is the greater).

But hold on! Why assume that there *are* any stages after M? If there are none, then our justification will fail. So, to justify Pairs, we will have to add another principle to the story we told in section 65.1, namely:

> *Stages-keep-going.* There is no last stage.

Is this principle justified? Nothing in Shoenfield's story stated *explicitly* that there is no last stage. Still, even if it is (strictly speaking) an extra addition to our story, it fits well with the basic idea that sets are formed in stages. We will simply accept it in what follows. And so, we will accept the Axiom of Pairs too.

Armed with this new Axiom, we can prove the existence of plenty more sets. For example:

Proposition 65.5. *For any sets a and b, the following sets exist:* sth:z:pairs:prop:pairsconsequences

1. $\{a\}$ sth:z:pairs:singleton
2. $a \cup b$ sth:z:pairs:binunion
3. $\langle a, b\rangle$ sth:z:pairs:tuples

Proof. (1). By Pairs, $\{a, a\}$ exists, which is $\{a\}$ by Extensionality.

(2). By Pairs, $\{a, b\}$ exists. Now $a \cup b = \bigcup\{a, b\}$ exists by Union.

(3). By (1), $\{a\}$ exists. By Pairs, $\{a, b\}$ exists. Now $\{\{a\}, \{a, b\}\} = \langle a, b\rangle$ exists, by Pairs again. □

Problem 65.1. Show that, for any sets a, b, c, the set $\{a, b, c\}$ exists.

Problem 65.2. Show that, for any sets $a_1, \ldots, a_n$, the set $\{a_1, \ldots, a_n\}$ exists.

65.5 Powersets

sth:z:power: sec

We will proceed with another axiom:

Axiom (Powersets). For any set A, the set $\wp(A) = \{x : x \subseteq A\}$ exists.

$$\forall A \exists P \forall x (x \in P \leftrightarrow (\forall z \in x) z \in A)$$

Our justification for this is pretty straightforward. Suppose A is formed at stage S. Then all of A's members were available before S (by *Stages-accumulate*). So, reasoning as in our justification for Separation, every subset of A is formed by stage S. So they are all available, to be formed into a single set, at any stage after S. And we know that there is some such stage, since S is not the last stage (by *Stages-keep-going*). So $\wp(A)$ exists.

Here is a nice consequence of Powersets:

Proposition 65.6. *Given any sets A, B, their Cartesian product $A \times B$ exists.*

Proof. The set $\wp(\wp(A \cup B))$ exists by Powersets and Proposition 65.5. So by Separation, this set exists:

$$C = \{z \in \wp(\wp(A \cup B)) : (\exists x \in A)(\exists y \in B) z = \langle x, y \rangle\}.$$

Now, for any $x \in A$ and $y \in B$, the set $\langle x, y \rangle$ exists by Proposition 65.5. Moreover, since $x, y \in A \cup B$, we have that $\{x\}, \{x, y\} \in \wp(A \cup B)$, and $\langle x, y \rangle \in \wp(\wp(A \cup B))$. So $A \times B = C$. □

In this proof, Powerset interacts with Separation. And that is no surprise. Without Separation, Powersets wouldn't be a very *powerful* principle. After all, Separation tells us which subsets of a set exist, and hence determines just how "fat" each Powerset is.

Problem 65.3. Show that, for any sets A, B: (i) the set of all relations with domain A and range B exists; and (ii) the set of all functions from A to B exists.

Problem 65.4. Let A be a set, and let $\sim$ be an equivalence relation on A. Prove that the set of equivalence classes under $\sim$ on A, i.e., $A/_\sim$, exists.

---

content/set-theory/z/infinity-again.tex

65.6 Infinity

sth:z:infinity-again: sec

We already have enough axioms to ensure that there are infinitely many sets (if there are any). For suppose some set exists, and so $\emptyset$ exists (by Proposition 65.2). Now for any set x, the set $x \cup \{x\}$ exists by Proposition 65.5. So, applying this a few times, we will get sets as follows:

0. $\emptyset$
1. $\{\emptyset\}$
2. $\{\emptyset, \{\emptyset\}\}$
3. $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$
4. $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\}$

and we can check that each of these sets is distinct.

We have started the numbering from 0, for a few reasons. But one of them is this. It is not that hard to check that the set we have labelled "n" has exactly n members, and (intuitively) is formed at the nth stage.

But. This gives us *infinitely many* sets, but it does not guarantee that there is an *infinite set*, i.e., a set with infinitely many members. And this really matters: unless we can find a (Dedekind) infinite set, we cannot construct a Dedekind algebra. But we want a Dedekind algebra, so that we can treat it as the set of natural numbers. (Compare section 6.4.)

Importantly, the axioms we have laid down so far do *not* guarantee the existence of any infinite set. So we have to lay down a new axiom:

Axiom (Infinity). There is a set, I, such that $\emptyset \in I$ and $x \cup \{x\} \in I$ whenever $x \in I$.

$$\begin{aligned}&\exists I((\exists o \in I)\forall x\ x \notin o \wedge \\ &\quad (\forall x \in I)(\exists s \in I)\forall z(z \in s \leftrightarrow (z \in x \vee z = x)))\end{aligned}$$

It is easy to see that the set I given to us by the Axiom of Infinity is Dedekind infinite. Its distinguished element is $\emptyset$, and the injection on I is given by $s(x) = x \cup \{x\}$. Now, Theorem 6.5 showed how to extract a Dedekind Algebra from a Dedekind infinite set; and we will treat this as our set of natural numbers. More precisely:

Definition 65.7. Let I be any set given to us by the Axiom of Infinity. Let s be the function $s(x) = x \cup \{x\}$. Let $\omega = \mathrm{clo}_s(\emptyset)$. We call the members of ω the *natural numbers*, and say that n is the result of n-many applications of s to $\emptyset$. sth:z:infinity-again: defnomega

You can now look back and check that the set labelled "n", a few paragraphs earlier, will be treated *as* the number n.

We will discuss this significance of this stipulation in section 65.8. For now, it enables us to prove an intuitive result:

Proposition 65.8. *No natural number is Dedekind infinite.* sth:z:infinity-again: naturalnumbersarentinfinite

Proof. The proof is by induction, i.e., Theorem 6.6. Clearly $0 = \emptyset$ is not Dedekind infinite. For the induction step, we will establish the contrapositive: if (absurdly) $s(n)$ is Dedekind infinite, then n is Dedekind infinite.

So suppose that $s(n)$ is Dedekind infinite, i.e., there is some injection f with $\text{ran}(f) \subsetneq \text{dom}(f) = s(n) = n \cup \{n\}$. There are two cases to consider.

Case 1: $n \notin \text{ran}(f)$. So $\text{ran}(f) \subseteq n$, and $f(n) \in n$. Let $g = f\restriction_n$; now $\text{ran}(g) = \text{ran}(f) \setminus \{f(n)\} \subsetneq n = \text{dom}(g)$. Hence n is Dedekind infinite.

Case 2: $n \in \text{ran}(f)$. Fix $m \in \text{dom}(f) \setminus \text{ran}(f)$, and define a function h with domain $s(n) = n \cup \{n\}$:

$$h(x) = \begin{cases} f(x) & \text{if } f(x) \neq n \\ m & \text{if } f(x) = n \end{cases}$$

So h and f agree everywhere, except that $h(f^{-1}(n)) = m \neq n = f(f^{-1}(n))$. Since f is an injection, $n \notin \text{ran}(h)$; and $\text{ran}(h) \subsetneq \text{dom}(h) = s(n)$. Now n is Dedekind infinite, using the argument of Case 1. □

The question remains, though, of how we might *justify* the Axiom of Infinity. The short answer is that we will need to add another principle to the story we have been telling. That principle is as follows:

> *Stages-hit-infinity.* There is an infinite stage. That is, there is a stage which (a) is not the first stage, and which (b) has some stages before it, but which (c) has no immediate predecessor.

The Axiom of Infinity follows straightforwardly from this principle. We know that natural number n is formed at stage n. So the set ω is formed at the first infinite stage. And ω itself witnesses the Axiom of Infinity.

This, however, simply pushes us back to the question of how we might justify *Stages-hit-infinity*. As with *Stages-keep-going*, it was not an explicit part of the story we told about the cumulative-iterative hierarchy. But more than that: nothing in the very idea of an iterative hierarchy, in which sets are formed stage by stage, forces us to think that the process involves an *infinite* stage. It seems perfectly coherent to think that the stages are ordered like the natural numbers.

This, however, gives rise to an obvious problem. In section 6.4, we considered Dedekind's "proof" that there is a Dedekind infinite set (of thoughts). This may not have struck you as very satisfying. But if *Stages-hit-infinity* is not "forced upon us" by the iterative conception of set (or by "the laws of thought"), then we are still left without an intrinsic justification for the claim that there is a Dedekind infinite set.

There is much more to say here, of course. But hopefully you are now at a point to start thinking about what it might *take* to justify an axiom (or principle). In what follows we will simply take *Stages-hit-infinity* for granted.

---

content/set-theory/z/milestone.tex

65.7 $\mathbf{Z}^-$: a Milestone

sth:z:milestone:sec

We will revisit *Stages-hit-infinity* in the next section. However, with the Axiom of Infinity, we have reached an important milestone. We now have all the axioms required for the theory $\mathbf{Z}^-$. In detail:

Definition 65.9. The theory $\mathbf{Z}^-$ has these axioms: Extensionality, Union, Pairs, Powersets, Infinity, and all instances of the Separation scheme.

The name stands for *Zermelo* set theory (*minus* something which we will come to later). Zermelo deserves the honour, since he essentially formulated this theory in his 1908a.[3]

This theory is powerful enough to allow us to do an enormous amount of mathematics. In particular, you *should* look back through part I, and convince yourself that everything we did, naïvely, could be done more formally within $\mathbf{Z}^-$. (Once you have done that for a bit, you might want to skip ahead and read section 65.9.) So, henceforth, and without any further comment, we will take ourselves to be working in $\mathbf{Z}^-$ (at least).

---

content/set-theory/z/nat.tex

65.8 Selecting our Natural Numbers

sth:z:nat:sec

In Definition 65.7, we explicitly defined the expression "natural numbers". How should you understand this stipulation? It is not a metaphysical claim, but just a decision to *treat* certain sets as the natural numbers. We touched upon reasons for thinking this in section 2.2, section 5.5 and section 6.4. But we can make these reasons even more pointed.

Our Axiom of Infinity follows von Neumann (1925). But here is another axiom, which we could have adopted instead:

> *Zermelo's 1908a Axiom of Infinity.* There is a set A such that $\emptyset \in A$ and $(\forall x \in A)\{x\} \in A$.

Had we used Zermelo's axiom, instead of our (von Neumann-inspired) Axiom of Infinity, we would equally well have been given a Dedekind infinite set, and so a Dedekind algebra. On Zermelo's approach, the distinguished element of our algebra would again have been $\emptyset$ (our surrogate for 0), but the injection would have been given by the map $x \mapsto \{x\}$, rather than $x \mapsto x \cup \{x\}$. The simplest upshot of this is that Zermelo treats 2 as $\{\{\emptyset\}\}$, whereas we (with von Neumann) treat 2 as $\{\emptyset, \{\emptyset\}\}$.

Why choose one axiom of Infinity rather than the other? The main practical reason is that von Neumann's approach "scales up" to handle transfinite numbers rather well. We will explore this from chapter 66 onwards. However,

[3] For interesting comments on the history and technicalities, see Potter (2004, Appendix A).

from the simple perspective of *doing arithmetic*, both approaches would do equally well. So if someone tells you that the natural numbers *are* sets, the obvious question is: *Which sets are they?*

This precise question was made famous by Benacerraf (1965). But it is worth emphasising that it is just the most famous example of a phenomenon that we have encountered many times already. The basic point is this. Set theory gives us a way to *simulate* a bunch of "intuitive" kinds of entities: the reals, rationals, integers, and naturals, yes; but also ordered pairs, functions, and relations. However, set theory never provides us with a *unique* choice of simulation. There are *always* alternatives which—straightforwardly—would have served us just as well.

---

content/set-theory/z/arbintersections.tex

65.9 Appendix: Closure, Comprehension, and Intersection

sth:z:arbintersections:sec

In section 65.7, we suggested that you should look back through the naïve work of part I and check that it can be carried out in $\mathbf{Z}^-$. If you followed that advice, *one* point might have tripped you up: the use of *intersection* in Dedekind's treatment of *closures*.

Recall from Definition 6.2 that

$$\mathrm{clo}_f(o) = \bigcap\{X : o \in X \text{ and } X \text{ is } f\text{-closed}\}.$$

The general shape of this is a definition of the form:

$$C = \bigcap\{X : \varphi(X)\}.$$

But this should ring alarm bells: since Naïve Comprehension fails, there is no guarantee that $\{X : \varphi(X)\}$ exists. It looks dangerously, then, like such definitions are *cheating*.

Fortunately, they are not cheating; or rather, if they *are* cheating as they stand, then we can engage in some honest toil to render them kosher. That honest toil was foreshadowed in Proposition 65.4, when we explained why $\bigcap A$ exists for any $A \neq \emptyset$. But we will spell it out explicitly.

Given Extensionality, if we attempt to define C as $\bigcap\{X : \varphi(X)\}$, all we are really asking is for an object C which obeys the following:

sth:z:arbintersections:bicondelimarbintersection

$$\forall x(x \in C \leftrightarrow \forall X(\varphi(X) \to x \in X)) \tag{*}$$

Now, suppose there is *some* set, S, such that $\varphi(S)$. Then to deliver eq. (*), we can simply define C using *Separation*, as follows:

$$C = \{x \in S : \forall X(\varphi(X) \to x \in X)\}.$$

We leave it as an exercise to check that this definition yields eq. (*), as desired. And this general strategy will allow us to circumvent any apparent use of Naïve Comprehension in defining intersections. In the particular case which got us started on this line of thought, namely that of $\mathrm{clo}_f(o)$, here is how that would work. We began the proof of Lemma 6.3 by noting that $o \in \mathrm{ran}(f) \cup \{o\}$ and that $\mathrm{ran}(f) \cup \{o\}$ is f-closed. So, we can define what we want thus:

$$\mathrm{clo}_f(o) = \{x \in \mathrm{ran}(f) \cup \{o\} : (\forall X \ni o)(X \text{ is } f\text{-closed} \to x \in X)\}.$$

Chapter 66

Ordinals

---

content/set-theory/ordinals/introduction.tex

66.1 Introduction

sth:ordinals:intro:sec

In chapter 65, we postulated that there is an infinite-th stage of the hierarchy, in the form of *Stages-hit-infinity* (see also our axiom of Infinity). However, given *Stages-keep-going*, we can't stop at the infinite-th stage; we have to keep going. So: at the next stage after the first infinite stage, we form all possible collections of sets that were available at the first infinite stage; and repeat; and repeat; and repeat; ...

Implicitly what has happened here is that we have started to invoke an "intuitive" notion of number, according to which there can be numbers *after* all the natural numbers. In particular, the notion involved is that of a *transfinite ordinal*. The aim of this chapter is to make this idea more rigorous. We will explore the general notion of an ordinal, and then explicitly define certain sets to be our ordinals.

---

content/set-theory/ordinals/idea.tex

66.2 The General Idea of an Ordinal

sth:ordinals:idea:sec

Consider the natural numbers, in their usual order:

$$0 < 1 < 2 < 3 < 4 < 5 < \cdots$$

We call this, in the jargon, an ω-sequence. And indeed, this general ordering is mirrored in our initial construction of the stages of the set hierarchy. But, now suppose we move 0 to the end of this sequence, so that it comes after all the other numbers:

$$1 < 2 < 3 < 4 < 5 < \cdots < 0$$

We have the same entities here, but ordered in a fundamentally different way: our first ordering had no last element; our new ordering does. Indeed, our new ordering consists of an ω-sequence of entities $(1, 2, 3, 4, 5, \ldots)$, followed by another entity. It will be an $\omega + 1$-sequence.

We can generate even more types of ordering, using just these entities. For example, consider all the even numbers (in their natural order) followed by all the odd numbers (in their natural order):

$$0 < 2 < 4 < \cdots < 1 < 3 < \cdots$$

This is an ω-sequence followed by another ω-sequence; an $\omega + \omega$-sequence.

Well, we can keep going. But what we would like is a general way to understand this talk about *orderings*.

---

content/set-theory/ordinals/wo.tex

66.3 Well-Orderings

sth:ordinals:wo:sec

The fundamental notion is as follows:

Definition 66.1. The relation $<$ *well-orders* A iff it meets these two conditions:

1. $<$ is connected, i.e., for all $a, b \in A$, either $a < b$ or $a = b$ or $b < a$;
2. every non-empty subset of A has a $<$-minimal element, i.e., if $\emptyset \neq X \subseteq A$ then $(\exists m \in X)(\forall z \in X) z \not< m$

It is easy to see that three examples we just considered were indeed well-ordering relations.

Problem 66.1. Section 66.2 presented three example orderings on the natural numbers. Check that each is a well-ordering.

Here are some elementary but extremely important observations concerning well-ordering.

sth:ordinals:wo:wo:strictorder

Proposition 66.2. *If $<$ well-orders A, then every non-empty subset of A has a unique $<$-least member, and $<$ is irreflexive, asymmetric and transitive.*

Proof. If X is a non-empty subset of A, it has a $<$-minimal element m, i.e., $(\forall z \in X) z \not< m$. Since $<$ is connected, $(\forall z \in X) m \leq z$. So m is the $<$-least element of X.

For irreflexivity, fix $a \in A$; the $<$-least element of $\{a\}$ is a, so $a \not< a$. For transitivity, if $a < b < c$, then since $\{a, b, c\}$ has a $<$-least element, $a < c$. Asymmetry follows from irreflexivity and transitivity □

Proposition 66.3. *If $<$ well-orders A, then for any formula $\varphi(x)$:* sth:ordinals:wo:propwoinduction

$$\text{if } (\forall a \in A)((\forall b < a)\varphi(b) \to \varphi(a)), \text{ then } (\forall a \in A)\varphi(a).$$

Proof. We will prove the contrapositive. Suppose $\neg(\forall a \in A)\varphi(a)$, i.e., that $X = \{x \in A : \neg\varphi(x)\} \neq \emptyset$. Then X has an $<$-minimal element, a. So $(\forall b < a)\varphi(b)$ but $\neg\varphi(a)$. □

This last property should remind you of the principle of strong induction on the naturals, i.e.: if $(\forall n \in \omega)((\forall m < n)\varphi(m) \to \varphi(n))$, then $(\forall n \in \omega)\varphi(n)$. And this property makes well-ordering into a very *robust* notion.[1]

---

content/set-theory/ordinals/iso.tex

66.4 Order-Isomorphisms

To explain *how* robust well-ordering is, we will start by introducing a method for comparing well-orderings. sth:ordinals:iso:sec

Definition 66.4. A *well-ordering* is a pair $\langle A, < \rangle$, such that $<$ well-orders A. The well-orderings $\langle A, < \rangle$ and $\langle B, \lessdot \rangle$ are *order-isomorphic* iff there is a bijection $f : A \to B$ such that: $x < y$ iff $f(x) \lessdot f(y)$. In this case, we write $\langle A, < \rangle \cong \langle B, \lessdot \rangle$, and say that f is an *order-isomorphism*.

In what follows, for brevity, we will speak of "isomorphisms" rather than "order-isomorphisms". Intuitively, isomorphisms are structure-preserving bijections. Here are some simple facts about isomorphisms.

Lemma 66.5. *Compositions of isomorphisms are isomorphisms, i.e.: if $f : A \to B$ and $g : B \to C$ are isomorphisms, then $(g \circ f) : A \to C$ is an isomorphism.* sth:ordinals:iso:isoscompose

Problem 66.2. Prove Lemma 66.5.

Proof. Left as an exercise. □

Corollary 66.6. *$X \cong Y$ is an equivalence relation.* sth:ordinals:iso:ordisoisequiv

Proposition 66.7. *If $\langle A, < \rangle$ and $\langle B, \lessdot \rangle$ are isomorphic well-orderings, then the isomorphism between them is unique.* sth:ordinals:iso:ordisounique

[1] A reminder: all formulas can have parameters (unless explicitly stated otherwise).

Proof. Let f and g be isomorphisms $A \to B$. We will prove the result by induction, i.e. using Proposition 66.3. Fix $a \in A$, and suppose (for induction) that $(\forall b < a) f(b) = g(b)$. Fix $x \in B$.

If $x \preccurlyeq f(a)$, then $f^{-1}(x) < a$, so $g(f^{-1}(x)) \preccurlyeq g(a)$, invoking the fact that f and g are isomorphisms. But since $f^{-1}(x) < a$, by our supposition $x = f(f^{-1}(x)) = g(f^{-1}(x))$. So $x \preccurlyeq g(a)$. Similarly, if $x \preccurlyeq g(a)$ then $x \preccurlyeq f(a)$.

Generalising, $(\forall x \in B)(x \preccurlyeq f(a) \leftrightarrow x \preccurlyeq g(a))$. It follows that $f(a) = g(a)$ by Proposition 2.27. So $(\forall a \in A) f(a) = g(a)$ by Proposition 66.3. □

This gives some sense that well-orderings are robust. But to continue explaining this, it will help to introduce some more notation.

Definition 66.8. When $\langle A, < \rangle$ is a well-ordering with $a \in A$, let $A_a = \{x \in A : x < a\}$. We say that A_a is a proper *initial segment* of A (and allow that A itself is an improper initial segment of A). Let $<_a$ be the restriction of $<$ to the initial segment, i.e., $<\restriction_{A_a^2}$.

Using this notation, we can state and prove that no well-ordering is isomorphic to any of its proper initial segments.

sth:ordinals:iso: wellordnotinitial

Lemma 66.9. *If $\langle A, < \rangle$ is a well-ordering with $a \in A$, then $\langle A, < \rangle \ncong \langle A_a, <_a \rangle$*

Proof. For reductio, suppose $f \colon A \to A_a$ is an isomorphism. Since f is a bijection and $A_a \subsetneq A$, using Proposition 66.2 let $b \in A$ be the $<$-least element of A such that $b \neq f(b)$. We'll show that $(\forall x \in A)(x < b \leftrightarrow x < f(b))$, from which it will follow by Proposition 2.27 that $b = f(b)$, completing the reductio.

Suppose $x < b$. So $x = f(x)$, by the choice of b. And $f(x) < f(b)$, as f is an isomorphism. So $x < f(b)$.

Suppose $x < f(b)$. So $f^{-1}(x) < b$, since f is an isomorphism, and so $f^{-1}(x) = x$ by the choice of b. So $x < b$. □

Our next result shows, roughly put, that an "initial segment" of an isomorphism is an isomorphism:

sth:ordinals:iso: wellordinitialsegment

Lemma 66.10. *Let $\langle A, < \rangle$ and $\langle B, \preccurlyeq \rangle$ be well-orderings. If $f \colon A \to B$ is an isomorphism and $a \in A$, then $f\restriction_{A_a} : A_a \to B_{f(a)}$ is an isomorphism.*

Proof. Since f is an isomorphism:

$$\begin{aligned} f[A_a] &= f[\{x \in A : x < a\}] \\ &= f[\{f^{-1}(y) \in A : f^{-1}(y) < a\}] \\ &= \{y \in B : y \preccurlyeq f(a)\} \\ &= B_{f(a)} \end{aligned}$$

And $f\restriction_{A_a}$ preserves order because f does. □

Our next two results establish that well-orderings are always comparable:

Lemma 66.11. *Let $\langle A, <\rangle$ and $\langle B, \lessdot\rangle$ be well-orderings. If $\langle A_{a_1}, <_{a_1}\rangle \cong \langle B_{b_1}, \lessdot_{b_1}\rangle$ and $\langle A_{a_2}, <_{a_2}\rangle \cong \langle B_{b_2}, \lessdot_{b_2}\rangle$, then $a_1 < a_2$ iff $b_1 \lessdot b_2$* sth:ordinals:iso:lemordsegments

Proof. We will prove *left to right*; the other direction is similar. Suppose both $\langle A_{a_1}, <_{a_1}\rangle \cong \langle B_{b_1}, \lessdot_{b_1}\rangle$ and $\langle A_{a_2}, <_{a_2}\rangle \cong \langle B_{b_2}, \lessdot_{b_2}\rangle$, with $f\colon A_{a_2} \to B_{b_2}$ our isomorphism. Let $a_1 < a_2$; then $\langle A_{a_1}, <_{a_1}\rangle \cong \langle B_{f(a_1)}, \lessdot_{f(a_1)}\rangle$ by Lemma 66.10. So $\langle B_{b_1}, \lessdot_{b_1}\rangle \cong \langle B_{f(a_1)}, \lessdot_{f(a_1)}\rangle$, and so $b_1 = f(a_1)$ by Lemma 66.9. Now $b_1 \lessdot b_2$ as f's domain is B_{b_2}. □

Theorem 66.12. *Given any two well-orderings, one is isomorphic to an initial segment (not necessarily proper) of the other.* sth:ordinals:iso:thm:woalwayscomparable

Proof. Let $\langle A, <\rangle$ and $\langle B, \lessdot\rangle$ be well-orderings. Using Separation, let

$$f = \{\langle a, b\rangle \in A \times B : \langle A_a, <_a\rangle \cong \langle B_b, \lessdot_b\rangle\}.$$

By Lemma 66.11, $a_1 < a_2$ iff $b_1 \lessdot b_2$ for all $\langle a_1, b_1\rangle, \langle a_2, b_2\rangle \in f$. So $f\colon \mathrm{dom}(f) \to \mathrm{ran}(f)$ is an isomorphism.

If $a_2 \in \mathrm{dom}(f)$ and $a_1 < a_2$, then $a_1 \in \mathrm{dom}(f)$ by Lemma 66.10; so $\mathrm{dom}(f)$ is an initial segment of A. Similarly, $\mathrm{ran}(f)$ is an initial segment of B. For reductio, suppose both are *proper* initial segments. Then let a be the $<$-least element of $A \setminus \mathrm{dom}(f)$, so that $\mathrm{dom}(f) = A_a$, and let b be the $\lessdot$-least element of $B \setminus \mathrm{ran}(f)$, so that $\mathrm{ran}(f) = B_b$. So $f\colon A_a \to B_b$ is an isomorphism, and hence $\langle a, b\rangle \in f$, a contradiction. □

---

content/set-theory/ordinals/vn.tex

66.5 Von Neumann's Construction of the Ordinals

Theorem 66.12 gives rise to a thought. We could introduce certain objects, called *order types*, to go proxy for the well-orderings. Writing $\mathrm{ord}(A, <)$ for the order type of the well-ordering $\langle A, <\rangle$, we would hope to secure the following two principles: sth:ordinals:vn:sec

$$\mathrm{ord}(A, <) = \mathrm{ord}(B, \lessdot) \text{ iff } \langle A, <\rangle \cong \langle B, \lessdot\rangle$$
$$\mathrm{ord}(A, <) < \mathrm{ord}(B, \lessdot) \text{ iff } \langle A, <\rangle \cong \langle B_b, \lessdot_b\rangle \text{ for some } b \in B$$

Moreover, we might hope to introduce order-types *as certain sets*, just as we can introduce the natural numbers as certain sets.

The most common way to do this—and the approach we will follow—is to define these order-types via certain *canonical* well-ordered sets. These canonical sets were first introduced by von Neumann:

Definition 66.13. The set A is *transitive* iff $(\forall x \in A) x \subseteq A$. Then A is an *ordinal* iff A is transitive and well-ordered by $\in$.

In what follows, we will use Greek letters for ordinals. It follows immediately from the definition that, if α is an ordinal, then $\langle \alpha, \in_\alpha \rangle$ is a well-ordering, where $\in_\alpha = \{\langle x, y\rangle \in \alpha^2 : x \in y\}$. So, abusing notation a little, we can just say that α *itself* is a well-ordering.

Here are our first few ordinals:

$$\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}, \dots$$

You will note that these are the first few ordinals that we encountered in our Axiom of Infinity, i.e., in von Neumann's definition of ω (see section 65.6). This is no coincidence. Von Neumann's definition of the ordinals treats natural numbers as ordinals, but allows for transfinite ordinals too.

As always, we can now ask: *are* these the ordinals? Or has von Neumann simply given us some sets that we can *treat* as the ordinals? The kinds of discussions one might have about this question are similar to the discussions we had in section 2.2, section 5.5, section 6.4, and section 65.8, so we will not belabour the point. Instead, in what follows, we will simply use "the ordinals" to speak of "the von Neumann ordinals".

---

content/set-theory/ordinals/basic.tex

66.6 Basic Properties of the Ordinals

sth:ordinals:basic:sec

We observed that the first few ordinals are the natural numbers. The main reason for developing a theory of ordinals is to extend the principle of induction which holds on the natural numbers. We will build up to this via a sequence of elementary results.

sth:ordinals:basic:ordmemberord

Lemma 66.14. *Every element of an ordinal is an ordinal.*

Proof. Let α be an ordinal with $b \in \alpha$. Since α is transitive, $b \subseteq \alpha$. So $\in$ well-orders b as $\in$ well-orders α.

To see that b is transitive, suppose $x \in c \in b$. So $c \in \alpha$ as $b \subseteq \alpha$. Again, as α is transitive, $c \subseteq \alpha$, so that $x \in \alpha$. So $x, c, b \in \alpha$. But $\in$ well-orders α, so that $\in$ is a transitive relation on α by Proposition 66.2. So since $x \in c \in b$, we have $x \in b$. Generalising, $c \subseteq b$ □

sth:ordinals:basic:ordissetofsmallerord

Corollary 66.15. $\alpha = \{\beta \in \alpha : \beta \text{ is an ordinal}\}$, *for any ordinal* α

Proof. Immediate from Lemma 66.14. □

The rough gist of the next two main results, Theorem 66.16 and Theorem 66.17, is that the ordinals themselves are well-ordered by membership:

sth:ordinals:basic:ordinductionschema

Theorem 66.16 (Transfinite Induction). *For any formula* $\varphi(x)$*:*

$$\text{if } \exists\alpha\varphi(\alpha), \text{ then } \exists\alpha(\varphi(\alpha) \wedge (\forall\beta \in \alpha)\neg\varphi(\beta))$$

where the displayed quantifiers are implicitly restricted to ordinals.

Proof. Suppose $\varphi(\alpha)$, for some ordinal α. If $(\forall\beta \in \alpha)\neg\varphi(\beta)$, then we are done. Otherwise, as α is an ordinal, it has some $\in$-least element which is φ, and this is an ordinal by Lemma 66.14. □

Note that we can equally express Theorem 66.16 as the scheme:

$$\text{if } \forall\alpha((\forall\beta \in \alpha)\varphi(\beta) \to \varphi(\alpha)), \text{ then } \forall\alpha\varphi(\alpha)$$

just by taking $\neg\varphi(\alpha)$ in Theorem 66.16, and then performing elementary logical manipulations.

Theorem 66.17 (Trichotomy). *$\alpha \in \beta \vee \alpha = \beta \vee \beta \in \alpha$, for any ordinals α and β.* *sth:ordinals:basic:ordtrichotomy*

Proof. The proof is by double induction, i.e., using Theorem 66.16 twice. Say that x is *comparable* with y iff $x \in y \vee x = y \vee y \in x$.

For induction, suppose that every ordinal in α is comparable with *every* ordinal. For further induction, suppose that α is comparable with every ordinal in β. We will show that α is comparable with β. By induction on β, it will follow that α is comparable with every ordinal; and so by induction on α, *every* ordinal is comparable with *every* ordinal, as required. It suffices to assume that $\alpha \notin \beta$ and $\beta \notin \alpha$, and show that $\alpha = \beta$.

To show that $\alpha \subseteq \beta$, fix $\gamma \in \alpha$; this is an ordinal by Lemma 66.14. So by the first induction hypothesis, γ is comparable with β. But if either $\gamma = \beta$ or $\beta \in \gamma$ then $\beta \in \alpha$ (invoking the fact that α is transitive if necessary), contrary to our assumption; so $\gamma \in \beta$. Generalising, $\alpha \subseteq \beta$.

Exactly similar reasoning, using the second induction hypothesis, shows that $\beta \subseteq \alpha$. So $\alpha = \beta$. □

As such, we will sometimes write $\alpha < \beta$ rather than $\alpha \in \beta$, since $\in$ is behaving as an ordering relation. There are no deep reasons for this, beyond familiarity, and because it is easier to write $\alpha \leq \beta$ than $\alpha \in \beta \vee \alpha = \beta$.[2]

Here are two quick consequences of our last results, the first of which puts our new notation into action:

Corollary 66.18. *If $\exists\alpha\varphi(\alpha)$, then $\exists\alpha(\varphi(\alpha) \wedge \forall\beta(\varphi(\beta) \to \alpha \leq \beta))$. Moreover, for any ordinals α, β, γ, both $\alpha \notin \alpha$ and $\alpha \in \beta \in \gamma \to \alpha \in \gamma$.* *sth:ordinals:basic:ordordered*

Proof. Just like Proposition 66.2. □

Problem 66.3. Complete the "exactly similar reasoning" in the proof of Theorem 66.17.

Corollary 66.19. *A is an ordinal iff A is a transitive set of ordinals.* *sth:ordinals:basic:corordtransitiveord*

Proof. Left-to-right. By Lemma 66.14. *Right-to-left.* If A is a transitive set of ordinals, then $\in$ well-orders A by Theorem 66.16 and Theorem 66.17. □

[2] We could write $\alpha \subseteq \beta$; but that would be wholly non-standard.

Now, we glossed Theorem 66.16 and Theorem 66.17 as telling us that $\in$ well-orders the ordinals. However, we have to be *very cautious* about this sort of claim, thanks to the following result:

sth:ordinals:basic: buraliforti

Theorem 66.20 (Burali-Forti Paradox). *There is no set of all the ordinals*

Proof. For reductio, suppose O is the set of all ordinals. If $\alpha \in \beta \in O$, then α is an ordinal, by Lemma 66.14, so $\alpha \in O$. So O is transitive, and hence O is an ordinal by Corollary 66.19. Hence $O \in O$, contradicting Corollary 66.18. □

This result is named after Burali-Forti. But, it was Cantor in 1899—in a letter to Dedekind—who first saw clearly the *contradiction* in supposing that there is a set of all the ordinals. As van Heijenoort explains:

> Burali-Forti himself considered the contradiction as establishing, by *reductio ad absurdum*, the result that the natural ordering of the ordinals is just a partial ordering. (Heijenoort, 1967, p. 105)

Setting Burali-Forti's mistake to one side, we can summarize the foregoing as follows. Ordinals are sets which are individually well-ordered by membership, and collectively well-ordered by membership (without collectively constituting a set).

Rounding this off, here are some more basic properties about the ordinals which follow from Theorem 66.16 and Theorem 66.17.

Proposition 66.21. *Any strictly descending sequence of ordinals is finite.*

Proof. Any infinite strictly descending sequence of ordinals $\alpha_0 > \alpha_1 > \alpha_2 > \dots$ has no $<$-minimal member, contradicting Theorem 66.16. □

sth:ordinals:basic: ordinalsaresubsets

Proposition 66.22. *$\alpha \subseteq \beta \vee \beta \subseteq \alpha$, for any ordinals α, β.*

Proof. If $\alpha \in \beta$, then $\alpha \subseteq \beta$ as β is transitive. Similarly, if $\beta \in \alpha$, then $\beta \subseteq \alpha$. And if $\alpha = \beta$, then $\alpha \subseteq \beta$ and $\beta \subseteq \alpha$. So by Theorem 66.17 we are done. □

sth:ordinals:basic: ordisoidentity

Proposition 66.23. *$\alpha = \beta$ iff $\alpha \cong \beta$, for any ordinals α, β.*

Proof. The ordinals are well-orders; so this is immediate from Trichotomy (Theorem 66.17) and Lemma 66.9. □

sth:ordinals:basic: probunionordinalsordinal

Problem 66.4. Prove that, if every member of X is an ordinal, then $\bigcup X$ is an ordinal.

66.7 Replacement

sth:ordinals:replacement:sec

In section 66.5, we motivated the introduction of ordinals by suggesting that we could treat them as order-types, i.e., canonical proxies for well-orderings. In order for that to work, we would need to prove that *every well-ordering is isomorphic to some ordinal*. This would allow us to define $\mathrm{ord}(A, <)$ as the ordinal α such that $\langle A, < \rangle \cong \alpha$.

Unfortunately, we *cannot* prove the desired result only the Axioms we provided introduced so far. (We will see why in section 68.2, but for now the point is: we can't.) We need a new thought, and here it is:

Axiom (Scheme of Replacement). For any formula $\varphi(x, y)$, the following is an axiom:

$$\text{for any } A, \text{ if } (\forall x \in A)\exists! y\, \varphi(x, y), \text{ then } \{y : (\exists x \in A)\varphi(x, y)\} \text{ exists.}$$

As with Separation, this is a scheme: it yields infinitely many axioms, for each of the infinitely many different φ's. And it can equally well be (and normally is) written down thus:

> For any formula $\varphi(x, y)$ which does not contain "B", the following is an axiom:
>
> $$\forall A[(\forall x \in A)\exists! y\, \varphi(x, y) \to \exists B \forall y(y \in B \leftrightarrow (\exists x \in A)\varphi(x, y))]$$

On first encounter, however, this is quite a tangled formula. The following quick consequence of Replacement probably gives a *clearer* expression to the intuitive idea we are working with:[3]

Corollary 66.24. *For any term $\tau(x)$, and any set A, this set exists:*

$$\{\tau(x) : x \in A\} = \{y : (\exists x \in A) y = \tau(x)\}.$$

Proof. Since τ is a *term*, $\forall x \exists! y\, \tau(x) = y$. A fortiori, $(\forall x \in A)\exists! y\, \tau(x) = y$. So $\{y : (\exists x \in A)\tau(x) = y\}$ exists by Replacement. □

This suggests that "Replacement" is a good name for the Axiom: given a set A, you can form a new set, $\{\tau(x) : x \in A\}$, by replacing every member of A with its image under τ. Indeed, following the notation for the image of a set under a function, we might write $\tau[A]$ for $\{\tau(x) : x \in A\}$.

Crucially, however, τ is a *term*. It need not be (a name for) a *function*, in the sense of section 3.3, i.e., a certain set of ordered pairs. After all, if f is a

[3] A term is an expression which picks out exactly one object (on any completion of its free variables). For example, "$\emptyset$" is a term which picks out the empty set; "$\{x\}$" is a term which picks out x's singleton (whatever x might be); "$x \cup y$" is a term which picks out the union of x and y (whatever they might be).

function (in that sense), then the set $f[A] = \{f(x) : x \in A\}$ is just a particular subset of $\mathrm{ran}(f)$, and that is already guaranteed to exist, just using the axioms of $\mathbf{Z}^-$.[4] Replacement, by contrast, is a *powerful* addition to our axioms, as we will see in chapter 68.

---

content/set-theory/ordinals/milestone.tex

66.8 $\mathbf{ZF}^-$: a milestone

sth:ordinals:zfm: sec

The question of how to justify Replacement (if at all) is not straightforward. As such, we will reserve that for chapter 68. However, with the addition of Replacement, we have reached another important milestone. We now have all the axioms required for the theory $\mathbf{ZF}^-$. In detail:

Definition 66.25. The theory $\mathbf{ZF}^-$ has these axioms: Extensionality, Union, Pairs, Powersets, Infinity, and all instances of the Separation and Replacement schemes. Otherwise put, $\mathbf{ZF}^-$ adds Replacement to $\mathbf{Z}^-$.

This stands for *Zermelo–Fraenkel* set theory (*minus* something which we will come to later). Fraenkel gets the honour, since he is credited with the formulation of Replacement in 1922, although the first precise formulation was due to Skolem (1922).

---

content/set-theory/ordinals/ordtype.tex

66.9 Ordinals as Order-Types

sth:ordinals:ordtype: sec

Armed with Replacement, and so now working in $\mathbf{ZF}^-$, we can finally prove the result we have been aiming for:

sth:ordinals:ordtype: thmOrdinalRepresentation

Theorem 66.26. *Every well-ordering is isomorphic to a unique ordinal.*

Proof. Let $\langle A, <\rangle$ be a well-order. By Proposition 66.23, it is isomorphic to at most one ordinal. So, for reductio, suppose $\langle A, <\rangle$ is not isomorphic to *any* ordinal. We will first "make $\langle A, <\rangle$ as small as possible". In detail: if some proper initial segment $\langle A_a, <_a\rangle$ is not isomorphic to any ordinal, there is a least $a \in A$ with that property; then let $B = A_a$ and $\lessdot = <_a$. Otherwise, let $B = A$ and $\lessdot = <$.

By definition, every proper initial segment of B is isomorphic to some ordinal, which is unique as above. So by Replacement, the following set exists, and is a function:

$$f = \{\langle \beta, b\rangle : b \in B \text{ and } \beta \cong \langle B_b, \lessdot_b\rangle\}$$

[4] Just consider $\{y \in \bigcup\bigcup f : (\exists x \in A) y = f(x)\}$.

To complete the reductio, we'll show that f is an isomorphism $\alpha \to B$, for some ordinal α.

It is obvious that $\mathrm{ran}(f) = B$. And by Lemma 66.11, f preserves ordering, i.e., $\gamma \in \beta$ iff $f(\gamma) \prec f(\beta)$. To show that $\mathrm{dom}(f)$ is an ordinal, by Corollary 66.19 it suffices to show that $\mathrm{dom}(f)$ is transitive. So fix $\beta \in \mathrm{dom}(f)$, i.e., $\beta \cong \langle B_b, \prec_b\rangle$ for some b. If $\gamma \in \beta$, then $\gamma \in \mathrm{dom}(f)$ by Lemma 66.10; generalising, $\beta \subseteq \mathrm{dom}(f)$. □

This result licenses the following definition, which we have wanted to offer since section 66.5:

Definition 66.27. If $\langle A, <\rangle$ is a well-ordering, then its order type, $\mathrm{ord}(A, <)$, is the unique ordinal α such that $\langle A, <\rangle \cong \alpha$.

Moreover, this definition licenses two nice principles:

Corollary 66.28. *Where $\langle A, <\rangle$ and $\langle B, \prec\rangle$ are well-orderings:* sth:ordinals:ordtype:ordtypesworklikeyouwant

$$\mathrm{ord}(A, <) = \mathrm{ord}(B, \prec) \text{ iff } \langle A, <\rangle \cong \langle B, \prec\rangle$$
$$\mathrm{ord}(A, <) \in \mathrm{ord}(B, \prec) \text{ iff } \langle A, <\rangle \cong \langle B_b, \prec_b\rangle \text{ for some } b \in B$$

Proof. The identity holds by Proposition 66.23. To prove the second claim, let $\mathrm{ord}(A, <) = \alpha$ and $\mathrm{ord}(B, \prec) = \beta$, and let $f : \beta \to \langle B, \prec\rangle$ be our isomorphism. Then:

$$\begin{aligned}\alpha \in \beta &\text{ iff } f\restriction_\alpha : \alpha \to B_{f(\alpha)} \text{ is an isomorphism}\\ &\text{iff } \langle A, <\rangle \cong \langle B_{f(\alpha)}, \prec_{f(\alpha)}\rangle\\ &\text{iff } \langle A, <\rangle \cong \langle B_b, \prec_b\rangle \text{ for some } b \in B\end{aligned}$$

by Proposition 66.7, Lemma 66.10, and Corollary 66.15. □

---

content/set-theory/ordinals/opps.tex

66.10 Successor and Limit Ordinals

In the next few chapters, we will use ordinals a great deal. So it will help if we introduce some simple notions. sth:ordinals:opps:sec

Definition 66.29. For any ordinal α, its *successor* is $\alpha^+ = \alpha \cup \{\alpha\}$. We say that α is a *successor* ordinal if $\beta^+ = \alpha$ for some ordinal β. We say that α is a *limit* ordinal iff α is neither empty nor a successor ordinal.

The following result shows that this is the right notion of *successor*:

Proposition 66.30. *For any ordinal α:*

1. $\alpha \in \alpha^+$;

2. *α^+ is an ordinal;*

3. *there is no ordinal β such that $\alpha \in \beta \in \alpha^+$.*

Proof. Trivially, $\alpha \in \alpha \cup \{\alpha\} = \alpha^+$. Equally, α^+ is a transitive set of ordinals, and hence an ordinal by Corollary 66.19. And it is impossible that $\alpha \in \beta \in \alpha^+$, since then either $\beta \in \alpha$ or $\beta = \alpha$, contradicting Corollary 66.18. □

This also licenses a variant of proof by transfinite induction:

sth:ordinals:opps:simpletransrecursion **Theorem 66.31 (Simple Transfinite Induction).** *Let $\varphi(x)$ be a formula such that:*

1. *$\varphi(\emptyset)$; and*

2. *for any ordinal α, if $\varphi(\alpha)$ then $\varphi(\alpha^+)$; and*

3. *if α is a limit ordinal and $(\forall \beta \in \alpha)\varphi(\beta)$, then $\varphi(\alpha)$.*

Then $\forall \alpha \varphi(\alpha)$.

Proof. We prove the contrapositive. So, suppose there is some ordinal which is $\neg\varphi$; let γ be the least such ordinal. Then either $\gamma = \emptyset$, or $\gamma = \alpha^+$ for some α such that $\varphi(\alpha)$; or γ is a limit ordinal and $(\forall \beta \in \gamma)\varphi(\beta)$. □

A final bit of notation will prove helpful later on:

sth:ordinals:opps:defsupstrict **Definition 66.32.** If X is a set of ordinals, then $\mathrm{lsub}(X) = \bigcup_{\alpha \in X} \alpha^+$.

Here, "lsub" stands for "least strict upper bound".[5] The following result explains this:

Proposition 66.33. *If X is a set of ordinals, $\mathrm{lsub}(X)$ is the least ordinal greater than every ordinal in X.*

Proof. Let $Y = \{\alpha^+ : \alpha \in X\}$, so that $\mathrm{lsub}(X) = \bigcup Y$. Since ordinals are transitive and every member of an ordinal is an ordinal, $\mathrm{lsub}(X)$ is a transitive set of ordinals, and so is an ordinal by Corollary 66.19.

If $\alpha \in X$, then $\alpha^+ \in Y$, so $\alpha^+ \subseteq \bigcup Y = \mathrm{lsub}(X)$, and hence $\alpha \in \mathrm{lsub}(X)$. So $\mathrm{lsub}(X)$ is strictly greater than every ordinal in X.

Conversely, if $\alpha \in \mathrm{lsub}(X)$, then $\alpha \in \beta^+ \in Y$ for some $\beta \in X$, so that $\alpha \leq \beta \in X$. So $\mathrm{lsub}(X)$ is the *least* strict upper bound on X. □

[5] Some books use "sup(X)" for this. But other books use "sup(X)" for the least *non-strict* upper bound, i.e., simply $\bigcup X$. If X has a greatest element, α, these notions come apart: the least *strict* upper bound is α^+, whereas the least *non-strict* upper bound is just α.

Chapter 67

Stages and Ranks

content/set-theory/spine/idea.tex

67.1 Defining the Stages as the V_αs

In chapter 66, we defined well-orderings and the (von Neumann) ordinals. In this chapter, we will use these to characterise the hierarchy of sets *itself*. To do this, recall that in section 66.10, we defined the idea of successor and limit ordinals. We use these ideas in following definition: sth:spine:valpha:sec

Definition 67.1. sth:spine:valpha:defValphas

$$\begin{aligned} V_\emptyset &= \emptyset && \\ V_{\alpha^+} &= \wp(V_\alpha) && \text{for any ordinal } \alpha \\ V_\alpha &= \bigcup_{\gamma<\alpha} V_\gamma && \text{when } \alpha \text{ is a limit ordinal} \end{aligned}$$

This will be a definition by *transfinite recursion* on the ordinals. In this regard, we should compare this with recursive definitions of functions on the natural numbers.[1] As when dealing with natural numbers, one defines a base case and successor cases; but when dealing with ordinals, we also need to describe the behaviour of *limit* cases.

This definition of the V_αs will be an important milestone. We have informally motivated our hierarchy of sets as forming sets by *stages*. The V_αs are, in effect, just those stages. Importantly, though, this is an *internal* characterisation of the stages. Rather than suggesting a possible *model* of the theory, we will have defined the stages *within* our set theory.

content/set-theory/spine/recursion.tex

[1] Cf. the definitions of addition, multiplication, and exponentiation in section 6.2.

67.2 The Transfinite Recursion Theorem(s)

sth:spine:recursion:sec

The first thing we must do, though, is confirm that Definition 67.1 is a successful definition. More generally, we need to prove that any attempt to offer a transfinite by (transfinite) recursion will succeed. That is the aim of this section.

Warning: this is tricky material. The overarching moral, though, is quite simple: Transfinite Induction plus Replacement guarantee the legitimacy of (several versions of) transfinite recursion.[2]

Definition 67.2. Let $\tau(x)$ be a term; let f be a function; let α be an ordinal. We say that f is an *α-approximation* for τ iff both $\mathrm{dom}(f) = \alpha$ and $(\forall \beta \in \alpha) f(\beta) = \tau(f{\restriction_\beta})$.

sth:spine:recursion:transrecursionfun

Lemma 67.3 (Bounded Recursion). *For any term $\tau(x)$ and any ordinal α, there is a unique α-approximation for τ.*

Proof. We will show that, for any $\gamma \leq \alpha$, there is a unique γ-approximation.

We first establish uniqueness. Let g and h (respectively) be γ- and δ-approximations. A transfinite induction on their arguments shows that $g(\beta) = h(\beta)$ for any $\beta \in \mathrm{dom}(g) \cap \mathrm{dom}(h) = \gamma \cap \delta = \min(\gamma, \delta)$. So our approximations are unique (if they exist), and agree on all values.

To establish existence, we now use a simple transfinite induction (Theorem 66.31) on ordinals $\delta \leq \alpha$.

The empty function is trivially an $\emptyset$-approximation.

If g is a γ-approximation, then $g \cup \{\langle \gamma, \tau(g)\rangle\}$ is a γ^+-approximation.

If γ is a limit ordinal and g_δ is a δ-approximation for all $\delta < \gamma$, let $g = \bigcup_{\delta\in\gamma} g_\delta$. This is a function, since our various g_δs agree on all values. And if $\delta \in \gamma$ then $g(\delta) = g_{\delta^+}(\delta) = \tau(g_{\delta^+}{\restriction_\delta}) = \tau(g{\restriction_\delta})$.

This completes the proof by transfinite induction. □

If we allow ourselves to define a *term* rather than a function, then we can remove the bound α from the previous result. In the statement and proof of the following result, when σ is a term, we let $\sigma{\restriction_\alpha} = \{\langle \beta, \sigma(\beta)\rangle : \beta \in \alpha\}$.

sth:spine:recursion:transrecursionschema

Theorem 67.4 (General Recursion). *For any term $\tau(x)$, we can explicitly define a term $\sigma(x)$, such that $\sigma(\alpha) = \tau(\sigma{\restriction_\alpha})$ for any ordinal α.*

[2] A reminder: all formulas and terms can have parameters (unless explicitly stated otherwise).

Proof. For each α, by Lemma 67.3 there is a unique α-approximation, f_α, for τ. Define $\sigma(\alpha)$ as $f_{\alpha^+}(\alpha)$. Now:

$$\begin{aligned}\sigma(\alpha) &= f_{\alpha^+}(\alpha)\\ &= \tau(f_{\alpha^+}{\restriction_\alpha})\\ &= \tau(\{\langle\beta, f_{\alpha^+}(\beta)\rangle : \beta \in \alpha\})\\ &= \tau(\{\langle\beta, f_{\beta^+}(\beta)\rangle : \beta \in \alpha\})\\ &= \tau(\sigma{\restriction_\alpha})\end{aligned}$$

noting that $f_{\beta^+}(\beta) = f_{\alpha^+}(\beta)$ for all $\beta < \alpha$, as in Lemma 67.3. □

Note that Theorem 67.4 is a *schema*. Crucially, we cannot expect σ to define a function, i.e., a certain kind of *set*, since then $\mathrm{dom}(\sigma)$ would be the set of all ordinals, contradicting the Burali-Forti Paradox (Theorem 66.20).

It still remains to show, though, that Theorem 67.4 vindicates our definition of the V_αs. This may not be immediately obvious; but it will become apparent with a last, simple, version of transfinite recursion.

Theorem 67.5 (Simple Recursion). *For any terms $\tau(x)$ and $\theta(x)$ and any set A, we can explicitly define a term $\sigma(x)$ such that:* sth:spine:recursion:simplerecursionschema

$$\begin{aligned}\sigma(\emptyset) &= A &&\\ \sigma(\alpha^+) &= \tau(\sigma(\alpha)) && \text{for any ordinal } \alpha\\ \sigma(\alpha) &= \theta(\mathrm{ran}(\sigma{\restriction_\alpha})) && \text{when } \alpha \text{ is a limit ordinal}\end{aligned}$$

Proof. We start by defining a term, $\xi(x)$, as follows:

$$\xi(x) = \begin{cases} A & \text{if } x \text{ is not a function whose}\\ & \quad\text{domain is an ordinal; otherwise:}\\ \tau(x(\alpha)) & \text{if } \mathrm{dom}(x) = \alpha^+\\ \theta(\mathrm{ran}(x)) & \text{if } \mathrm{dom}(x) \text{ is a limit ordinal}\end{cases}$$

By Theorem 67.4, there is a term $\sigma(x)$ such that $\sigma(\alpha) = \xi(\sigma{\restriction_\alpha})$ for every ordinal α; moreover, $\sigma{\restriction_\alpha}$ is a function with domain α. We show that σ has the required properties, by simple transfinite induction (Theorem 66.31).

First, $\sigma(\emptyset) = \xi(\emptyset) = A$.

Next, $\sigma(\alpha^+) = \xi(\sigma{\restriction_{\alpha^+}}) = \tau(\sigma{\restriction_{\alpha^+}}(\alpha)) = \tau(\sigma(\alpha))$.

Last, $\sigma(\alpha) = \xi(\sigma{\restriction_\alpha}) = \theta(\mathrm{ran}(\sigma{\restriction_\alpha}))$, when α is a limit. □

Now, to vindicate Definition 67.1, just take $A = \emptyset$ and $\tau(x) = \wp(x)$ and $\theta(x) = \bigcup x$. At long last, this vindicates the definition of the V_αs!

67.3 Basic Properties of Stages

sth:spine:Valphabasic:sec

To bring out the foundational importance of the definition of the V_αs, we will present a few basic results about them. We start with a definition:[3]

Definition 67.6. The set A is *potent* iff $\forall x((\exists y \in A)x \subseteq y \to x \in A)$.

sth:spine:Valphabasic:Valphabasicprops

Lemma 67.7. *For each ordinal α:*

1. *Each V_α is transitive.* sth:spine:Valphabasic:Valphatrans
2. *Each V_α is potent.* sth:spine:Valphabasic:Valphapotent
3. *If $\gamma \in \alpha$, then $V_\gamma \in V_\alpha$ (and hence also $V_\gamma \subseteq V_\alpha$ by (1))* sth:spine:Valphabasic:Valphacum

Proof. We prove this by a (simultaneous) transfinite induction. For induction, suppose that (1)–(3) holds for each ordinal $\beta < \alpha$.

The case of $\alpha = \emptyset$ is trivial.

Suppose $\alpha = \beta^+$. To show (3), if $\gamma \in \alpha$ then $V_\gamma \subseteq V_\beta$ by hypothesis, so $V_\gamma \in \wp(V_\beta) = V_\alpha$. To show (2), suppose $A \subseteq B \in V_\alpha$ i.e., $A \subseteq B \subseteq V_\beta$; then $A \subseteq V_\beta$ so $A \in V_\alpha$. To show (1), note that if $x \in A \in V_\alpha$ we have $A \subseteq V_\beta$, so $x \in V_\beta$, so $x \subseteq V_\beta$ as V_β is transitive by hypothesis, and so $x \in V_\alpha$.

Suppose α is a limit ordinal. To show (3), if $\gamma \in \alpha$ then $\gamma \in \gamma^+ \in \alpha$, so that $V_\gamma \in V_{\gamma^+}$ by assumption, hence $V_\gamma \in \bigcup_{\beta\in\alpha} V_\beta = V_\alpha$. To show (1) and (2), just observe that a union of transitive (respectively, potent) sets is transitive (respectively, potent). □

sth:spine:Valphabasic:Valphanotref

Lemma 67.8. *For each ordinal α, $V_\alpha \notin V_\alpha$.*

Proof. By transfinite induction. Evidently $V_\emptyset \notin V_\emptyset$.

If $V_{\alpha^+} \in V_{\alpha^+} = \wp(V_\alpha)$, then $V_{\alpha^+} \subseteq V_\alpha$; and since $V_\alpha \in V_{\alpha^+}$ by Lemma 67.7, we have $V_\alpha \in V_\alpha$. Conversely: if $V_\alpha \notin V_\alpha$ then $V_{\alpha^+} \notin V_{\alpha^+}$

If α is a limit and $V_\alpha \in V_\alpha = \bigcup_{\beta\in\alpha} V_\beta$, then $V_\alpha \in V_\beta$ for some $\beta \in \alpha$; but then also $V_\beta \in V_\alpha$ so that $V_\beta \in V_\beta$ by Lemma 67.7 (twice). Conversely, if $V_\beta \notin V_\beta$ for all $\beta \in \alpha$, then $V_\alpha \notin V_\alpha$. □

Corollary 67.9. *For any ordinals α, β: $\alpha \in \beta$ iff $V_\alpha \in V_\beta$*

Proof. Lemma 67.7 gives one direction. Conversely, suppose $V_\alpha \in V_\beta$. Then $\alpha \neq \beta$ by Lemma 67.8; and $\beta \notin \alpha$, for otherwise we would have $V_\beta \in V_\alpha$ and hence $V_\beta \in V_\beta$ by Lemma 67.7 (twice), contradicting Lemma 67.8. So $\alpha \in \beta$ by Trichotomy. □

All of this allows us to think of each V_α as the αth stage of the hierarchy. Here is why.

Certainly our V_αs can be thought of as being formed in an *iterative* process, for our use of ordinals tracks the notion of iteration. Moreover, if one stage

[3] There's no standard terminology for "potent"; this is the name used by Button (2021).

is formed before the other, i.e., $V_\beta \in V_\alpha$, i.e., $\beta \in \alpha$, then our process of formation is *cumulative*, since $V_\beta \subseteq V_\alpha$. Finally, we are indeed forming *all* possible collections of sets that were available at any earlier stage, since any successor stage V_{α^+} is the power-set of its predecessor V_α.

In short: with $\mathbf{ZF}^-$, we are *almost* done, in articulating our vision of the cumulative-iterative hierarchy of sets. (Though, of course, we still need to justify Replacement.)

---

content/set-theory/spine/foundation.tex

67.4 Foundation

sth:spine:foundation:sec

We are only *almost* done—and not *quite* finished—because nothing in $\mathbf{ZF}^-$ guarantees that *every* set is in some V_α, i.e., that every set is formed at some stage.

Now, there is a fairly straightforward (mathematical) sense in which we don't *care* whether there are sets outside the hierarchy. (If there are any there, we can simply ignore them.) But we have motivated our *concept* of set with the thought that every set is formed at some stage (see *Stages-are-key* in section 65.1). So we will want to preclude the possibility of sets which fall outside of the hierarchy. Accordingly, we must add a new axiom, which ensures that every set occurs somewhere in the hierarchy.

Since the V_αs are our stages, we might simply consider adding the following as an axiom:

> *Regularity.* $\forall A \exists \alpha\, A \subseteq V_\alpha$

This would be a perfectly reasonable approach. However, for reasons that will be explained in the next section, we will instead adopt an alternative axiom:

Axiom (Foundation). $(\forall A \neq \emptyset)(\exists B \in A) A \cap B = \emptyset$.

With some effort, we can show (in $\mathbf{ZF}^-$) that Foundation entails Regularity:

Definition 67.10. For each set A, let:

$$\begin{aligned} \mathrm{cl}_0(A) &= A, \\ \mathrm{cl}_{n+1}(A) &= \bigcup \mathrm{cl}_n(A), \\ \mathrm{trcl}(A) &= \bigcup_{n<\omega} \mathrm{cl}_n(A). \end{aligned}$$

We call $\mathrm{trcl}(A)$ the *transitive closure* of A.

The name "transitive closure" is apt:

sth:spine:foundation: subsetoftrcl

Proposition 67.11. *$A \subseteq \mathrm{trcl}(A)$ and $\mathrm{trcl}(A)$ is a transitive set.*

Proof. Evidently $A = \mathrm{cl}_0(A) \subseteq \mathrm{trcl}(A)$. And if $x \in b \in \mathrm{trcl}(A)$, then $b \in \mathrm{cl}_n(A)$ for some n, so $x \in \mathrm{cl}_{n+1}(A) \subseteq \mathrm{trcl}(A)$. □

sth:spine:foundation: lem:TransitiveWellFounded

Lemma 67.12. *If A is a transitive set, then there is some α such that $A \subseteq V_\alpha$.*

Proof. Recalling the definition of "lsub(X)" from Definition 66.32, define two sets:

$$D = \{x \in A : \forall \delta\ x \nsubseteq V_\delta\}$$
$$\alpha = \mathrm{lsub}\{\delta : (\exists x \in A)(x \subseteq V_\delta \wedge (\forall \gamma \in \delta) x \nsubseteq V_\gamma)\}$$

Suppose $D = \emptyset$. So if $x \in A$, then there is some δ such that $x \subseteq V_\delta$ and, by the well-ordering of the ordinals, $(\forall \gamma \in \delta) x \nsubseteq V_\gamma$; hence $\delta \in \alpha$ and so $x \in V_\alpha$ by Lemma 67.7. Hence $A \subseteq V_\alpha$, as required.

So it suffices to show that $D = \emptyset$. For reductio, suppose otherwise. By Foundation, there is some $B \in D \subseteq A$ such that $D \cap B = \emptyset$. If $x \in B$ then $x \in A$, since A is transitive, and since $x \notin D$, it follows that $\exists \delta\ x \subseteq V_\delta$. So now let

$$\beta = \mathrm{lsub}\{\delta : (\exists x \in b)(x \subseteq V_\delta \wedge (\forall \gamma < \delta) x \nsubseteq V_\gamma)\}.$$

As before, $B \subseteq V_\beta$, contradicting the claim that $B \in D$. □

sth:spine:foundation: zfentailsregularity

Theorem 67.13. *Regularity holds.*

Proof. Fix A; now $A \subseteq \mathrm{trcl}(A)$ by Proposition 67.11, which is transitive. So there is some α such that $A \subseteq \mathrm{trcl}(A) \subseteq V_\alpha$ by Lemma 67.12 □

These results show that **ZF**$^-$ proves the conditional *Foundation* ⇒ *Regularity*. In Proposition 67.22, we will show that **ZF**$^-$ proves *Regularity* ⇒ *Foundation*. As such, Foundation and Regularity are *equivalent* (modulo **ZF**$^-$). But this means that, given **ZF**$^-$, we can justify Foundation by noting that it is equivalent to Regularity. And we can justify Regularity immediately on the basis of *Stages-are-key*.

---

content/set-theory/spine/zf.tex

67.5 Z and ZF: A Milestone

sth:spine:zf: sec

With Foundation, we reach another important milestone. We have considered theories **Z**$^-$ and **ZF**$^-$, which we said were certain theories "minus" a certain something. That certain something is Foundation. So:

Definition 67.14. The theory **Z** adds Foundation to **Z**$^-$. So its axioms are Extensionality, Union, Pairs, Powersets, Infinity, Foundation, and all instances of the Separation scheme.

The theory **ZF** adds Foundation to **ZF**$^-$. Otherwise put, **ZF** adds all instances of Replacement to **Z**.

Still, one question might have occurred to you. If Regularity is equivalent over $\mathbf{ZF}^-$ to Foundation, and Regularity's justification is clear, why bother to go around the houses, and take Foundation as our basic axiom, rather than Regularity?

Setting aside historical reasons (to do with who formulated what and when), the basic reason is that Foundation can be presented without employing the definition of the V_αs. That definition relied upon all of the work of section 67.2: we needed to prove Transfinite Recursion, to show that it was justified. But our proof of Transfinite Recursion employed *Replacement*. So, whilst Foundation and Regularity are equivalent modulo $\mathbf{ZF}^-$, they are not equivalent modulo $\mathbf{Z}^-$.

Indeed, the matter is more drastic than this simple remark suggests. Though it goes well beyond this book's remit, it turns out that both $\mathbf{Z}^-$ and $\mathbf{Z}$ are too weak to define the V_αs. So, if you are working only in $\mathbf{Z}$, then Regularity (as we have formulated it) does not even make *sense*. This is why our official axiom is Foundation, rather than Regularity.

From now on, we will work in $\mathbf{ZF}$ (unless otherwise stated), without any further comment.

---

content/set-theory/spine/rank.tex

67.6 Rank

sth:spine:rank:sec

Now that we have defined the stages as the V_α's, and we know that every set is a subset of some stage, we can define the *rank* of a set. Intuitively, the rank of A is the first moment at which A is formed. More precisely:

sth:spine:rank:defnsetrank

Definition 67.15. For each set A, $\mathrm{rank}(A)$ is the least ordinal α such that $A \subseteq V_\alpha$.

sth:spine:rank:ranksexist

Proposition 67.16. $\mathrm{rank}(A)$ *exists, for any* A.

Proof. Left as an exercise. □

Problem 67.1. Prove Proposition 67.16.

The well-ordering of ranks allows us to prove some important results:

sth:spine:rank:valphalowerrank

Proposition 67.17. *For any ordinal* α, $V_\alpha = \{x : \mathrm{rank}(x) \in \alpha\}$.

Proof. If $\mathrm{rank}(x) \in \alpha$ then $x \subseteq V_{\mathrm{rank}(x)} \in V_\alpha$, so $x \in V_\alpha$ as V_α is potent (invoking Lemma 67.7 multiple times). Conversely, if $x \in V_\alpha$ then $x \subseteq V_\alpha$, so $\mathrm{rank}(x) \leq \alpha$; now a simple transfinite induction shows that $x \notin V_\alpha$. □

Problem 67.2. Complete the simple transfinite induction mentioned in Proposition 67.17.

sth:spine:rank:rankmemberslower

Proposition 67.18. *If* $B \in A$, *then* $\mathrm{rank}(B) \in \mathrm{rank}(A)$.

Proof. $A \subseteq V_{\mathrm{rank}(A)} = \{x : \mathrm{rank}(x) \in \mathrm{rank}(A)\}$ by Proposition 67.17. □

Using this fact, we can establish a result which allows us to prove things about *all sets* by a form of induction:

Theorem 67.19 (∈-Induction Scheme). *For any formula φ:*

$$\forall A((\forall x \in A)\varphi(x) \to \varphi(A)) \to \forall A \varphi(A).$$

Proof. We will prove the contrapositive. So, suppose $\neg\forall A\varphi(A)$. By Transfinite Induction (Theorem 66.16), there is some non-φ of least possible rank; i.e. some A such that $\neg\varphi(A)$ and $\forall x(\mathrm{rank}(x) \in \mathrm{rank}(A) \to \varphi(x))$. Now if $x \in A$ then $\mathrm{rank}(x) \in \mathrm{rank}(A)$, by Proposition 67.18, so that $\varphi(x)$; i.e. $(\forall x \in A)\varphi(x) \wedge \neg\varphi(A)$. □

Here is an informal way to gloss this powerful result. Say that φ is *hereditary* iff whenever every element of a set is φ, the set itself is φ. Then ∈-Induction tells you the following: if φ is hereditary, every set is φ.

To wrap up the discussion of ranks (for now), we'll prove a few claims which we have foreshadowed a few times.

sth:spine:rank: ranksupstrict

Proposition 67.20. $\mathrm{rank}(A) = \mathrm{lsub}_{x \in A} \mathrm{rank}(x)$.

Proof. Let $\alpha = \mathrm{lsub}_{x \in A} \mathrm{rank}(x)$. By Proposition 67.18, $\alpha \leq \mathrm{rank}(A)$. But if $x \in A$ then $\mathrm{rank}(x) \in \alpha$, so that $x \in V_\alpha$ by Proposition 67.17, and hence $A \subseteq V_\alpha$, i.e., $\mathrm{rank}(A) \leq \alpha$. Hence $\mathrm{rank}(A) = \alpha$. □

sth:spine:rank: ordsetrankalpha

Corollary 67.21. *For any ordinal α,* $\mathrm{rank}(\alpha) = \alpha$.

Proof. Suppose for transfinite induction that $\mathrm{rank}(\beta) = \beta$ for all $\beta \in \alpha$. Now $\mathrm{rank}(\alpha) = \mathrm{lsub}_{\beta \in \alpha} \mathrm{rank}(\beta) = \mathrm{lsub}_{\beta \in \alpha} \beta = \alpha$ by Proposition 67.20. □

Finally, here is a quick proof of the result promised at the end of section 67.4, that $\mathbf{ZF}^-$ proves the conditional *Regularity* ⇒ *Foundation*. (Note that the notion of "rank" and Proposition 67.18 are available for use in this proof since—as mentioned at the start of this section—they can be presented using $\mathbf{ZF}^-$ + Regularity.)

sth:spine:rank: zfminusregularityfoundation

Proposition 67.22 (working in $\mathbf{ZF}^-$ + Regularity). *Foundation holds.*

Proof. Fix $A \neq \emptyset$, and some $B \in A$ of least possible rank. If $c \in B$ then $\mathrm{rank}(c) \in \mathrm{rank}(B)$ by Proposition 67.18, so that $c \notin A$ by choice of B. □

Chapter 68

Replacement

content/set-theory/replacement/introduction.tex

68.1 Introduction

sth:replacement:intro:sec

Replacement is the axiom scheme which makes the difference between **ZF** and **Z**. We helped ourselves to it throughout chapters 66 to 67. In this chapter, we will finally consider the question: is Replacement justified?

To make the question sharp, it is worth observing that Replacement is really rather *strong*. We will get a sense of just how strong it is, during this chapter (and again in section 71.5). But this will suggest that justification really is required.

We will discuss two kinds of justification. Roughly: an *extrinsic* justification is an attempt to justify an axiom by its fruits; an *intrinsic* justification is an attempt to justify an axiom by suggesting that it is vindicated by the mathematical concepts in question. We will get a greater sense of what this means during this chapter, but it is just the tip of an iceberg. For more, see in particular Maddy (1988a and 1988b).

content/set-theory/replacement/strength.tex

68.2 The Strength of Replacement

sth:replacement:strength:sec

We begin with a simple observation about the strength of Replacement: unless we go beyond **Z**, we cannot prove the existence of any von Neumann ordinal greater than or equal to $\omega + \omega$.

Here is a sketch of why. Working in **ZF**, consider the set $V_{\omega+\omega}$. This set acts as the domain for a *model* for **Z**. To see this, we introduce some notation for the *relativization* of a formula:

sth:replacement:strength:formularelativization

Definition 68.1. For any set M, and any formula φ, let φ^M be the formula which results by restricting all of φ's quantifiers to M. That is, replace "$\exists x$" with "$(\exists x \in M)$", and replace "$\forall x$" with "$(\forall x \in M)$".

It can be shown that, for every axiom φ of **Z**, we have that $\mathbf{ZF} \vdash \varphi^{V_{\omega+\omega}}$. But $\omega + \omega$ is not *in* $V_{\omega+\omega}$, by Corollary 67.21. So **Z** is consistent with the non-existence of $\omega + \omega$.

This is why we said, in section 66.7, that Theorem 66.26 cannot be proved without Replacement. For it is easy, within **Z**, to define an explicit well-ordering which intuitively *should* have order-type $\omega + \omega$. Indeed, we gave an informal example of this in section 66.2, when we presented the ordering on the natural numbers given by:

$$n \lessdot m \text{ iff either } n < m \text{ and } m - n \text{ is even,}$$
$$\text{or } n \text{ is even and } m \text{ is odd.}$$

But if $\omega + \omega$ does not exist, this well-ordering is not isomorphic to any ordinal. So **Z** does *not* prove Theorem 66.26.

Flipping things around: Replacement allows us to prove the existence of $\omega + \omega$, and hence must allow us to prove the existence of $V_{\omega+\omega}$. And not just that. For *any* well-ordering we can define, Theorem 66.26 tells us that there is some α isomorphic with that well-ordering, and hence that V_α exists. In a straightforward way, then, Replacement guarantees that the hierarchy of sets must be *very tall*.

Over the next few sections, and then again in section 71.5, we'll get a better sense of better just *how* tall Replacement forces the hierarchy to be. The simple point, for now, is that Replacement really *does* stand in need of justification!

---

content/set-theory/replacement/extrinsic.tex

68.3 Extrinsic Considerations about Replacement

sth:replacement:extrinsic:sec

We start by considering an *extrinsic* attempt to justify Replacement. Boolos suggests one, as follows.

> [...] the reason for adopting the axioms of replacement is quite simple: they have many desirable consequences and (apparently) no undesirable ones. In addition to theorems about the iterative conception, the consequences include a satisfactory if not ideal theory of infinite numbers, and a highly desirable result that justifies inductive definitions on well-founded relations. (Boolos, 1971, 229)

The gist of Boolos's idea is that we should justify Replacement by its fruits. And the specific fruits he mentions are the things we have discussed in the past few chapters. Replacement allowed us to prove that the von Neumann ordinals were excellent surrogates for the idea of a well-ordering type (this is our "satisfactory if not ideal theory of infinite numbers"). Replacement

also allowed us to define the V_αs, establish the notion of rank, and prove $\in$-Induction (this amounts to our "theorems about the iterative conception"). Finally, Replacement allows us to prove the Transfinite Recursion Theorem (this is the "inductive definitions on well-founded relations").

These are, indeed, desirable consequences. But do these desirable consequences suffice to *justify* Replacement? *No.* Or at least, not straightforwardly.

Here is a simple problem. Whilst we have stated some desirable consequences of Replacement, we could have obtained many of them via other means. This is not as well known as it ought to be, though, so we should pause to explain the situation.

There is a simple theory of sets, Level Theory, or **LT** for short.[1] **LT**'s axioms are just Extensionality, Separation, and the claim that every set is a subset of some *level*, where "level" is cunningly defined so that the levels behave like our friends, the V_αs. So **ZF** proves **LT**; but **LT** is *much* weaker than **ZF**. In fact, **LT** does not give you Pairs, Powersets, Infinity, or Replacement. Let **Zr** be the result of adding Infinity and Powersets to **LT**; this delivers Pairs too, so, **Zr** is at least as strong as **Z**. But, in fact, **Zr** is strictly stronger than **Z**, since it adds the claim that every set has a rank (hence my suggestion that we call it **Zr**). Indeed, **Zr** delivers: a perfectly satisfactory theory of ordinals; results which stratify the hierarchy into well-ordered stages; a proof of $\in$-Induction; and a *version* of Transfinite Recursion.

In short: although Boolos didn't know this, all of the desirable consequences which he mentions could have been arrived at *without* Replacement; he simply needed to use **Zr** rather than **Z**.

(Given all of this, why did we follow the conventional route, of teaching you **ZF**, rather than **LT** and **Zr**? There are two reasons. First: for purely historical reasons, starting with **LT** is rather nonstandard; we wanted to equip you to be able to read more standard discussions of set theory. Second: when you are ready to appreciate **LT** and **Zr**, you can simply read Potter 2004 and Button 2021.)

Of course, since **Zr** is strictly weaker than **ZF**, there are results which **ZF** proves which **Zr** leaves open. So one could try to justify Replacement on extrinsic grounds by pointing to one of these results. But, once you know how to use **Zr**, it is quite hard to find many examples of things that are (a) settled by Replacement but not otherwise, and (b) are intuitively true. (For more on this, see Potter 2004, §13.2.)

The bottom line is this. To provide a compelling extrinsic justification for Replacement, one would need to find a result which *cannot* be achieved without Replacement. And that's not an easy enterprise.

Let's consider a further problem which arises for any attempt to offer a purely extrinsic justification for Replacement. (This problem is perhaps more fundamental than the first.) Boolos does not just point out that Replace-

[1] The first versions of **LT** are offered by Montague (1965) and Scott (1974); this was simplified, and given a book-length treatment, by Potter (2004); and Button (2021) has recently simplified **LT** further.

ment has many desirable consequences. He also states that Replacement has "(apparently) no undesirable" consequences. But this parenthetical caveat, "apparently," is surely absolutely crucial.

Recall how we ended up here: Naïve Comprehension ran into inconsistency, and we responded to this inconsistency by embracing the cumulative-iterative conception of set. This conception comes equipped with a story which, we hope, assures us of its consistency. But if we cannot justify Replacement from within that story, then we have (as yet) no reason to believe that **ZF** is consistent. Or rather: we have no reason to believe that **ZF** is consistent, apart from the (perhaps merely contingent) fact that no one has discovered a contradiction *yet*. In exactly that sense, Boolos's comment seems to come down to this: "(apparently) **ZF** is consistent". We should demand greater reassurance of consistency than this.

This issue will affect any *purely* extrinsic attempt to justify Replacement, i.e., any justification which is couched solely in terms of the (known) consequences of **ZF**. As such, we will want to look for an *intrinsic* justification of Replacement, i.e., a justification which suggests that the story which we told about sets somehow "already" commits us to Replacement.

---

content/set-theory/replacement/limofsize.tex

68.4 Limitation-of-size

sth:replacement:limofsize:sec

Perhaps the most common attempt to offer an "intrinsic" justification of Replacement comes via the following notion:

> *Limitation-of-size.* Any things form a set, provided that there are not too many of them.

This principle will immediately vindicate Replacement. After all, any set formed by Replacement cannot be any larger than any set from which it was formed. Stated precisely: suppose you form a set $\tau[A] = \{\tau(x) : x \in A\}$ using Replacement; then $\tau[A] \preceq A$; so if the elements of A were not too numerous to form a set, their images are not too numerous to form $\tau[A]$.

The obvious difficulty with invoking *Limitation-of-size* to justify Replacement is that we have *not* yet laid down any principle like *Limitation-of-size*. Moreover, when we told our story about the cumulative-iterative conception of set in chapters 64 to 65, nothing ever *hinted* in the direction of *Limitation-of-size*. This, indeed, is precisely why Boolos at one point wrote: "Perhaps one may conclude that there are at least two thoughts 'behind' set theory" (1989, p. 19). On the one hand, the ideas surrounding the cumulative-iterative conception of set are meant to vindicate **Z**. On the other hand, *Limitation-of-size* is meant to vindicate Replacement.

But the issue it is not just that we have thus far been *silent* about *Limitation-of-size*. Rather, the issue is that *Limitation-of-size* (as just formulated) seems

to sit quite badly with the cumulative-iterative notion of set. After all, it mentions nothing about the idea of sets as formed in *stages*.

This is really not much of a surprise, given the history of these "two thoughts" (i.e., the cumulative-iterative conception of set, and *Limitation-of-size*). These "two thoughts" ultimately amount to two rather different projects for blocking the set-theoretic paradoxes. The cumulative-iterative notion of set blocks Russell's paradox by saying, roughly: *we should never have expected a Russell set to exist, because it would not be "formed" at any stage*. By contrast, *Limitation-of-size* is meant to rule out the Russell set, by saying, roughly: *we should never have expected a Russell set to exist, because it would have been too big*.

Put like this, then, let's be blunt: considered as a reply to the paradoxes, *Limitation-of-size* stands in need of *much* more justification. Consider, for example, this version of Russell's Paradox: *no pug sniffs exactly the pugs which don't sniff themselves* (see section 64.2). If you ask "why is there no such pug?", it is not a good answer to be told that such a pug would have to sniff too many pugs. So why would it be a good intuitive explanation, of the non-existence of a Russell set, that it would have to be "too big" to exist?

In short, it's forgivable if you are a bit mystified concerning the "intuitive" motivation for *Limitation-of-size*.

---

content/set-theory/replacement/absinf.tex

68.5 Replacement and "Absolute Infinity"

sth:replacement:absinf:sec

We will now put *Limitation-of-size* behind us, and explore a different family of (intrinsic) attempts to justify Replacement, which do take seriously the idea of the sets as formed in stages.

When we first outlined the iterative process, we offered some principles which explained what happens at each stage. These were *Stages-are-key*, *Stages-are-ordered*, and *Stages-accumulate*. Later, we added some principles which told us something about the number of stages: *Stages-keep-going* told us that the process of set-formation never ends, and *Stages-hit-infinity* told us that the process goes through an infinite-th stage.

It is reasonable to suggest that these two latter principles fall out of some a broader principle, like:

> *Stages-are-inexhaustible*. There are absolutely infinitely many stages; the hierarchy is as tall as it could possibly be.

Obviously this is an informal principle. But even if it is not immediately *entailed* by the cumulative-iterative conception of set, it certainly seems *consonant* with it. At the very least, and unlike *Limitation-of-size*, it retains the idea that sets are formed stage-by-stage.

The hope, now, is to leverage *Stages-are-inexhaustible* into a justification of Replacement. So let us see how this might be done.

In section 66.2, we saw that it is easy to construct a well-ordering which (morally) should be isomorphic to $\omega+\omega$. Otherwise put, we can easily imagine a stage-by-stage iterative process, whose order-type (morally) is $\omega+\omega$. As such, if we have accepted *Stages-are-inexhaustible*, then we should surely accept that there is at least an $\omega+\omega$-th stage of the hierarchy, i.e., $V_{\omega+\omega}$, for the hierarchy surely *could* continue thus far.

This thought generalizes as follows: for any well-ordering, the process of building the iterative hierarchy should run at least as far as that well-ordering. And we could guarantee this, just by treating Theorem 66.26 as an *axiom*. This would tell us that any well-ordering is isomorphic to a von Neumann ordinal. Since each von Neumann ordinal will be equal to its own rank, Theorem 66.26 will then tell us that, whenever we can describe a well-ordering in our set theory, the iterative process of set building must outrun that well-ordering.

This idea certainly seems like a corollary of *Stages-are-inexhaustible*. Unfortunately, if our aim is to extract Replacement from this idea, then we face a simple, technical, barrier: Replacement is strictly stronger than Theorem 66.26. (This observation is made by Potter (2004, §13.2); we will prove it in section 68.8.)

The upshot is that, if we are going to understand *Stages-are-inexhaustible* in such a way as to yield Replacement, then it cannot *merely* say that the hierarchy outruns any well-ordering. It must make a stronger claim than that. To this end, Shoenfield (1977) proposed a very natural strengthening of the idea, as follows: the hierarchy is not *cofinal* with any set.[2] In slightly more detail: if τ is a mapping which sends sets to stages of the hierarchy, the image of any set A under τ does not exhaust the hierarchy. Otherwise put (schematically):

> *Stages-are-super-cofinal.* If A is a set and $\tau(x)$ is a stage for every $x \in A$, then there is a stage which comes after each $\tau(x)$ for $x \in A$.

It is obvious that **ZF** proves a suitably formalised version of *Stages-are-super-cofinal*. Conversely, we can informally argue that *Stages-are-super-cofinal* justifies Replacement.[3] For suppose $(\forall x \in A)\exists! y\, \varphi(x,y)$. Then for each $x \in A$, let $\sigma(x)$ be the y such that $\varphi(x,y)$, and let $\tau(x)$ be the stage at which $\sigma(x)$ is first formed. By *Stages-are-super-cofinal*, there is a stage V such that $(\forall x \in A)\tau(x) \in V$. Now since each $\tau(x) \in V$ and $\sigma(x) \subseteq \tau(x)$, by Separation we can obtain $\{y \in V : (\exists x \in A)\sigma(x) = y\} = \{y : (\exists x \in A)\varphi(x,y)\}$.

Problem 68.1. Formalize *Stages-are-super-cofinal* within **ZF**.

So *Stages-are-super-cofinal* vindicates Replacement. And it is at least plausible that *Stages-are-inexhaustible* vindicates *Stages-are-super-cofinal*. For suppose *Stages-are-super-cofinal* fails. So the hierarchy is cofinal with some set A,

[2] Gödel seems to have proposed a similar thought; see Potter (2004, p. 223). For discussion of Gödel and Shoenfield, see Incurvati (2020, 90–5).

[3] It would be harder to prove Replacement using some formalisation of *Stages-are-super-cofinal*, since **Z** on its own is not strong enough to define the stages, so it is not clear how one would formalise *Stages-are-super-cofinal*. One option, though, is to work in some extension of **LT**, as discussed in section 68.3.

i.e., we have a map τ such that for any stage S there is some $x \in A$ such that $S \in \tau(x)$. In that case, we do have a way to get a handle on the supposed "absolute infinity" of the hierarchy: it is *exhausted* by the range of τ applied to A. And that compromises the thought that the hierarchy is "absolutely infinite". Contraposing: *Stages-are-inexhaustible* entails *Stages-are-super-cofinal*, which in turn justifies Replacement.

This represents a genuinely promising attempt to provide an intrinsic justification for Replacement. But whether it ultimately works, or not, we will have to leave to you to decide.

---

content/set-theory/replacement/ref.tex

68.6 Replacement and Reflection

Our last attempt to justify Replacement, via *Stages-are-inexhaustible*, begins with a deep and lovely result:[4] sth:replacement:ref:sec

Theorem 68.2 (Reflection Schema). *For any formula φ:* *sth:replacement:ref:reflectionschema*

$$\forall\alpha\exists\beta > \alpha(\forall x_1 \ldots, x_n \in V_\beta)(\varphi(x_1, \ldots, x_n) \leftrightarrow \varphi^{V_\beta}(x_1, \ldots, x_n))$$

As in Definition 68.1, φ^{V_β} is the result of restricting every quantifier in φ to the set V_β. So, intuitively, Reflection says this: if φ is true in the entire hierarchy, then φ is true in arbitrarily many *initial segments* of the hierarchy.

Montague (1961) and Lévy (1960) showed that (suitable formulations of) Replacement and Reflection are equivalent, modulo **Z**, so that adding either gives you **ZF**. (We prove these results in section 68.7.) Given this equivalence, one might hope to justify Reflection and Replacement via *Stages-are-inexhaustible* as follows: given *Stages-are-inexhaustible*, the hierarchy should be very, very tall; so tall, in fact, that nothing we can say about it is sufficient to bound its height. And we can understand this as the thought that, if any sentence φ is true in the entire hierarchy, then it is true in arbitrarily many initial segments of the hierarchy. And that is just Reflection.

Again, this seems like a genuinely promising attempt to provide an intrinsic justification for Replacement. But there is much too much to say about it here. You must now decide for yourself whether it succeeds.[5]

---

content/set-theory/replacement/refproofs.tex

68.7 Appendix: Results surrounding Replacement

In this section, we will prove Reflection within **ZF**. We will also prove a sense in which Reflection is equivalent to Replacement. And we will prove an interesting sth:replacement:refproofs:sec

[4] A reminder: all formulas can have parameters (unless explicitly stated otherwise).

[5] Though you might like to continue by reading Incurvati (2020, 95–100).

consequence of all this, concerning the strength of Reflection/Replacement. *Warning: this is easily the most advanced bit of mathematics in this textbook.*

We'll start with a lemma which, for brevity, employs the notational device of *overlining* to deal with sequences of variables or objects. So: "$\overline{a}_k$" abbreviates "$a_{k_1}, \ldots, a_{k_n}$", where n is determined by context.

sth:replacement:refproofs: lemreflection

Lemma 68.3. *For each $1 \leq i \leq k$, let $\varphi_i(\overline{v}_i, x)$ be a formula. Then for each α there is some $\beta > \alpha$ such that, for any $\overline{a}_1, \ldots, \overline{a}_k \in V_\beta$ and each $1 \leq i \leq k$:*

$$\exists x \varphi_i(\overline{a}_i, x) \to (\exists x \in V_\beta)\varphi_i(\overline{a}_i, x)$$

Proof. We define a term μ as follows: $\mu(\overline{a}_1, \ldots, \overline{a}_k)$ is the least stage, V, which satisfies all of the following conditionals, for $1 \leq i \leq k$:

$$\exists x \varphi_i(\overline{a}_i, x) \to (\exists x \in V)\varphi_i(\overline{a}_i, x))$$

It is easy to confirm that $\mu(\overline{a}_1, \ldots, \overline{a}_k)$ exists for all $\overline{a}_1, \ldots, \overline{a}_k$. Now, using Replacement and our recursion theorem, define:

$$\begin{aligned} S_0 &= V_{\alpha+1} \\ S_{n+1} &= S_n \cup \bigcup \{\mu(\overline{a}_1, \ldots, \overline{a}_k) : \overline{a}_1, \ldots, \overline{a}_k \in S_n\} \\ S &= \bigcup_{m<\omega} S_n. \end{aligned}$$

Each S_n, and hence S itself, is a stage after V_α. Now fix $\overline{a}_1, \ldots, \overline{a}_k \in S$; so there is some $n < \omega$ such that $\overline{a}_1, \ldots, \overline{a}_k \in S_n$. Fix some $1 \leq i \leq k$, and suppose that $\exists x \varphi_i(\overline{a}_i, x)$. So $(\exists x \in \mu(\overline{a}_1, \ldots, \overline{a}_k))\varphi_i(\overline{a}_i, x)$ by construction, so $(\exists x \in S_{n+1})\varphi_i(\overline{a}_i, x)$ and hence $(\exists x \in S)\varphi_i(\overline{a}_i, x)$. So S is our V_β. □

We can now prove Theorem 68.2 quite straightforwardly:

Proof. Fix α. Without loss of generality, we can assume φ's only connectives are $\exists$, $\neg$ and $\wedge$ (since these are expressively adequate). Let $\psi_1, \ldots, \psi_k$ enumerate each of φ's subformulas according to complexity, so that $\psi_k = \varphi$. By Lemma 68.3, there is a $\beta > \alpha$ such that, for any $\overline{a}_i \in V_\beta$ and each $1 \leq i \leq k$:

$$\exists x \psi_i(\overline{a}_i, x) \to (\exists x \in V_\beta)\psi_i(\overline{a}_i, x) \qquad (*)$$

By induction on complexity of ψ_i, we will show that $\psi_i(\overline{a}_i) \leftrightarrow \psi_i^{V_\beta}(\overline{a}_i)$, for any $\overline{a}_i \in V_\beta$. If ψ_i is atomic, this is trivial. The biconditional also establishes that, when ψ_i is a negation or conjunction of subformulas satisfying this property, ψ_i itself satisfies this property. So the only interesting case concerns quantification. Fix $\overline{a}_i \in V_\beta$; then:

$$\begin{aligned} (\exists x \psi_i(\overline{a}_i, x))^{V_\beta} &\text{ iff } (\exists x \in V_\beta)\psi_i^{V_\beta}(\overline{a}_i, x) && \text{by definition} \\ &\text{iff } (\exists x \in V_\beta)\psi_i(\overline{a}_i, x) && \text{by hypothesis} \\ &\text{iff } \exists x \psi_i(\overline{a}_i, x) && \text{by } (*) \end{aligned}$$

This completes the induction; the result follows as $\psi_k = \varphi$. □

We have proved Reflection in **ZF**. Our proof essentially followed Montague (1961). We now want to prove in **Z** that Reflection entails Replacement. The proof follows Lévy (1960), but with a simplification.

Since we are working in **Z**, we cannot present Reflection in exactly the form given above. After all, we formulated Reflection using the "V_α" notation, and that cannot be defined in **Z** (see section 67.5). So instead we will offer an apparently weaker formulation of Replacement, as follows:

> *Weak-Reflection.* For any formula φ, there is a transitive set S such that 0, 1, and any parameters to φ are elements of S, and $(\forall \overline{x} \in S)(\varphi \leftrightarrow \varphi^S)$.

To use this to prove Replacement, we will first follow Lévy (1960, first part of Theorem 2) and show that we can "reflect" two formulas at once:

Lemma 68.4 (in Z + Weak-Reflection.). *For any formulas ψ, χ, there is a transitive set S such that 0 and 1 (and any parameters to the formulas) are elements of S, and $(\forall \overline{x} \in S)((\psi \leftrightarrow \psi^S) \wedge (\chi \leftrightarrow \chi^S))$.*

sth:replacement:refproofs:lem:reflect

Proof. Let φ be the formula $(z = 0 \wedge \psi) \vee (z = 1 \wedge \chi)$.

Here we use an abbreviation; we should spell out "$z = 0$" as "$\forall t\, t \notin z$" and "$z = 1$" as "$\forall s(s \in z \leftrightarrow \forall t\, t \notin s)$". But since $0, 1 \in S$ and S is transitive, these formulas are *absolute* for S; that is, they will apply to the same object whether we restrict their quantifiers to S.[6]

By Weak-Reflection, we have some appropriate S such that:

$$\begin{aligned}
&(\forall z, \overline{x} \in S)(\varphi \leftrightarrow \varphi^S)\\
\text{i.e. } &(\forall z, \overline{x} \in S)(((z = 0 \wedge \psi) \vee (z = 1 \wedge \chi)) \leftrightarrow\\
&\qquad ((z = 0 \wedge \psi) \vee (z = 1 \wedge \chi))^S)\\
\text{i.e. } &(\forall z, \overline{x} \in S)(((z = 0 \wedge \psi) \vee (z = 1 \wedge \chi)) \leftrightarrow\\
&\qquad ((z = 0 \wedge \psi^S) \vee (z = 1 \wedge \chi^S)))\\
\text{i.e. } &(\forall \overline{x} \in S)((\psi \leftrightarrow \psi^S) \wedge (\chi \leftrightarrow \chi^S))
\end{aligned}$$

The second claim entails the third because "$z = 0$" and "$z = 1$" are absolute for S; the fourth claim follows since $0 \neq 1$. □

We can now obtain Replacement, just by following and simplifying Lévy (1960, Theorem 6):

Theorem 68.5 (in Z + Weak-Reflection). *For any formula $\varphi(v, w)$, and any A, if $(\forall x \in A)\exists! y\, \varphi(x, y)$, then $\{y : (\exists x \in A)\varphi(x, y)\}$ exists.*

[6] More formally, letting ξ be either of these formulas, $\xi(z) \leftrightarrow \xi^S(z)$.

Proof. Fix A such that $(\forall x \in A)\exists! y\, \varphi(x,y)$, and define formulas:

$$\psi \text{ is } (\varphi(x,z) \wedge A = A)$$
$$\chi \text{ is } \exists y\, \varphi(x,y)$$

Using Lemma 68.4, since A is a parameter to ψ, there is a transitive S such that $0, 1, A \in S$ (along with any other parameters), and such that:

$$(\forall x, z \in S)((\psi \leftrightarrow \psi^S) \wedge (\chi \leftrightarrow \chi^S))$$

So in particular:

$$(\forall x, z \in S)(\varphi(x,z) \leftrightarrow \varphi^S(x,z))$$
$$(\forall x \in S)(\exists y \varphi(x,y) \leftrightarrow (\exists y \in S)\varphi^S(x,y))$$

Combining these, and observing that $A \subseteq S$ since $A \in S$ and S is transitive:

$$(\forall x \in A)(\exists y \varphi(x,y) \leftrightarrow (\exists y \in S)\varphi(x,y))$$

Now $(\forall x \in A)(\exists! y \in S)\varphi(x,y)$, because $(\forall x \in A)\exists! y\, \varphi(x,y)$. Now Separation yields $\{y \in S : (\exists x \in A)\varphi(x,y)\} = \{y : (\exists x \in A)\varphi(x,y)\}$. □

---

content/set-theory/replacement/finiteaxiomatizability.tex

68.8 Appendix: Finite axiomatizability

lacement:finiteaxiomatizability: sec

We close this chapter by extracting some results from Replacement. The first result is due to Montague (1961); note that it is not a proof *within* **ZF**, but a proof *about* **ZF**:

eplacement:finiteaxiomatizability: zfnotfinitely

Theorem 68.6. **ZF** *is not finitely axiomatizable. More generally: if* $\mathbf{T}$ *is finite and* $\mathbf{T} \vdash \mathbf{ZF}$*, then* $\mathbf{T}$ *is inconsistent.*

(Here, we tacitly restrict ourselves to first-order sentences whose only non-logical primitive is $\in$*, and we write* $\mathbf{T} \vdash \mathbf{ZF}$ *to indicate that* $\mathbf{T} \vdash \varphi$ *for all* $\varphi \in \mathbf{ZF}$*.)*

Proof. Fix finite $\mathbf{T}$ such that $\mathbf{T} \vdash \mathbf{ZF}$. So, $\mathbf{T}$ proves Reflection, i.e. Theorem 68.2. Since $\mathbf{T}$ is finite, we can rewrite it as a single conjunction, θ. Reflecting with this formula, $\mathbf{T} \vdash \exists \beta(\theta \leftrightarrow \theta^{V_\beta})$. Since trivially $\mathbf{T} \vdash \theta$, we find that $\mathbf{T} \vdash \exists \beta\ \theta^{V_\beta}$.

Now, let $\psi(X)$ abbreviate:

$$\theta^X \wedge X \text{ is transitive} \wedge (\forall Y \in X)(Y \text{ is transitive} \to \neg\theta^Y)$$

roughly this says: X is a transitive model of θ, and $\in$-minimal in this regard. Now, recalling that $\mathbf{T} \vdash \exists \beta\ \theta^{V_\beta}$, by basic facts about ranks within **ZF** and hence within $\mathbf{T}$, we have:

$$\mathbf{T} \vdash \exists M \psi(M). \tag{*}$$

Using the first conjunct of $\psi(X)$, whenever $\mathbf{T} \vdash \sigma$, we have that $\mathbf{T} \vdash \forall X(\psi(X) \rightarrow \sigma^X)$. So, by (*):

$$\mathbf{T} \vdash \forall X(\psi(X) \rightarrow (\exists N \psi(N))^X)$$

Using this, and (*) again:

$$\mathbf{T} \vdash \exists M(\psi(M) \wedge (\exists N \psi(N))^M)$$

In particular, then:

$$\mathbf{T} \vdash \exists M(\psi(M) \wedge (\exists N \in M)((N \text{ is transitive})^N \wedge (\theta^N)^M))$$

So, by elementary reasoning concerning transitivity:

$$\mathbf{T} \vdash \exists M(\psi(M) \wedge (\exists N \in M)(N \text{ is transitive} \wedge \theta^N))$$

So that $\mathbf{T}$ is inconsistent.[7] □

Here is a similar result, noted by Potter (2004, 223):

Proposition 68.7. *Let* $\mathbf{T}$ *extend* $\mathbf{Z}$ *with finitely many new axioms. If* $\mathbf{T} \vdash \mathbf{ZF}$*, then* $\mathbf{T}$ *is inconsistent. (Here we use the same tacit restrictions as for Theorem 68.6.)* sth:replacement:finiteaxiomatizab finiteextensionofZ

Proof. Use θ for the conjunction of all of $\mathbf{T}$'s axioms *except* for the (infinitely many) instances of Separation. Defining ψ from θ as in Theorem 68.6, we can show that $\mathbf{T} \vdash \exists M \psi(M)$.

As in Theorem 68.6, we can establish the schema that, whenever $\mathbf{T} \vdash \sigma$, we have that $\mathbf{T} \vdash \forall X(\psi(X) \rightarrow \sigma^X)$. We then finish our proof, exactly as in Theorem 68.6.

However, establishing the schema involves a little more work than in Theorem 68.6. After all, the Separation-instances are in $\mathbf{T}$, but they are not conjuncts of θ. However, we can overcome this obstacle by proving that $\mathbf{T} \vdash \forall X(X \text{ is transitive} \rightarrow \sigma^X)$, for every Separation-instance σ. We leave this to the reader. □

Problem 68.2. Show that, for every Separation-instance σ, we have: $\mathbf{Z} \vdash \forall X(X \text{ is transitive} \rightarrow \sigma^X)$. (We used this schema in Proposition 68.7.)

Problem 68.3. Show that, for every $\varphi \in \mathbf{Z}$, we have $\mathbf{ZF} \vdash \varphi^{V_{\omega+\omega}}$.

Problem 68.4. Confirm the remaining schematic results invoked in the proofs of Theorem 68.6 and Proposition 68.7.

[7] This "elementary reasoning" involves proving certain "absoluteness facts" for transitive sets.

As remarked in section 68.5, this shows that Replacement is strictly stronger than Theorem 66.26. Or, slightly more strictly: if **Z** + "every well-ordering is isomorphic to a unique ordinal" is consistent, then it fails to prove some Replacement-instance.

Chapter 69

Ordinal Arithmetic

---

content/set-theory/ord-arithmetic/introduction.tex

69.1 Introduction

sth:ord-arithmetic:intro:sec

In chapter 66, we developed a theory of ordinal numbers. We saw in chapter 67 that we can think of the ordinals as a spine around which the remainder of the hierarchy is constructed. But that is not the only role for the ordinals. There is also the task of performing ordinal arithmetic.

We already gestured at this, back in section 66.2, when we spoke of ω, $\omega+1$ and $\omega+\omega$. At the time, we spoke informally; the time has come to spell it out properly. However, we should mention that there is not much philosophy in this chapter; just technical developments, coupled with a (mildly) interesting observation that we can do the same thing in two different ways.

---

content/set-theory/ord-arithmetic/addition.tex

69.2 Ordinal Addition

sth:ord-arithmetic:add:sec

Suppose we want to add α and β. We can simply put a copy of β immediately after a copy of α. (We need to take *copies*, since we know from Proposition 66.22 that either $\alpha \subseteq \beta$ or $\beta \subseteq \alpha$.) The intuitive effect of this is to run through an α-sequence of steps, *and then* to run through a β-sequence. The resulting sequence will be well-ordered; so by Theorem 66.26 it is isomorphic to a (unique) ordinal. That ordinal can be regarded as the *sum* of α and β.

That is the intuitive idea behind ordinal addition. To define it rigorously, we start with the idea of taking *copies* of sets. The idea here is to use arbitrary tags, 0 and 1, to keep track of which object came from where:

Definition 69.1. The *disjoint sum* of A and B is $A \sqcup B = (A \times \{0\}) \cup (B \times \{1\})$. sth:ord-arithmetic:add:defdissum

We next define an ordering on pairs of ordinals:

Definition 69.2. For any ordinals $\alpha_1, \alpha_2, \beta_1, \beta_2$, say that:

$$\langle \alpha_1, \alpha_2 \rangle \lhd \langle \beta_1, \beta_2 \rangle \text{ iff either } \alpha_2 \in \beta_2 \text{ or both } \alpha_2 = \beta_2 \text{ and } \alpha_1 \in \beta_1$$

This is a *reverse lexicographic* ordering, since you order by the second element, then by the first. Now recall that we wanted to define $\alpha + \beta$ as the order type of a copy of α followed by a copy of β. To achieve that, we say:

Definition 69.3. For any ordinals α, β, their sum is $\alpha + \beta = \mathrm{ord}(\alpha \sqcup \beta, \lhd)$. sth:ord-arithmetic:add:defordplus

Note that we slightly abused notation here; strictly we should write "$\{\langle x, y \rangle \in \alpha \sqcup \beta : x \lhd y\}$" in place of "$\lhd$". For brevity, though, we will continue to abuse notation in this way in what follows.

The following result, together with Theorem 66.26, confirms that our definition is well-formed:

Lemma 69.4. *$\langle \alpha \sqcup \beta, \lhd \rangle$ is a well-order, for any ordinals α and β.* sth:ord-arithmetic:add:ordsumlessiswo

Proof. Obviously $\lhd$ is connected on $\alpha \sqcup \beta$. To show it is well-founded, fix a non-empty $X \subseteq \alpha \sqcup \beta$. Let Y be the subset of X whose second coordinate is as small as possible, i.e. $Y = \{\langle \gamma, i \rangle \in X : (\forall \langle \delta, j \rangle \in X) i \leq j\}$. Now choose the element of Y with smallest first coordinate. □

So we have a nice, explicit definition of ordinal addition. Here is an unsurprising fact (recall that $1 = \{0\}$, by Definition 65.7):

Proposition 69.5. *$\alpha + 1 = \alpha^+$, for any ordinal α.*

Proof. Consider the isomorphism f from $\alpha^+ = \alpha \cup \{\alpha\}$ to $\alpha \sqcup 1 = (\alpha \times \{0\}) \sqcup (\{0\} \times \{1\})$ given by $f(\gamma) = \langle \gamma, 0 \rangle$ for $\gamma \in \alpha$, and $f(\alpha) = \langle 0, 1 \rangle$. □

Moreover, it is easy to show that addition obeys certain recursive conditions:

Lemma 69.6. *For any ordinals α, β, we have:* sth:ord-arithmetic:add:ordadditionrecursion

$$\begin{aligned} \alpha + 0 &= \alpha \\ \alpha + (\beta + 1) &= (\alpha + \beta) + 1 \\ \alpha + \beta &= \operatorname{lsub}_{\delta < \beta}(\alpha + \delta) \qquad \text{if } \beta \text{ is a limit ordinal} \end{aligned}$$

Proof. We check case-by-case; first:

$$\begin{aligned}\alpha + 0 &= \mathrm{ord}((\alpha \times \{0\}) \cup (0 \times \{1\}), \lhd) \\ &= \mathrm{ord}((\alpha \times \{0\}) \cup \{0\}, \lhd) \\ &= \alpha \\ \alpha + (\beta + 1) &= \mathrm{ord}((\alpha \times \{0\}) \cup (\beta^+ \times \{1\}), \lhd) \\ &= \mathrm{ord}((\alpha \times \{0\}) \cup (\beta \times \{1\}), \lhd) + 1 \\ &= (\alpha + \beta) + 1\end{aligned}$$

Now let $\beta \neq \emptyset$ be a limit. If $\delta < \beta$ then also $\delta + 1 < \beta$, so $\alpha + \delta$ is a proper initial segment of $\alpha + \beta$. So $\alpha + \beta$ is a strict upper bound on $X = \{\alpha + \delta : \delta < \beta\}$. Moreover, if $\alpha \leq \gamma < \alpha + \beta$, then clearly $\gamma = \alpha + \delta$ for some $\delta < \beta$. So $\alpha + \beta = \mathrm{lsub}_{\delta<\beta}(\alpha + \delta)$. □

But here is a striking fact. To define ordinal addition, we could *instead* have simply used the Transfinite Recursion Theorem, and laid down the recursion equations, exactly as given in Lemma 69.6 (though using "β^+" rather than "$\beta + 1$").

There are, then, two different ways to define operations on the ordinals. We can define them *synthetically*, by explicitly constructing a well-ordered set and considering its order type. Or we can define them *recursively*, just by laying down the recursion equations. Done correctly, though, the outcome is identical. For Theorem 66.26 guarantees that these recursion equations pin down *unique* ordinals.

In many ways, ordinal arithmetic behaves just like addition of the natural numbers. For example, we can prove the following:

sth:ord-arithmetic:add:ordinaladditionisnice
sth:ord-arithmetic:add:ordaddition1
sth:ord-arithmetic:add:ordaddition2
sth:ord-arithmetic:add:ordaddition3
sth:ord-arithmetic:add:ordaddition4

Lemma 69.7. *If α, β, γ are ordinals, then:*

1. *if $\beta < \gamma$, then $\alpha + \beta < \alpha + \gamma$*
2. *if $\alpha + \beta = \alpha + \gamma$, then $\beta = \gamma$*
3. *$\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$, i.e., addition is associative*
4. *If $\alpha \leq \beta$, then $\alpha + \gamma \leq \beta + \gamma$*

Proof. We prove (3), leaving the rest as an exercise. The proof is by Simple Transfinite Induction on γ, using Lemma 69.6. When $\gamma = 0$:

$$(\alpha + \beta) + 0 = \alpha + \beta = \alpha + (\beta + 0)$$

When $\gamma = \delta + 1$, suppose for induction that $(\alpha + \beta) + \delta = \alpha + (\beta + \delta)$; now using Lemma 69.6 three times:

$$\begin{aligned}(\alpha + \beta) + (\delta + 1) &= ((\alpha + \beta) + \delta) + 1 \\ &= (\alpha + (\beta + \delta)) + 1 \\ &= \alpha + ((\beta + \delta) + 1) \\ &= \alpha + (\beta + (\delta + 1))\end{aligned}$$

When γ is a limit ordinal, suppose for induction that if $\delta \in \gamma$ then $(\alpha+\beta)+\delta = \alpha+(\beta+\delta)$; now:

$$\begin{aligned}(\alpha+\beta)+\gamma &= \operatorname*{lsub}_{\delta<\gamma}((\alpha+\beta)+\delta)\\ &= \operatorname*{lsub}_{\delta<\gamma}(\alpha+(\beta+\delta))\\ &= \alpha+\operatorname*{lsub}_{\delta<\gamma}(\beta+\delta)\\ &= \alpha+(\beta+\gamma)\end{aligned}$$

□

Problem 69.1. Prove the remainder of Lemma 69.7.

In these ways, ordinal addition should be very familiar. But, there is a crucial way in which ordinal addition is *not* like addition on the natural numbers.

Proposition 69.8. *Ordinal addition is not commutative;* $1+\omega = \omega < \omega+1$. *sth:ord-arithmetic:add:ordsumnotcommute*

Proof. Note that $1+\omega = \operatorname{lsub}_{n<\omega}(1+n) = \omega \in \omega \cup \{\omega\} = \omega^+ = \omega+1$. □

Whilst this may initially come as a surprise, *it shouldn't*. On the one hand, when you consider $1+\omega$, you are thinking about the order type you get by putting an extra element *before* all the natural numbers. Reasoning as we did with Hilbert's Hotel in section 6.1, intuitively, this extra first element shouldn't make any difference to the overall order type. On the other hand, when you consider $\omega+1$, you are thinking about the order type you get by putting an extra element *after* all the natural numbers. And that's a radically different beast!

---

content/set-theory/ord-arithmetic/using-addition.tex

69.3 Using Ordinal Addition

Using addition on the ordinals, we can explicitly calculate the ranks of various sets, in the sense of Definition 67.15: *sth:ord-arithmetic:using-additionsec*

Lemma 69.9. *If* $\operatorname{rank}(A) = \alpha$ *and* $\operatorname{rank}(B) = \beta$*, then:* *sth:ord-arithmetic:using-additionrankcomputation*

1. $\operatorname{rank}(\wp(A)) = \alpha+1$ *sth:ord-arithmetic:using-additionexrankpow*
2. $\operatorname{rank}(\{A,B\}) = \max(\alpha,\beta)+1$ *sth:ord-arithmetic:using-additionexrankpair*
3. $\operatorname{rank}(A \cup B) = \max(\alpha,\beta)$ *sth:ord-arithmetic:using-additionexrankcup*
4. $\operatorname{rank}(\langle A,B\rangle) = \max(\alpha,\beta)+2$ *sth:ord-arithmetic:using-additionexranktuple*
5. $\operatorname{rank}(A \times B) \leq \max(\alpha,\beta)+2$ *sth:ord-arithmetic:using-additionexranktimes*
6. $\operatorname{rank}(\bigcup A) = \alpha$ *when* α *is empty or a limit;* $\operatorname{rank}(\bigcup A) = \gamma$ *when* $\alpha = \gamma+1$ *sth:ord-arithmetic:using-additionexrankunion*

Proof. Throughout, we invoke Proposition 67.20 repeatedly.

(1). If $x \subseteq A$ then $\operatorname{rank}(x) \leq \operatorname{rank}(A)$. So $\operatorname{rank}(\wp(A)) \leq \alpha + 1$. Since $A \in \wp(A)$ in particular, $\operatorname{rank}(\wp(A)) = \alpha + 1$.

(2). By Proposition 67.20

(3). By Proposition 67.20.

(4). By (2), twice.

(5). Note that $A \times B \subseteq \wp(\wp(A \cup B))$, and invoke (4).

(6). If $\alpha = \gamma + 1$, there is some $c \in A$ with $\operatorname{rank}(c) = \gamma$, and no element of A has higher rank; so $\operatorname{rank}(\bigcup A) = \gamma$. If α is a limit ordinal, then A has elements with rank arbitrarily close to (but strictly less than) α, so that $\bigcup A$ also has elements with rank arbitrarily close to (but strictly less than) α, so that $\operatorname{rank}(\bigcup A) = \alpha$. □

We leave it as an exercise to show why (5) involves an *in*equality.

Problem 69.2. Produce sets A and B such that $\operatorname{rank}(A \times B) = \max(\operatorname{rank}(A), \operatorname{rank}(B))$. Produce sets A and B such that $\operatorname{rank}(A \times B) \max(\operatorname{rank}(A), \operatorname{rank}(B)) + 2$. Are any other ranks possible?

We are also now in a position to show that several reasonable notions of what it might mean to describe an ordinal as "finite" or "infinite" coincide:

sth:ord-arithmetic:using-addition:ordinfinitycharacter
sth:ord-arithmetic:using-addition:ord:notinomega
sth:ord-arithmetic:using-addition:ord:omegaplus
sth:ord-arithmetic:using-addition:ord:oneplus
sth:ord-arithmetic:using-addition:ord:plusone
sth:ord-arithmetic:using-addition:ord:infinite

Lemma 69.10. *For any ordinal α, the following are equivalent:*

1. $\alpha \notin \omega$, *i.e.,* α *is not a natural number*
2. $\omega \leq \alpha$
3. $1 + \alpha = \alpha$
4. $\alpha \approx \alpha + 1$, *i.e.,* α *and* $\alpha + 1$ *are equinumerous*
5. α *is Dedekind infinite*

So we have five provably equivalent ways to understand what it takes for an ordinal to be (in)finite.

Proof. *(1)* ⇒ *(2).* By Trichotomy.

(2) ⇒ *(3).* Fix $\alpha \geq \omega$. By Transfinite Induction, there is some least ordinal γ (possibly 0) such that there is a limit ordinal β with $\alpha = \beta + \gamma$. Now:

$$1 + \alpha = 1 + (\beta + \gamma) = (1 + \beta) + \gamma = \operatorname*{lsub}_{\delta < \beta}(1 + \delta) + \gamma = \beta + \gamma = \alpha.$$

(3) ⇒ *(4).* There is clearly a bijection $f \colon (\alpha \sqcup 1) \to (1 \sqcup \alpha)$. If $1 + \alpha = \alpha$, there is an isomorphism $g \colon (1 \sqcup \alpha) \to \alpha$. Now consider $g \circ f$.

(4) ⇒ *(5).* If $\alpha \approx \alpha + 1$, there is a bijection $f \colon (\alpha \sqcup 1) \to \alpha$. Define $g(\gamma) = f(\gamma, 0)$ for each $\gamma < \alpha$; this injection witnesses that α is Dedekind infinite, since $f(0, 1) \in \alpha \setminus \operatorname{ran}(g)$.

(5) ⇒ *(1).* This is Proposition 65.8. □

69.4 Ordinal Multiplication

We now turn to ordinal multiplication, and we approach this much like ordinal addition. So, suppose we want to multiply α by β. To do this, you might imagine a rectangular grid, with width α and height β; the product of α and β is now the result of moving along each row, then moving through the next row… until you have moved through the entire grid. Otherwise put, the product of α and β arises by replacing *each* element in β with a copy of α. sth:ord-arithmetic:mult:sec

To make this formal, we simply use the reverse lexicographic ordering on the Cartesian product of α and β:

Definition 69.11. For any ordinals α, β, their product $\alpha \cdot \beta = \mathrm{ord}(\alpha \times \beta, \lhd)$.

We must again confirm that this is a well-formed definition:

Lemma 69.12. *$\langle \alpha \times \beta, \lhd \rangle$ is a well-order, for any ordinals α and β.* sth:ord-arithmetic:mult:ordtimeslessiswo

Proof. Exactly as for Lemma 69.4. □

And it is not hard to prove that multiplication behaves thus:

Lemma 69.13. *For any ordinals α, β:* sth:ord-arithmetic:mult:ordtimesrecursion

$$\begin{aligned} \alpha \cdot 0 &= 0 \\ \alpha \cdot (\beta + 1) &= (\alpha \cdot \beta) + \alpha \\ \alpha \cdot \beta &= \operatorname{lsub}_{\delta < \beta}(\alpha \cdot \delta) && \text{when } \beta \text{ is a limit ordinal.} \end{aligned}$$

Proof. Left as an exercise. □

Indeed, just as in the case of addition, we could have defined ordinal multiplication via these recursion equations, rather than offering a direct definition. Equally, as with addition, certain behaviour is familiar:

Lemma 69.14. *If α, β, γ are ordinals, then:* sth:ord-arithmetic:mult:ordinalmultiplicationisnice

1. *if $\alpha \neq 0$ and $\beta < \gamma$, then $\alpha \cdot \beta < \alpha \cdot \gamma$;* sth:ord-arithmetic:mult:ordtimes1
2. *if $\alpha \neq 0$ and $\alpha \cdot \beta = \alpha \cdot \gamma$, then $\beta = \gamma$;* sth:ord-arithmetic:mult:ordtimes2
3. $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$; sth:ord-arithmetic:mult:ordtimes3
4. *If $\alpha \leq \beta$, then $\alpha \cdot \gamma \leq \beta \cdot \gamma$;* sth:ord-arithmetic:mult:ordtimes4
5. $\alpha \cdot (\beta + \gamma) = (\alpha \cdot \beta) + (\alpha \cdot \gamma)$. sth:ord-arithmetic:mult:ordtimes5

Proof. Left as an exercise. □

You can prove (or look up) other results, to your heart's content. But, given Proposition 69.8, the following should not come as a surprise:

Proposition 69.15. *Ordinal multiplication is not commutative:* $2 \cdot \omega = \omega < \omega \cdot 2$

Proof. $2 \cdot \omega = \text{lsub}_{n<\omega}(2 \cdot n) = \omega \in \text{lsub}_{n<\omega}(\omega + n) = \omega + \omega = \omega \cdot 2$. □

Again, the intuitive rationale is quite straightforward. To compute $2 \cdot \omega$, you replace each natural number with two entities. You would get the same order type if you simply inserted all the "half" numbers into the natural numbers, i.e., you considered the natural ordering on $\{n/2 : n \in \omega\}$. And, put like that, the order type is plainly the same as that of ω itself. But, to compute $\omega \cdot 2$, you place down two copies of ω, one after the other.

Problem 69.3. Prove Lemma 69.12, Lemma 69.13, and Lemma 69.14

---

content/set-theory/ord-arithmetic/exponentiation.tex

69.5 Ordinal Exponentiation

sth:ord-arithmetic:expo:sec

We now move to ordinal exponentiation. Sadly, there is no *nice* synthetic definition for ordinal exponentiation.

Sure, there *are* explicit synthetic definitions. Here is one. Let $\text{finfun}(\alpha, \beta)$ be the set of all functions $f : \alpha \to \beta$ such that $\{\gamma \in \alpha : f(\gamma) \neq 0\}$ is equinumerous with some natural number. Define a well-ordering on $\text{finfun}(\alpha, \beta)$ by $f \sqsubset g$ iff $f \neq g$ and $f(\gamma_0) < g(\gamma_0)$, where $\gamma_0 = \max\{\gamma \in \alpha : f(\gamma) \neq g(\gamma)\}$. Then we can define $\alpha^{(\beta)}$ as $\text{ord}(\text{finfun}(\alpha, \beta), \sqsubset)$. Potter employs this explicit definition, and then immediately explains:

> The choice of this ordering is determined purely by our desire to obtain a definition of ordinal exponentiation which obeys the appropriate recursive condition..., and it is much harder to picture than either the ordered sum or the ordered product. (Potter, 2004, p. 199)

Quite. We explained addition as "a copy of α followed by a copy of β", and multiplication as "a β-sequence of copies of α". But we have nothing pithy to say about $\text{finfun}(\alpha, \gamma)$. So instead, we'll offer the definition of ordinal exponentiation just *by* transfinite recursion, i.e.:

sth:ord-arithmetic:expo:ordexporecursion

Definition 69.16.

$$\begin{aligned} \alpha^{(0)} &= 1 \\ \alpha^{(\beta+1)} &= \alpha^{(\beta)} \cdot \alpha \\ \alpha^{(\beta)} &= \bigcup_{\delta<\beta} \alpha^{(\delta)} && \text{when } \beta \text{ is a limit ordinal} \end{aligned}$$

If we were working *as* set theorists, we might want to explore some of the properties of ordinal exponentiation. But we have nothing much more to add, except to note the unsurprising fact that ordinal exponentiation does not commute. Thus $2^{(\omega)} = \bigcup_{\delta<\omega} 2^{(\delta)} = \omega$, whereas $\omega^{(2)} = \omega \cdot \omega$. But then, we should not *expect* exponentiation to commute, since it does not commute with natural numbers: $2^{(3)} = 8 < 9 = 3^{(2)}$.

Problem 69.4. Using Transfinite Induction, prove that, if we define $\alpha^{(\beta)} = \mathrm{ord}(\mathrm{finfun}(\alpha, \beta), \sqsubset)$, we obtain the recursion equations of Definition 69.16.

Chapter 70

Cardinals

---

content/set-theory/cardinals/cp.tex

70.1 Cantor's Principle

sth:cardinals:cp:sec

Cast your mind back to section 66.5. We were discussing well-ordered sets, and suggested that it would be nice to have objects which go proxy for well-orders. With this is mind, we introduced ordinals, and then showed in Corollary 66.28 that these behave as we would want them to, i.e.:

$$\mathrm{ord}(A, <) = \mathrm{ord}(B, \lessdot) \text{ iff } \langle A, < \rangle \cong \langle B, \lessdot \rangle.$$

Cast your mind back even further, to section 4.8. There, working naïvely, we introduced the notion of the "size" of a set. Specifically, we said that two sets are equinumerous, $A \approx B$, just in case there is a bijection $f \colon A \to B$. This is an intrinsically simpler notion than that of a well-ordering: we are only interested in bijections, and not (as with order-isomorphisms) whether the bijections "preserve any structure".

This all gives rise to an obvious thought. Just as we introduced certain objects, *ordinals*, to calibrate well-orders, we can introduce certain objects, *cardinals*, to calibrate size. That is the aim of this chapter.

Before we say what these cardinals will be, we should lay down a principle which they ought to satisfy. Writing $|X|$ for the cardinality of the set X, we

would want them to obey:

$$|A| = |B| \text{ iff } A \approx B.$$

We'll call this *Cantor's* Principle, since Cantor was probably the first to have it very clearly in mind. (We'll say more about its relationship to *Hume's* Principle in section 70.5.) So our aim is to define $|X|$, for each X, in such a way that it delivers Cantor's Principle.

---

content/set-theory/cardinals/cardsasords.tex

70.2 Cardinals as Ordinals

sth:cardinals:cardsasords:sec

In fact, our theory of cardinals will just make (shameless) use of our theory of ordinals. That is: we will just define cardinals as certain specific ordinals. In particular, we will offer the following:

sth:cardinals:cardsasords:defcardinalasordinal

Definition 70.1. If A can be well-ordered, then $|A|$ is the least ordinal γ such that $A \approx \gamma$. For any ordinal γ, we say that γ is a *cardinal* iff $\gamma = |\gamma|$.

We just used the phrase "A can be well-ordered". As is almost always the case in mathematics, the modal locution here is just a hand-waving gloss on an existential claim: to say "A can be well-ordered" is just to say "there is a relation which well-orders A".

But there is a snag with Definition 70.1. We would like it to be the case that *every* set has a size, i.e., that $|A|$ exists for every A. The definition we just gave, though, begins with a conditional: "*If* A can be well-ordered…". If there is some set A which cannot be well-ordered, then our definition will simply fail to define an object $|A|$.

So, to use Definition 70.1, we need a guarantee that every set can be well-ordered. Sadly, though, this guarantee is unavailable in **ZF**. So, if we want to use Definition 70.1, there is no alternative but to add a new axiom, such as:

Axiom (Well-Ordering). Every set can be well-ordered.

We will discuss whether the Well-Ordering Axiom is acceptable in chapter 72. From now on, though, we will simply help ourselves to it. And, using it, it is quite straightforward to prove that cardinals (as defined in Definition 70.1) exist and behave nicely:

sth:cardinals:cardsasords:lem:CardinalsExist

Lemma 70.2. *For every set A:*

sth:cardinals:cardsasords:cardaexists

1. *$|A|$ exists and is unique;*

sth:cardinals:cardsasords:cardaapprox

2. *$|A| \approx A$;*

sth:cardinals:cardsasords:cardaidem

3. *$|A|$ is a cardinal, i.e., $|A| = ||A||$;*

Proof. Fix A. By Well-Ordering, there is a well-ordering $\langle A, R\rangle$. By Theorem 66.26, $\langle A, R\rangle$ is isomorphic to a unique ordinal, β. So $A \approx \beta$. By Transfinite Induction, there is a uniquely least ordinal, γ, such that $A \approx \gamma$. So $|A| = \gamma$, establishing (1) and (2). To establish (3), note that if $\delta \in \gamma$ then $\delta \prec A$, by our choice of γ, so that also $\delta \prec \gamma$ since equinumerosity is an equivalence relation (Proposition 4.20). So $\gamma = |\gamma|$. □

The next result guarantees Cantor's Principle, and more besides. (Note that cardinals inherit their ordering from the ordinals, i.e., $\mathfrak{a} < \mathfrak{b}$ iff $\mathfrak{a} \in \mathfrak{b}$. In formulating this, we will use Fraktur letters for objects we know to be cardinals. This is fairly standard. A common alternative is to use Greek letters, since cardinals are ordinals, but to choose them from the middle of the alphabet, e.g.: κ, λ.):

sth:cardinals:cardsasords: lem:CardinalsBehaveRight

Lemma 70.3. *For any sets A and B:*

$$A \approx B \text{ iff } |A| = |B|$$
$$A \preceq B \text{ iff } |A| \leq |B|$$
$$A \prec B \text{ iff } |A| < |B|$$

Proof. We will prove the left-to-right direction of the second claim (the other cases are similar, and left as an exercise). So, consider the following diagram:

The double-headed arrows indicate bijections, whose existence is guaranteed by Lemma 70.2. In assuming that $A \preceq B$, there is an injection $A \to B$. Now, chasing the arrows around from $|A|$ to A to B to $|B|$, we obtain an injection $|A| \to |B|$ (the dashed arrow). □

We can also use Lemma 70.3 to re-prove Schröder–Bernstein. This is the claim that if $A \preceq B$ and $B \preceq A$ then $A \approx B$. We stated this as Theorem 4.25, but first proved it—with some effort—in section 6.5. Now consider:

Re-proof of Schröder-Bernstein. If $A \preceq B$ and $B \preceq A$, then $|A| \leq |B|$ and $|B| \leq |A|$ by Lemma 70.3. So $|A| = |B|$ and $A \approx B$ by Trichotomy and Lemma 70.3. □

Whilst this is a very simple proof, it implicitly relies on both Replacement (to secure Theorem 66.26) and on Well-Ordering (to guarantee Lemma 70.3). By contrast, the proof of section 6.5 was much more self-standing (indeed, it can be carried out in $\mathbf{Z}^-$).

70.3 ZFC: A Milestone

sth:cardinals:zfc:sec

With the addition of Well-Ordering, we have reached the final theoretical milestone. We now have all the axioms required for **ZFC**. In detail:

Definition 70.4. The theory **ZFC** has these axioms: Extensionality, Union, Pairs, Powersets, Infinity, Foundation, Well-Ordering and all instances of the Separation and Replacement schemes. Otherwise put, **ZFC** adds Well-Ordering to **ZF**.

ZFC stands for *Zermelo–Fraenkel* set theory with *Choice*. Now this might seem slightly odd, since the axiom we added was called "Well-Ordering", not "Choice". But, when we later formulate Choice, it will turn out that Well-Ordering is equivalent (modulo **ZF**) to Choice (see Theorem 72.6). So which to take as our "basic" axiom is a matter of indifference. And the name "**ZFC**" is entirely standard in the literature.

---

content/set-theory/cardinals/classing.tex

70.4 Finite, Enumerable, Non-enumerable

sth:cardinals:classing:sec

Now that we have been introduced to cardinals, it is worth spending a little time talking about different varieties of cardinals; specifically, finite, enumerable, and non-enumerable cardinals.

Our first two results entail that the finite cardinals will be exactly the finite ordinals, which we defined as our *natural numbers* back in Definition 65.7:

sth:cardinals:classing:finitecardisoequal

Proposition 70.5. *Let $n, m \in \omega$. Then $n = m$ iff $n \approx m$.*

Proof. *Left-to-right* is trivial. To prove *right-to-left*, suppose $n \approx m$ although $n \neq m$. By Trichotomy, either $n \in m$ or $m \in n$; suppose $n \in m$ without loss of generality. Then $n \subsetneq m$ and there is a bijection $f : m \to n$, so that m is Dedekind infinite, contradicting Proposition 65.8. □

sth:cardinals:classing:naturalsarecardinals

Corollary 70.6. *If $n \in \omega$, then n is a cardinal.*

Proof. Immediate. □

It also follows that several reasonable notions of what it might mean to describe a cardinal as "finite" or "infinite" coincide:

sth:cardinals:classing:generalinfinitycharacter

Theorem 70.7. *For any set A, the following are equivalent:*

sth:cardinals:classing:card:notinomega

1. *$|A| \notin \omega$, i.e., A is not a natural number;*

sth:cardinals:classing:card:omegaplus

2. *$\omega \leq |A|$;*

sth:cardinals:classing:card:infinite

3. *A is Dedekind infinite.*

Proof. From Lemma 69.10, Lemma 70.3, and Corollary 70.6. □

This licenses the following *definition* of some notions which we used rather informally in part I:

Definition 70.8. We say that A is *finite* iff $|A|$ is a natural number, i.e., $|A| \in \omega$. Otherwise, we say that A is *infinite*. sth:cardinals:classing:defnfinite

But note that this definition is presented against the background of **ZFC**. After all, we needed Well-Ordering to guarantee that every set has a cardinality. And indeed, without Well-Ordering, there can be a set which is neither finite nor Dedekind infinite. We will return to this sort of issue in chapter 72. For now, we continue to rely upon Well-Ordering.

Let us now turn from the finite cardinals to the infinite cardinals. Here are two elementary points:

Corollary 70.9. *ω is the least infinite cardinal.* sth:cardinals:classing:omegaisacardinal

Proof. ω is a cardinal, since ω is Dedekind infinite and if $\omega \approx n$ for any $n \in \omega$ then n would be Dedekind infinite, contradicting Proposition 65.8. Now ω is the least infinite cardinal by definition. □

Corollary 70.10. *Every infinite cardinal is a limit ordinal.*

Proof. Let α be an infinite successor ordinal, so $\alpha = \beta + 1$ for some β. By Proposition 70.5, β is also infinite, so $\beta \approx \beta + 1$ by Lemma 69.10. Now $|\beta| = |\beta + 1| = |\alpha|$ by Lemma 70.3, so that $\alpha \neq |\alpha|$. □

Now, as early as Definition 4.27, we flagged we can distinguish between enumerable and non-enumerable infinite sets. That definition naturally leads to the following:

Proposition 70.11. *A is enumerable iff $|A| \leq \omega$, and A is non-enumerable iff $\omega < |A|$.*

Proof. By Trichotomy, the two claims are equivalent, so it suffices to prove that A is enumerable iff $|A| \leq \omega$. For *right-to-left*: if $|A| \leq \omega$, then $A \preceq \omega$ by Lemma 70.3 and Corollary 70.9. For *left-to-right*: suppose A is enumerable; then by Definition 4.27 there are three possible cases:

1. if $A = \emptyset$, then $|A| = 0 \in \omega$, by Corollary 70.6 and Lemma 70.3.
2. if $n \approx A$, then $|A| = n \in \omega$, by Corollary 70.6 and Lemma 70.3.
3. if $\omega \approx A$, then $|A| = \omega$, by Corollary 70.9.

So in all cases, $|A| \leq \omega$. □

Indeed, ω has a special place. Whilst there are many countable ordinals:

Corollary 70.12. *ω is the only enumerable infinite cardinal.*

Proof. Let $\mathfrak{a}$ be an enumerable infinite cardinal. Since $\mathfrak{a}$ is infinite, $\omega \leq \mathfrak{a}$. Since $\mathfrak{a}$ is an enumerable cardinal, $\mathfrak{a} = |\mathfrak{a}| \leq \omega$. So $\mathfrak{a} = \omega$ by Trichotomy. □

Of course, there are infinitely many cardinals. So we might ask: *How many cardinals are there?* The following results show that we might want to reconsider that question.

sth:cardinals:classing:unioncardinalscardinal

Proposition 70.13. *If every member of X is a cardinal, then $\bigcup X$ is a cardinal.*

Proof. It is easy to check that $\bigcup X$ is an ordinal. Let $\alpha \in \bigcup X$ be an ordinal; then $\alpha \in \mathfrak{b} \in X$ for some cardinal $\mathfrak{b}$. Since $\mathfrak{b}$ is a cardinal, $\alpha \prec \mathfrak{b}$. Since $\mathfrak{b} \subseteq \bigcup X$, we have $\mathfrak{b} \preceq \bigcup X$, and so $\alpha \not\approx \bigcup X$. Generalising, $\bigcup X$ is a cardinal. □

sth:cardinals:classing:lem:NoLargestCardinal

Theorem 70.14. *There is no largest cardinal.*

Proof. For any cardinal $\mathfrak{a}$, Cantor's Theorem (Theorem 4.24) and Lemma 70.2 entail that $\mathfrak{a} < |\wp(\mathfrak{a})|$. □

Theorem 70.15. *The set of all cardinals does not exist.*

Proof. For reductio, suppose $C = \{\mathfrak{a} : \mathfrak{a} \text{ is a cardinal}\}$. Now $\bigcup C$ is a cardinal by Proposition 70.13, so by Theorem 70.14 there is a cardinal $\mathfrak{b} > \bigcup C$. By definition $\mathfrak{b} \in C$, so $\mathfrak{b} \subseteq \bigcup C$, so that $\mathfrak{b} \leq \bigcup C$, a contradiction. □

You should compare this with both Russell's Paradox and Burali-Forti.

---

content/set-theory/cardinals/hp.tex

70.5 Appendix: Hume's Principle

sth:cardinals:hp:sec

In section 70.1, we described Cantor's Principle. This was:

$$|A| = |B| \text{ iff } A \approx B.$$

This is very similar to what is now called *Hume's Principle*, which says:

$$\#x\, F(x) = \#x\, G(x) \text{ iff } F \sim G$$

where '$F \sim G$' abbreviates that there are exactly as many Fs as Gs, i.e., the Fs can be put into a bijection with the Gs, i.e.:

$$\begin{aligned}&\exists R(\forall v \forall y(Rvy \to (Fv \wedge Gy)) \wedge \\ &\quad \forall v(Fv \to \exists! y\, Rvy) \wedge \\ &\quad \forall y(Gy \to \exists! v\, Rvy))\end{aligned}$$

But there is a type-difference between Hume's Principle and Cantor's Principle. In the statement of Cantor's Principle, the variables "A" and "B" are first-order terms which stand for *sets*. In the statement of Hume's Principle, "F", "G" and "R" are *not* first-order terms; rather, they are in *predicate position*. (Maybe they stand for *properties*.) So we might gloss Hume's Principle in English as: the number of Fs is the number of Gs iff the Fs are bijective with the Gs. This is called *Hume's Principle*, because Hume once wrote this:

> When two numbers are so combined as that the one has always an unit answering to every unit of the other, we pronounce them equal. (Hume, 1740, Pt.III Bk.1 §1)

And Hume's Principle was brought to contemporary mathematico-logical prominence by Frege (1884, §63), who quoted this passage from Hume, before (in effect) sketching (what we have called) Hume's Principle.

You should note the structural similarity between Hume's Principle and Basic Law V. We formulated this in section 64.6 as follows:

$$\epsilon x\, F(x) = \epsilon x\, G(x) \text{iff } \forall x\, (F(x) \leftrightarrow G(x)).$$

And, at this point, some commentary and comparison might help.

There are two ways to take a principle like Hume's Principle or Basic Law V: *predicatively* or *impredicatively* (recall section 64.3). On the impredicative reading of Basic Law V, for each F, the object $\epsilon x\, F(x)$ falls within the domain of quantification that we used in formulating Basic Law V itself. Similarly, on the impredicative reading of Hume's Principle, for each F, the object $\#x\, F(x)$ falls within the domain of quantification that we used in formulating Hume's Principle. By contrast, on the *predicative* understanding, the objects $\epsilon x\, F(x)$ and $\#x\, F(x)$ would be entities from some *different* domain.

Now, if we read Basic Law V impredicatively, it leads to inconsistency, via Naïve Comprehension (for the details, see section 64.6). Much like Naïve Comprehension, it can be rendered consistent by reading it *predicatively*. But it probably will not do everything that we wanted it to.

Hume's Principle, however, *can* consistently be read impredicatively. And, read thus, it is quite powerful.

To illustrate: consider the predicate "$x \neq x$", which obviously nothing satisfies. Hume's Principle now yields an object $\#x(x \neq x)$. We might treat this as the number 0. Now, on the *impredicative* understanding—but *only* on the impredicative understanding—this entity 0 falls within our original domain of quantification. So we can sensibly apply Hume's Principle with the predicate "$x = 0$" to obtain an object $\#x(x = 0)$. We might treat this as the number 1. Moreover, Hume's Principle entails that $0 \neq 1$, since there cannot be a bijection from the non-self-identical objects to the objects identical with 0 (there are none of the former, but one of the latter). Now, working impredicatively again, 1 falls within our original domain of quantification. So we can sensibly apply Hume's Principle with the predicate "$(x = 0 \vee x = 1)$" to obtain an object

$\#x(x = 0 \vee x = 1)$. We might treat this as the number 2, and we can show that $0 \neq 2$ and $1 \neq 2$ and so on.

In short, taken impredicatively, Hume's Principle entails that there are *infinitely many objects*. And this has encouraged *neo-Fregean logicists* to take Hume's Principle as the foundation for arithmetic.

Frege *himself*, though, did not take Hume's Principle as his foundation for arithmetic. Instead, Frege proved Hume's Principle from an explicit definition: $\#x\, F(x)$ is defined as the extension of the concept $F \sim \Phi$. In modern terms, we might attempt to render this as $\#x\, F(x) = \{G : F \sim G\}$; but this will pull us back into the problems of Naïve Comprehension.

Chapter 71

Cardinal Arithmetic

---

content/set-theory/card-arithmetic/opps.tex

71.1 Defining the Basic Operations

sth:card-arithmetic:opps:sec

Since we do not need to keep track of order, cardinal arithmetic is rather easier to define than ordinal arithmetic. We will define addition, multiplication, and exponentiation simultaneously.

Definition 71.1. When $\mathfrak{a}$ and $\mathfrak{b}$ are cardinals:

$$\begin{aligned} \mathfrak{a} \oplus \mathfrak{b} &= |\mathfrak{a} \sqcup \mathfrak{b}| \\ \mathfrak{a} \otimes \mathfrak{b} &= |\mathfrak{a} \times \mathfrak{b}| \\ \mathfrak{a}^{\mathfrak{b}} &= \left|{}^{\mathfrak{b}}\mathfrak{a}\right| \end{aligned}$$

where ${}^{X}Y = \{f : f \text{ is a function } X \to Y\}$. (It is easy to show that ${}^{X}Y$ exists for any sets X and Y; we leave this as an exercise.)

Problem 71.1. Prove in $\mathbf{Z}^-$ that ${}^{X}Y$ exists for any sets X and Y. Working in **ZF**, compute $\mathrm{rank}({}^{X}Y)$ from $\mathrm{rank}(X)$ and $\mathrm{rank}(Y)$, in the manner of Lemma 69.9.

It might help to explain this definition. Concerning addition: this uses the notion of disjoint sum, $\sqcup$, as defined in Definition 69.1; and it is easy to see that this definition gives the right verdict for finite cases. Concerning multiplication: Proposition 1.27 tells us that if A has n members and B has m members then $A \times B$ has $n \cdot m$ members, so our definition simply generalises the idea to transfinite multiplication. Exponentiation is similar: we are simply generalising the thought from the finite to the transfinite. Indeed, in certain ways, transfinite cardinal arithmetic looks much more like "ordinary" arithmetic than does transfinite ordinal arithmetic:

Proposition 71.2. *$\oplus$ and $\otimes$ are commutative and associative.* *sth:card-arithmetic:opps: cardplustimescommute*

Proof. For commutativity, by Lemma 70.3 it suffices to observe that $(\mathfrak{a} \sqcup \mathfrak{b}) \approx (\mathfrak{b} \sqcup \mathfrak{a})$ and $(\mathfrak{a} \times \mathfrak{b}) \approx (\mathfrak{b} \times \mathfrak{a})$. We leave associativity as an exercise. □

Problem 71.2. Prove that $\oplus$ and $\otimes$ are associative.

Proposition 71.3. *A is infinite iff $|A| \oplus 1 = 1 \oplus |A| = |A|$.*

Proof. As in Theorem 70.7, from Lemma 69.10 and Lemma 70.3. □

This explains why we need to use different symbols for ordinal versus cardinal addition/multiplication: these are genuinely *different* operations. This next pair of results shows that ordinal versus cardinal exponentiation are also different operations. (Recall that Definition 65.7 entails that $2 = \{0, 1\}$):

Lemma 71.4. *$|\wp(A)| = 2^{|A|}$, for any A.* *sth:card-arithmetic:opps: lem:SizePowerset2Exp*

Proof. For each subset $B \subseteq A$, let $\chi_B \in {}^{A}2$ be given by:

$$\chi_B(x) = \begin{cases} 1 & \text{if } x \in B \\ 0 & \text{otherwise.} \end{cases}$$

Now let $f(B) = \chi_B$; this defines a bijection $f : \wp(A) \to {}^{A}2$. So $\wp(A) \approx {}^{A}2$. Hence $\wp(A) \approx {}^{|A|}2$, so that $|\wp(A)| = \left|{}^{|A|}2\right| = 2^{|A|}$. □

This snappy proof essentially subsumes the discussion of section 4.13. There, we showed how to "reduce" the uncountability of $\wp(\omega)$ to the uncountability of the set of infinite binary strings, $\mathbb{B}^\omega$. In effect, $\mathbb{B}^\omega$ is just ${}^{\omega}2$; and the preceding proof showed that the reasoning we went through in section 4.13 will go through using any set A in place of ω. The result also yields a quick fact about cardinal exponentiation:

Corollary 71.5. *$\mathfrak{a} < 2^{\mathfrak{a}}$ for any cardinal $\mathfrak{a}$.* *sth:card-arithmetic:opps: cantorcor*

Proof. From Cantor's Theorem (Theorem 4.24) and Lemma 71.4. □

So $\omega < 2^\omega$. But note: this is a result about *cardinal* exponentiation. It should be contrasted with *ordinal* exponentiation, since in the latter case $\omega = 2^{(\omega)}$ (see section 69.5).

Whilst we are on the topic of cardinal exponentiation, we can also be a bit more precise about the "way" in which $\mathbb{R}$ is non-enumerable.

sth:card-arithmetic:opps:continuumis2aleph0

Theorem 71.6. $|\mathbb{R}| = 2^\omega$

Proof skeleton. There are plenty of ways to prove this. The most straightforward is to argue that $\wp(\omega) \preceq \mathbb{R}$ and $\mathbb{R} \preceq \wp(\omega)$, and then use Schröder-Bernstein to infer that $\mathbb{R} \approx \wp(\omega)$, and Lemma 71.4 to infer that $|\mathbb{R}| = 2^\omega$. We leave it as an (illuminating) exercise to define injections $f\colon \wp(\omega) \to \mathbb{R}$ and $g\colon \mathbb{R} \to \wp(\omega)$.□

Problem 71.3. Complete the proof of Theorem 71.6, by showing that $\wp(\omega) \preceq \mathbb{R}$ and $\mathbb{R} \preceq \wp(\omega)$.

---

content/set-theory/card-arithmetic/simp.tex

71.2 Simplifying Addition and Multiplication

sth:card-arithmetic:simp:sec

It turns out that transfinite cardinal addition and multiplication is *extremely* easy. This follows from the fact that cardinals are (certain) ordinals, and so well-ordered, and so can be manipulated in a certain way. Showing this, though, is *not* so easy. To start, we need a tricksy definition:

Definition 71.7. We define a *canonical ordering*, $\lhd$, on pairs of ordinals, by stipulating that $\langle \alpha_1, \alpha_2\rangle \lhd \langle \beta_1, \beta_2\rangle$ iff either:

1. $\max(\alpha_1, \alpha_2) < \max(\beta_1, \beta_2)$; or
2. $\max(\alpha_1, \alpha_2) = \max(\beta_1, \beta_2)$ and $\alpha_1 < \beta_1$; or
3. $\max(\alpha_1, \alpha_2) = \max(\beta_1, \beta_2)$ and $\alpha_1 = \beta_1$ and $\alpha_2 < \beta_2$

Lemma 71.8. *$\langle \alpha \times \alpha, \lhd\rangle$ is a well-order, for any ordinal α.*

Proof. Evidently $\lhd$ is connected on $\alpha \times \alpha$. For suppose that neither $\langle \alpha_1, \alpha_2\rangle$ nor $\langle \beta_1, \beta_2\rangle$ is $\lhd$-less than the other. Then $\max(\alpha_1, \alpha_2) = \max(\beta_1, \beta_2)$ and $\alpha_1 = \beta_1$ and $\alpha_2 = \beta_2$, so that $\langle \alpha_1, \alpha_2\rangle = \langle \beta_1, \beta_2\rangle$.

To show well-ordering, let $X \subseteq \alpha \times \alpha$ be non-empty. Since α is an ordinal, some δ is the least member of $\{\max(\gamma_1, \gamma_2) : \langle \gamma_1, \gamma_2\rangle \in X\}$. Now discard all pairs from $\{\langle \gamma_1, \gamma_2\rangle \in X : \max(\gamma_1, \gamma_2) = \delta\}$ except those with least first coordinate; from among these, the pair with least second coordinate is the $\lhd$-least element of X. □

Now for a teensy, simple observation:

sth:card-arithmetic:simp:simplecardproduct

Proposition 71.9. *If $\alpha \approx \beta$, then $\alpha \times \alpha \approx \beta \times \beta$.*

Proof. Just let $f : \alpha \to \beta$ induce $\langle \gamma_1, \gamma_2 \rangle \mapsto \langle f(\gamma_1), f(\gamma_2) \rangle$. □

And now we will put all this to work, in proving a crucial lemma:

Lemma 71.10. *$\alpha \approx \alpha \times \alpha$, for any infinite ordinal α* *sth:card-arithmetic:simp:alphatimesalpha*

Proof. For reductio, let α be the least infinite ordinal for which this is false. Proposition 4.12 shows that $\omega \approx \omega \times \omega$, so $\omega \in \alpha$. Moreover, α is a cardinal: suppose otherwise, for reductio; then $|\alpha| \in \alpha$, so that $|\alpha| \approx |\alpha| \times |\alpha|$, by hypothesis; and $|\alpha| \approx \alpha$ by definition; so that $\alpha \approx \alpha \times \alpha$ by Proposition 71.9.

Now, for each $\langle \gamma_1, \gamma_2 \rangle \in \alpha \times \alpha$, consider the segment:

$$\mathrm{Seg}(\gamma_1, \gamma_2) = \{\langle \delta_1, \delta_2 \rangle \in \alpha \times \alpha : \langle \delta_1, \delta_2 \rangle \lhd \langle \gamma_1, \gamma_2 \rangle\}$$

Letting $\gamma = \max(\gamma_1, \gamma_2)$, note that $\langle \gamma_1, \gamma_2 \rangle \lhd \langle \gamma + 1, \gamma + 1 \rangle$. So, when γ is infinite, observe:

$$\begin{aligned}
\mathrm{Seg}(\gamma_1, \gamma_2) &\precsim ((\gamma + 1) \cdot (\gamma + 1)) \\
&\approx (\gamma \cdot \gamma), \text{ by Lemma 69.10 and Proposition 71.9} \\
&\approx \gamma, \text{ by the induction hypothesis} \\
&\prec \alpha, \text{ since } \alpha \text{ is a cardinal}
\end{aligned}$$

So $\mathrm{ord}(\alpha \times \alpha, \lhd) \leq \alpha$, and hence $\alpha \times \alpha \preceq \alpha$. Since of course $\alpha \preceq \alpha \times \alpha$, the result follows by Schröder-Bernstein. □

Finally, we get to our simplifying result:

Theorem 71.11. *If $\mathfrak{a}, \mathfrak{b}$ are infinite cardinals, then:* *sth:card-arithmetic:simp:cardplustimesmax*

$$\mathfrak{a} \otimes \mathfrak{b} = \mathfrak{a} \oplus \mathfrak{b} = \mathit{max}(\mathfrak{a}, \mathfrak{b}).$$

Proof. Without loss of generality, suppose $\mathfrak{a} = \max(\mathfrak{a}, \mathfrak{b})$. Then invoking Lemma 71.10, $\mathfrak{a} \otimes \mathfrak{a} = \mathfrak{a} \leq \mathfrak{a} \oplus \mathfrak{b} \leq \mathfrak{a} \oplus \mathfrak{a} \leq \mathfrak{a} \otimes \mathfrak{a}$. □

Similarly, if $\mathfrak{a}$ is infinite, an $\mathfrak{a}$-sized union of $\leq \mathfrak{a}$-sized sets has size $\leq \mathfrak{a}$:

Proposition 71.12. *Let $\mathfrak{a}$ be an infinite cardinal. For each ordinal $\beta \in \mathfrak{a}$, let X_β be a set with $|X_\beta| \leq \mathfrak{a}$. Then $\left|\bigcup_{\beta \in \mathfrak{a}} X_\beta\right| \leq \mathfrak{a}$.* *sth:card-arithmetic:simp:kappaunionkappasize*

Proof. For each $\beta \in \mathfrak{a}$, fix an injection $f_\beta : X_\beta \to \mathfrak{a}$.[1] Define an injection $g : \bigcup_{\beta \in \mathfrak{a}} X_\beta \to \mathfrak{a} \times \mathfrak{a}$ by $g(v) = \langle \beta, f_\beta(v) \rangle$, where $v \in X_\beta$ and $v \notin X_\gamma$ for any $\gamma \in \beta$. Now $\bigcup_{\beta \in \mathfrak{a}} X_\beta \preceq \mathfrak{a} \times \mathfrak{a} \approx \mathfrak{a}$ by Theorem 71.11. □

content/set-theory/card-arithmetic/expotough.tex

[1] How are these "fixed"? See section 72.5.

71.3 Some Simplification with Cardinal Exponentiation

sth:card-arithmetic:expotough:sec

Whilst defining $\lhd$ was a little involved, the upshot is a useful result concerning cardinal addition and multiplication, Theorem 71.11. Transfinite exponentiation, however, cannot be simplified so straightforwardly. To explain why, we start with a result which extends a familiar pattern from the finitary case (though its proof is at a high level of abstraction):

sth:card-arithmetic:expotough:simplecardexpo

Proposition 71.13. *$\mathfrak{a}^{\mathfrak{b}\oplus\mathfrak{c}} = \mathfrak{a}^{\mathfrak{b}}\otimes\mathfrak{a}^{\mathfrak{c}}$ and $(\mathfrak{a}^{\mathfrak{b}})^{\mathfrak{c}} = \mathfrak{a}^{\mathfrak{b}\otimes\mathfrak{c}}$, for any cardinals $\mathfrak{a},\mathfrak{b},\mathfrak{c}$.*

Proof. For the first claim, consider a function $f\colon(\mathfrak{b}\sqcup\mathfrak{c})\to\mathfrak{a}$. Now "split this", by defining $f_{\mathfrak{b}}(\beta) = f(\beta,0)$ for each $\beta\in\mathfrak{b}$, and $f_{\mathfrak{c}}(\gamma) = f(\gamma,1)$ for each $\gamma\in\mathfrak{c}$. The map $f\mapsto(f_{\mathfrak{b}}\times f_{\mathfrak{c}})$ is a bijection ${}^{\mathfrak{b}\sqcup\mathfrak{c}}\mathfrak{a}\to({}^{\mathfrak{b}}\mathfrak{a}\times{}^{\mathfrak{c}}\mathfrak{a})$.

For the second claim, consider a function $f\colon\mathfrak{c}\to({}^{\mathfrak{b}}\mathfrak{a})$; so for each $\gamma\in\mathfrak{c}$ we have some function $f(\gamma)\colon\mathfrak{b}\to\mathfrak{a}$. Now define $f^*(\beta,\gamma) = (f(\gamma))(\beta)$ for each $\langle\beta,\gamma\rangle\in\mathfrak{b}\times\mathfrak{c}$. The map $f\mapsto f^*$ is a bijection ${}^{\mathfrak{c}}({}^{\mathfrak{b}}\mathfrak{a})\to{}^{\mathfrak{b}\otimes\mathfrak{c}}\mathfrak{a}$. □

Now, what we would *like* is an easy way to compute $\mathfrak{a}^{\mathfrak{b}}$ when we are dealing with infinite cardinals. Here is a nice step in this direction:

sth:card-arithmetic:expotough:cardexpo2reduct

Proposition 71.14. *If $2\leq\mathfrak{a}\leq\mathfrak{b}$ and $\mathfrak{b}$ is infinite, then $\mathfrak{a}^{\mathfrak{b}} = 2^{\mathfrak{b}}$*

Proof.

$$\begin{aligned}
2^{\mathfrak{b}} &\leq \mathfrak{a}^{\mathfrak{b}}, \text{ as } 2\leq\mathfrak{a}\\
&\leq (2^{\mathfrak{a}})^{\mathfrak{b}}, \text{ by Lemma 71.4}\\
&= 2^{\mathfrak{a}\otimes\mathfrak{b}}, \text{ by Proposition 71.13}\\
&= 2^{\mathfrak{b}}, \text{ by Theorem 71.11}
\end{aligned}$$

□

We should not really expect to be able to simplify this any further, since $\mathfrak{b} < 2^{\mathfrak{b}}$ by Lemma 71.4. However, this does not tell us what to say about $\mathfrak{a}^{\mathfrak{b}}$ when $\mathfrak{b} < \mathfrak{a}$. Of course, if $\mathfrak{b}$ is *finite*, we know what to do.

Proposition 71.15. *If $\mathfrak{a}$ is infinite and $n\in\omega$ then $\mathfrak{a}^n = \mathfrak{a}$*

Proof. $\mathfrak{a}^n = \mathfrak{a}\otimes\mathfrak{a}\otimes\ldots\otimes\mathfrak{a} = \mathfrak{a}$, by Theorem 71.11. □

Additionally, in some other cases, we can control the size of $\mathfrak{a}^{\mathfrak{b}}$:

Proposition 71.16. *If $2\leq\mathfrak{b}<\mathfrak{a}\leq 2^{\mathfrak{b}}$ and $\mathfrak{b}$ is infinite, then $\mathfrak{a}^{\mathfrak{b}} = 2^{\mathfrak{b}}$*

Proof. $2^{\mathfrak{b}}\leq\mathfrak{a}^{\mathfrak{b}}\leq(2^{\mathfrak{b}})^{\mathfrak{b}} = 2^{\mathfrak{b}\otimes\mathfrak{b}} = 2^{\mathfrak{b}}$, reasoning as in Proposition 71.14. □

But, beyond this point, things become rather more subtle.

71.4 The Continuum Hypothesis

The previous result hints (correctly) that cardinal exponentiation would be quite *easy*, if infinite cardinals are guaranteed to "play straightforwardly" with powers of 2, i.e., (by Lemma 71.4) with taking powersets. But we cannot assume that infinite cardinals *do* play straightforwardly powersets. sth:card-arithmetic:ch:sec

To start unpacking this, we introduce some nice notation.

Definition 71.17. Where $\mathfrak{a}^{\oplus}$ is the least cardinal strictly greater than $\mathfrak{a}$, we define two infinite sequences:

$$\begin{aligned} \aleph_0 &= \omega & \beth_0 &= \omega \\ \aleph_{\alpha+1} &= (\aleph_\alpha)^{\oplus} & \beth_{\alpha+1} &= 2^{\beth_\alpha} \\ \aleph_\alpha &= \bigcup_{\beta<\alpha} \aleph_\beta & \beth_\alpha &= \bigcup_{\beta<\alpha} \beth_\beta \end{aligned} \qquad \text{when } \alpha \text{ is a limit ordinal.}$$

The definition of $\mathfrak{a}^{\oplus}$ is in order, since Theorem 70.14 tells us that, for each cardinal $\mathfrak{a}$, there is some cardinal greater than $\mathfrak{a}$, and Transfinite Induction guarantees that there is a *least* cardinal greater than $\mathfrak{a}$. The rest of the definition of $\mathfrak{a}$ is provided by transfinite recursion.

Cantor introduced this "$\aleph$" notation; this is *aleph*, the first letter in the Hebrew alphabet and the first letter in the Hebrew word for "infinite". Peirce introduced the "$\beth$" notation; this is *beth*, which is the second letter in the Hebrew alphabet.[2] Now, these notations provide us with infinite cardinals.

Proposition 71.18. *$\aleph_\alpha$ and $\beth_\alpha$ are cardinals, for every ordinal α.*

Proof. Both results hold by a simple transfinite induction. $\aleph_0 = \beth_0 = \omega$ is a cardinal by Corollary 70.9. Assuming $\aleph_\alpha$ and $\beth_\alpha$ are both cardinals, $\aleph_{\alpha+1}$ and $\beth_{\alpha+1}$ are explicitly defined as cardinals. And the union of a set of cardinals is a cardinal, by Proposition 70.13. □

Moreover, every infinite cardinal is an $\aleph$:

Proposition 71.19. *If $\mathfrak{a}$ is an infinite cardinal, then $\mathfrak{a} = \aleph_\gamma$ for some unique γ.*

Proof. By transfinite induction on cardinals. For induction, suppose that if $\mathfrak{b} < \mathfrak{a}$ then $\mathfrak{b} = \aleph_{\gamma_\mathfrak{b}}$. If $\mathfrak{a} = \mathfrak{b}^{\oplus}$ for some $\mathfrak{b}$, then $\mathfrak{a} = (\aleph_{\gamma_\mathfrak{b}})^{\oplus} = \aleph_{\gamma_\mathfrak{b}+1}$. If $\mathfrak{a}$ is not the successor of any cardinal, then since cardinals are ordinals $\mathfrak{a} = \bigcup_{\mathfrak{b}<\mathfrak{a}} \mathfrak{b} = \bigcup_{\mathfrak{b}<\mathfrak{a}} \aleph_{\gamma_\mathfrak{b}}$, so $\mathfrak{a} = \aleph_\gamma$ where $\gamma = \bigcup_{\mathfrak{b}<\mathfrak{a}} \gamma_\mathfrak{b}$. □

Since every infinite cardinal is an $\aleph$, this prompts us to ask: is every infinite cardinal a $\beth$? Certainly if that *were* the case, then the infinite cardinals would "play straightforwardly" with the operation of taking powersets. Indeed, we would have the following:

[2] Peirce used this notation in a letter to Cantor of December 1900. Unfortunately, Peirce also gave a bad argument there that $\beth_\alpha$ does not exist for $\alpha \geq \omega$.

Generalized Continuum Hypothesis (GCH). $\aleph_\alpha = \beth_\alpha$, for all α.

Moreover, if GCH held, then we could make some considerable simplifications with cardinal exponentiation. In particular, we could show that when $\mathfrak{b} < \mathfrak{a}$, the value of $\mathfrak{a}^\mathfrak{b}$ is trapped by $\mathfrak{a} \leq \mathfrak{a}^\mathfrak{b} \leq \mathfrak{a}^\oplus$. We could then go on to give precise conditions which determine which of the two possibilities obtains (i.e., whether $\mathfrak{a} = \mathfrak{a}^\mathfrak{b}$ or $\mathfrak{a}^\mathfrak{b} = \mathfrak{a}^\oplus$).[3]

But GCH is a *hypothesis*, not a *theorem*. In fact, Gödel (1938) proved that if **ZFC** is consistent, then so is **ZFC** + GCH. But it later turned out that we can equally add ¬GCH to **ZFC**. Indeed, consider the simplest non-trivial *instance* of GCH, namely:

Continuum Hypothesis (CH). $\aleph_1 = \beth_1$.

Cohen (1963) proved that if **ZFC** is consistent then so is **ZFC** + ¬CH. So the Continuum Hypothesis is independent from **ZFC**.

The Continuum Hypothesis is so-called, since "the continuum" is another name for the real line, $\mathbb{R}$. Theorem 71.6 tells us that $|\mathbb{R}| = \beth_1$. So the Continuum Hypothesis states that there is no cardinal between the cardinality of the natural numbers, $\aleph_0 = \beth_0$, and the cardinality of the continuum, $\beth_1$.

Given the *independence* of (G)CH from **ZFC**, what should say about their *truth*? Well, there is *much* to say. Indeed, and much fertile recent work in set theory has been directed at investigating these issues. But two very quick points are certainly worth emphasising.

First: it does not *immediately* follow from these formal independence results that either GCH or CH is *indeterminate* in truth value. After all, maybe we just need to add more axioms, which strike us as natural, and which will settle the question one way or another. Gödel himself suggested that this was the right response.

Second: the independence of CH from **ZFC** is certainly *striking*, but it is certainly not *incredible* (in the literal sense). The point is simply that, for all **ZFC** tells us, moving from cardinals to their successors may involve a less blunt tool than simply taking powersets.

With those two observations made, if you want to know more, you will now have to turn to the various philosophers and mathematicians with horses in the race.[4]

---

content/set-theory/card-arithmetic/fix.tex

[3] The condition is dictated by *cofinality*.

[4] Though you might want to start by reading Potter (2004, §15.6).

71.5 ℵ-Fixed Points

sth:card-arithmetic:fix:sec

In chapter 67, we suggested that Replacement stands in need of justification, because it forces the hierarchy to be rather tall. Having done some cardinal arithmetic, we can give a little illustration of the height of the hierarchy.

Evidently $0 < \aleph_0$, and $1 < \aleph_1$, and $2 < \aleph_2 \ldots$ and, indeed, the difference in size only gets *bigger* with every step. So it is tempting to conjecture that $\kappa < \aleph_\kappa$ for every ordinal κ.

But this conjecture is *false*, given **ZFC**. In fact, we can prove that there are ℵ-*fixed-points*, i.e., cardinals κ such that $\kappa = \aleph_\kappa$.

sth:card-arithmetic:fix:alephfixed

Proposition 71.20. *There is an ℵ-fixed-point.*

Proof. Using recursion, define:

$$\begin{aligned} \kappa_0 &= 0 \\ \kappa_{n+1} &= \aleph_{\kappa_n} \\ \kappa &= \bigcup_{n<\omega} \kappa_n \end{aligned}$$

Now κ is a cardinal by Proposition 70.13. But now:

$$\kappa = \bigcup_{n<\omega} \kappa_{n+1} = \bigcup_{n<\omega} \aleph_{\kappa_n} = \bigcup_{\alpha<\kappa} \aleph_\alpha = \aleph_\kappa$$

□

Boolos once wrote an article about exactly the ℵ-fixed-point we just constructed. After noting the existence of κ, at the start of his article, he said:

> [κ is] a *pretty big* number, by the lights of those with no previous exposure to set theory, so big, it seems to me, that it calls into question the truth of any theory, one of whose assertions is the claim that there are at least κ objects. (Boolos, 2000, p. 257)

And he ultimately concluded his paper by asking:

> [do] we suspect that, however it may have been at the beginning of the story, by the time we have come thus far the wheels are spinning and we are no longer listening to a description of anything that is the case? (Boolos, 2000, p. 268)

If we have, indeed, outrun "anything that is the case", then we must point the finger of blame directly at Replacement. For it is this axiom which allows our proof to work. In which case, one assumes, Boolos would need to revisit the claim he made, a few decades earlier, that Replacement has "no undesirable" consequences (see section 68.3).

But is the existence of κ so bad? It might help, here, to consider Russell's *Tristram Shandy paradox*. Tristram Shandy documents his life in his diary, but it takes him a year to record a single day. With every passing year, Tristram

falls further and further behind: after one year, he has recorded only one day, and has lived 364 days unrecorded days; after two years, he has only recorded two days, and has lived 728 unrecorded days; after three years, he has only recorded three days, and lived 1092 unrecorded days ...[5] Still, if Tristram is *immortal*, Tristram will manage to record every day, for he will record the nth day on the nth year of his life. And so, "at the end of time", Tristram will have a complete diary.

Now: why is this so different from the thought that α is smaller than $\aleph_\alpha$—and indeed, increasingly, desperately smaller—up until κ, at which point, we catch up, and $\kappa = \aleph_\kappa$?

Setting that aside, and assuming we accept **ZFC**, let's close with a little more fun concerning fixed-point constructions. The next three results establish, intuitively, that there is a (non-trivial) point at which the hierarchy is as wide as it is tall:

sth:card-arithmetic:fix:bethfixed

Proposition 71.21. *There is a $\beth$-fixed-point, i.e., a κ such that $\kappa = \beth_\kappa$.*

Proof. As in Proposition 71.20, using "$\beth$" in place of "$\aleph$". □

sth:card-arithmetic:fix:stagesize

Proposition 71.22. *$|V_{\omega+\alpha}| = \beth_\alpha$. If $\omega \cdot \omega \leq \alpha$, then $|V_\alpha| = \beth_\alpha$.*

Proof. The first claim holds by a simple transfinite induction. The second claim follows, since if $\omega \cdot \omega \leq \alpha$ then $\omega + \alpha = \alpha$. To establish this, we use facts about ordinal arithmetic from chapter 69. First note that $\omega \cdot \omega = \omega \cdot (1 + \omega) = (\omega \cdot 1) + (\omega \cdot \omega) = \omega + (\omega \cdot \omega)$. Now if $\omega \cdot \omega \leq \alpha$, i.e., $\alpha = (\omega \cdot \omega) + \beta$ for some β, then $\omega + \alpha = \omega + ((\omega \cdot \omega) + \beta) = (\omega + (\omega \cdot \omega)) + \beta = (\omega \cdot \omega) + \beta = \alpha$. □

Corollary 71.23. *There is a κ such that $|V_\kappa| = \kappa$.*

Proof. Let κ be a $\beth$-fixed point, as given by Proposition 71.21. Clearly $\omega \cdot \omega < \kappa$. So $|V_\kappa| = \beth_\kappa = \kappa$ by Proposition 71.22. □

There are as many stages beneath V_κ as there are elements of V_κ. Intuitively, then, V_κ is as wide as it is tall. This is very Tristram-Shandy-esque: we move from one stage to the next by taking *powersets*, thereby making our hierarchy *much* bigger with each step. But, "in the end", i.e., at stage κ, the hierarchy's width catches up with its height.

One might ask: *How often does the hierarchy's width match its height?* The answer is: *As often as there are ordinals.* But this needs a little explanation.

We define a term τ as follows. For any A, let:

$$\begin{aligned} \tau_0(A) &= |A| \\ \tau_{n+1}(A) &= \beth_{\tau_n(A)} \\ \tau(A) &= \bigcup_{n<\omega} \tau_n(A) \end{aligned}$$

[5] Forgetting about leap years.

As in Proposition 71.21, $\tau(A)$ is a $\supseteq$-fixed point for any A, and trivially $|A| < \tau(A)$. So now consider this recursive definition:

$$\begin{aligned} W_0 &= 0 \\ W_{\alpha+1} &= \tau(W_\alpha) \\ W_\alpha &= \bigcup_{\beta<\alpha} W_\beta, \text{ when } \alpha \text{ is a limit} \end{aligned}$$

The construction is defined for all ordinals. Intuitively, then, W is "an injection" from the ordinals to $\supseteq$-fixed points. And, exactly as before, V_{W_α} is as wide as it is tall, for any α.

Chapter 72

Choice

---

content/set-theory/choice/introduction.tex

72.1 Introduction

sth:choice:intro:sec

In chapters 70 to 71, we developed a theory of cardinals by treating cardinals as ordinals. That approach depends upon the Axiom of Well-Ordering. It turns out that Well-Ordering is equivalent to another principle—the Axiom of Choice—and there has been serious philosophical discussion of its acceptability. Our question for this chapter are: How is the Axiom used, and can it be justified?

---

content/set-theory/choice/tarskiscott.tex

72.2 The Tarski–Scott Trick

sth:choice:tarskiscott:sec

In Definition 70.1, we defined cardinals as ordinals. To do this, we assumed the Axiom of Well-Ordering. We did this, for no other reason than that it is the "industry standard".

Before we discuss any of the philosophical issues surrounding Well-Ordering, then, it is important to be clear that we *can* depart from the industry standard,

and develop a theory of cardinals *without* assuming Well-Ordering. We can still employ the definitions of $A \approx B$, $A \preceq B$ and $A \prec B$, as they appeared in chapter 4. We will just need a new notion of *cardinal*.

A naïve thought would be to attempt to define A's cardinality thus:

$$\{x : A \approx x\}.$$

You might want to compare this with Frege's definition of $\#xFx$, sketched at the very end of section 70.5. And, for reasons we gestured at there, this definition fails. Any singleton set is equinumerous with $\{\emptyset\}$. But new singleton sets are formed at every successor stage of the hierarchy (just consider the singleton of the previous stage). So $\{x : A \approx x\}$ does not exist, since it cannot have a rank.

To get around this problem, we use a trick due to Tarski and Scott:[1]

Definition 72.1 (Tarski–Scott). For any formula $\varphi(x)$, let $[x : \varphi(x)]$ be the set of all x, of least possible rank, such that $\varphi(x)$ (or $\emptyset$, if there are no φs).

We should check that this definition is legitimate. Working in **ZF**, Theorem 67.13 guarantees that $\mathrm{rank}(x)$ exists for every x. Now, if there are any entities satisfying φ, then we can let α be the least rank such that $(\exists x \subseteq V_\alpha)\varphi(x)$, i.e., $(\forall \beta \in \alpha)(\forall x \subseteq V_\beta)\neg\varphi(x)$. We can then define $[x : \varphi(x)]$ by Separation as $\{x \in V_{\alpha+1} : \varphi(x)\}$.

Having justified the Tarski–Scott trick, we can now use it to define a notion of cardinality:

Definition 72.2. The TS-cardinality of A is $\mathrm{tsc}(A) = [x : A \approx x]$.

The definition of a TS-cardinal does not use Well-Ordering. But, even without that Axiom, we can show that TS-*cardinals* behave rather like *cardinals* as defined in Definition 70.1. For example, if we restate Lemma 70.3 and Lemma 71.4 in terms of TS-cardinals, the proofs go through just fine in **ZF**, without assuming Well-Ordering.

Whilst we are on the topic, it is worth noting that we can also develop a theory of ordinals using the Tarski–Scott trick. Where $\langle A, <\rangle$ is a well-ordering, let $\mathrm{tso}(A, <) = [\langle X, R\rangle : \langle A, <\rangle \cong \langle X, R\rangle]$. For more on this treatment of cardinals and ordinals, see Potter (2004, chs. 9–12).

---

content/set-theory/choice/hartogs.tex

72.3 Comparability and Hartogs' Lemma

sth:choice:hartogs:sec

That's the plus side. Here's the minus side. Without Choice, things get *messy*. To see why, here is a nice result due to Hartogs (1915):

sth:choice:hartogs:HartogsLemma

Lemma 72.3 (*in* ZF). *For any set A, there is an ordinal α such that $\alpha \not\preceq A$*

[1] A reminder: all formulas may have parameters (unless explicitly stated otherwise).

Proof. If $B \subseteq A$ and $R \subseteq B^2$, then $\langle B, R\rangle \subseteq V_{\mathrm{rank}(A)+4}$ by Lemma 69.9. So, using Separation, consider:

$$C = \{\langle B, R\rangle \in V_{\mathrm{rank}(A)+5} : B \subseteq A \text{ and } \langle B, R\rangle \text{ is a well-ordering}\}$$

Using Replacement and Theorem 66.26, form the set:

$$\alpha = \{\mathrm{ord}(B, R) : \langle B, R\rangle \in C\}.$$

By Corollary 66.19, α is an ordinal, since it is a transitive set of ordinals. After all, if $\gamma \in \beta \in \alpha$, then $\beta = \mathrm{ord}(B, R)$ for some $B \subseteq R$, whereupon $\gamma = \mathrm{ord}(B_b, R_b)$ for some $b \in B$ by Lemma 66.10, so that $\gamma \in \alpha$.

For reductio, suppose there is an injection $f : \alpha \to A$. Then, where:

$$B = \mathrm{ran}(f)$$
$$R = \{\langle f(\alpha), f(\beta)\rangle \in A \times A : \alpha \in \beta\}.$$

Clearly $\alpha = \mathrm{ord}(B, R)$ and $\langle B, R\rangle \in C$. So $\alpha \in \alpha$, which is a contradiction. □

This entails a deep result:

Theorem 72.4 (*in* ZF). *The following claims are equivalent:*

1. *The Axiom of Well-Ordering* — *sth:choice:hartogs: equivwo*
2. *Either $A \preceq B$ or $B \preceq A$, for any sets A and B* — *sth:choice:hartogs: equivcompare*

Proof. *(1) ⇒ (2).* Fix A and B. Invoking (1), there are well-orderings $\langle A, R\rangle$ and $\langle B, S\rangle$. Invoking Theorem 66.26, let $f : \alpha \to \langle A, R\rangle$ and $g : \beta \to \langle B, S\rangle$ be isomorphisms. By Proposition 66.22, either $\alpha \subseteq \beta$ or $\beta \subseteq \alpha$. If $\alpha \subseteq \beta$, then $g \circ f^{-1} : A \to B$ is an injection, and hence $A \preceq B$; similarly, if $\beta \subseteq \alpha$ then $B \preceq A$.

(2) ⇒ (1). Fix A; by Lemma 72.3 there is some ordinal β such that $\beta \not\preceq A$. Invoking (2), we have $A \preceq \beta$. So there is some injection $f : A \to \beta$, and we can use this injection to well-order the elements of A, by defining an order $\{\langle a, b\rangle \in A \times A : f(a) \in f(b)\}$. □

As an immediate consequence: if Well-Ordering fails, then some sets are *literally incomparable* with regard to their size. So, if Well-Ordering fails, then transfinite cardinal arithmetic will be messy. For example, we will have to abandon the idea that if A and B are infinite then $A \sqcup B \approx A \times B \approx M$, where M is the larger of A and B (see Theorem 71.11). The problem is simple: if we cannot *compare* the size of A and B, then it is nonsensical to ask which is larger.

---

content/set-theory/choice/wellorderingproblem.tex

72.4 The Well-Ordering Problem

sth:choice:woproblem:sec

Evidently rather a lot hangs on whether we accept Well-Ordering. But the discussion of this principle has tended to focus on an equivalent principle, the Axiom of Choice. So we will now turn our attention to that (and prove the equivalence).

In 1883, Cantor expressed his support for the Axiom of Well-Ordering, calling it "a law of thought which appears to me to be fundamental, rich in its consequences, and particularly remarkable for its general validity" (cited in Potter 2004, p. 243). But Cantor ultimately became convinced that the "Axiom" was in need of proof. So did the mathematical community.

The problem was "solved" by Zermelo in 1904. To explain his solution, we need some definitions.

Definition 72.5. A function f is a *choice function* iff $f(x) \in x$ for all $x \in \mathrm{dom}(f)$. We say that f is a *choice function for* A iff f is a choice function with $\mathrm{dom}(f) = A \setminus \{\emptyset\}$.

Intuitively, for every (non-empty) set $x \in A$, a choice function for A *chooses* a particular element, $f(x)$, from x. The Axiom of Choice is then:

Axiom (Choice). Every set has a choice function.

Zermelo showed that Choice entails well-ordering, and vice versa:

sth:choice:woproblem:thmwochoice

Theorem 72.6 (in ZF). *Well-Ordering and Choice are equivalent.*

Proof. Left-to-right. Let A be a set of sets. Then $\bigcup A$ exists by the Axiom of Union, and so by Well-Ordering there is some $<$ which well-orders $\bigcup A$. Now let $f(x) =$ the $<$-least member of x. This is a choice function for A.

Right-to-left. Fix A. By Choice, there is a choice function, f, for $\wp(A) \setminus \{\emptyset\}$. Using Transfinite Recursion, define a function:

$$g(0) = f(A)$$
$$g(\alpha) = \begin{cases} \text{stop!} & \text{if } A = g[\alpha] \\ f(A \setminus g[\alpha]) & \text{otherwise} \end{cases}$$

The indication to "stop!" is just a shorthand for what would otherwise be a more long-winded definition. That is, when $A = g[\alpha]$ for the first time, let $g(\delta) = A$ for all $\delta \leq \alpha$. Now, in the first instance, we can only be sure that this defines a *term* (see the remarks after Theorem 67.4); but we will show that we indeed have a function.

Since f is a choice function, for each α (when defined) we have $g(\alpha) = f(A \setminus g[\alpha]) \in A \setminus g[\alpha]$; i.e., $g(\alpha) \notin g[\alpha]$. So if $g(\alpha) = g(\beta)$ then $g(\beta) \notin g[\alpha]$, i.e., $\beta \notin \alpha$, and similarly $\alpha \notin \beta$. So $\alpha = \beta$, by Trichotomy. So g is injective.

Next, observe that we do stop!, i.e. that there is some (least) ordinal α such that $A = g[\alpha]$. For suppose otherwise; then as g is injective we would have

$\alpha \prec \wp(A) \setminus \{\emptyset\}$ for every ordinal α, contradicting Lemma 72.3. Hence also $\mathrm{ran}(g) = A$.

Assembling these facts, g is a bijection from some ordinal to A. Now g can be used to well-order A. □

So Well-Ordering and Choice stand or fall together. But the question remains: do they stand or fall?

---

content/set-theory/choice/countablechoice.tex

72.5 Countable Choice

sth:choice:countablechoice:sec

It is easy to prove, without any use of Choice/Well-Ordering, that:

Lemma 72.7 (in $\mathbf{Z}^-$). *Every finite set has a choice function.*

Proof. Let $a = \{b_1, \ldots, b_n\}$. Suppose for simplicity that each $b_i \neq \emptyset$. So there are objects $c_1, \ldots, c_n$ such that $c_1 \in b_1, \ldots, c_n \in b_n$. Now by Proposition 65.5, the set $\{\langle b_1, c_1\rangle, \ldots, \langle b_n, c_n\rangle\}$ exists; and this is a choice function for a. □

But matters get murkier as soon as we consider infinite sets. For example, consider this "minimal" extension to the above:

> *Countable Choice.* Every *countable* set has a choice function.

This is a special case of Choice. And it transpires that this principle was invoked fairly frequently, without an obvious awareness of its use. Here are two nice examples.[2]

Example 72.8. Here is a natural thought: for any set A, either $\omega \preceq A$, or $A \approx n$ for some $n \in \omega$. This is one way to state the intuitive idea, that every set is either finite or infinite. Cantor, and many other mathematicians, made this claim without proving it. Cautious as we are, we proved this in Theorem 70.7. But in that proof we were working in **ZFC**, since we were assuming that any set A can be well-ordered, and hence that $|A|$ is guaranteed to exist. That is: we explicitly assumed Choice.

In fact, Dedekind (1888) offered his own proof of this claim, as follows:

Theorem 72.9 (in $\mathbf{Z}^- +$ Countable Choice). *For any A, either $\omega \preceq A$ or $A \approx n$ for some $n \in \omega$.*

Proof. Suppose $A \not\approx n$ for all $n \in \omega$. Then in particular for each $n < \omega$ there is subset $A_n \subseteq A$ with exactly 2^n elements. Using this sequence $A_0, A_1, A_2, \ldots$, we define for each n:

$$B_n = A_n \setminus \bigcup_{i<n} A_i.$$

[2] Due to Potter (2004, §9.4) and Luca Incurvati.

Now note the following

$$\begin{aligned}\left|\bigcup_{i<n} A_n\right| &\leq |A_0| + |A_1| + \ldots + |A_{n-1}| \\ &= 1 + 2 + \ldots + 2^{n-1} \\ &= 2^n - 1 \\ &< 2^n = |A_n|\end{aligned}$$

Hence each B_n has at least one member, c_n. Moreover, the B_ns are pairwise disjoint; so if $c_n = c_m$ then $n = m$. But every $c_n \in A$. So the function $f(n) = c_n$ is an injection $\omega \to A$. □

Dedekind did not flag that he had used Countable Choice. But, did *you* spot its use? Look again. (Really: *look again.*)

The proof used Countable Choice twice. We used it once, to obtain our sequence of sets A_0, A_1, A_2, ... We then used it again to select our elements c_n from each B_n. Moreover, this use of Choice is ineliminable. Cohen (1966, p. 138) proved that the result fails if we have no version of Choice. That is: it is consistent with **ZF** that there are sets which are *incomparable* with ω.

Example 72.10. In 1878, Cantor stated that a countable union of countable sets is countable. He did not present a proof, perhaps indicating that he took the proof to be obvious. Now, cautious as we are, we proved a more general version of this result in Proposition 71.12. But our proof explicitly assumed Choice. And even the proof of the less general result requires Countable Choice.

Theorem 72.11 (in $\mathbf{Z}^- +$ Countable Choice). *If A_n is countable for each $n \in \omega$, then $\bigcup_{n<\omega} A_n$ is countable.*

Proof. Without loss of generality, suppose that each $A_n \neq \emptyset$. So for each $n \in \omega$ there is a surjection $f_n : \omega \to A_n$. Define $f : \omega \times \omega \to \bigcup_{n<\omega} A_n$ by $f(m, n) = f_n(m)$. The result follows because $\omega \times \omega$ is countable (Proposition 4.12) and f is a surjection. □

Did you spot the use of the Countable Choice? It is used to choose our sequence of functions f_0, f_1, f_2, ...[3] And again, the result fails in the absence of any Choice principle. Specifically, Feferman and Levy (1963) proved that it is consistent with **ZF** that a countable union of countable sets has cardinality $\beth_1$. But here is a much funnier statement of the point, from Russell:

> This is illustrated by the millionaire who bought a pair of socks whenever he bought a pair of boots, and never at any other time, and who had such a passion for buying both that at last he had $\aleph_0$ pairs of boots and $\aleph_0$ pairs of socks... Among boots we can

[3] A similar use of Choice occurred in Proposition 71.12, when we gave the instruction "For each $\beta \in \mathfrak{a}$, fix an injection f_β".

> distinguish right and left, and therefore we can make a selection of one out of each pair, namely, we can choose all the right boots or all the left boots; but with socks no such principle of selection suggests itself, and we cannot be sure, unless we assume the multiplicative axiom [i.e., in effect Choice], that there is any class consisting of one sock out of each pair. (Russell, 1919, p. 126)

In short, some form of Choice is needed to prove the following: If you have countably many pairs of socks, then you have (only) countably many socks. And in fact, without Countable Choice (or something equivalent), a countable union of countable sets can fail to be countable.

The moral is that Countable Choice was used repeatedly, without much awareness of its users. The philosophical question is: How could we *justify* Countable Choice?

An attempt at an intuitive justification might invoke an appeal to a supertask. Suppose we make the first choice in $1/2$ a minute, our second choice in $1/4$ a minute, ..., our n-th choice in $1/2^n$ a minute, ... Then within 1 minute, we will have made an ω-sequence of choices, and defined a choice function.

But what, really, could such a thought-experiment tell us? For a start, it relies upon taking this idea of "choosing" rather literally. For another, it seems to bind up mathematics in metaphysical possibility.

More important: it is not going to give us any justification for Choice *tout court*, rather than *mere* Countable Choice. For if we need *every* set to have a choice function, then we'll need to be able to perform a "supertask of arbitrary ordinal length." Bluntly, that idea is laughable.

---

content/set-theory/choice/justifications.tex

72.6 Intrinsic Considerations about Choice

sth:choice:justifications:sec

The broader question, then, is whether Well-Ordering, or Choice, or indeed the comparability of all sets as regards their size—it doesn't matter which—can be justified.

Here is an attempted *intrinsic* justification. Back in section 65.1, we introduced several principles about the hierarchy. One of these is worth restating:

> *Stages-accumulate.* For any stage S, and for any sets which were formed *before* stage S: a set is formed at stage S whose members are exactly those sets. Nothing else is formed at stage S.

In fact, many authors have suggested that the Axiom of Choice can be justified via (something like) this principle. We will briefly provide a gloss on that approach.

We will start with a simple little result, which offers *yet another* equivalent for Choice:

sth:choice:justifications: choiceset

Theorem 72.12 (in ZF). *Choice is equivalent to the following principle. If the elements of A are disjoint and non-empty, then there is some C such that $C \cap x$ is a singleton for every $x \in A$. (We call such a C a choice set for A.)*

The proof of this result is straightforward, and we leave it as an exercise for the reader.

Problem 72.1. Prove Theorem 72.12. If you struggle, you can find a proof in (Potter, 2004, pp. 242–3).

The essential point is that a choice set for A is just the range of a choice function for A. So, to justify Choice, we can simply try to justify its equivalent formulation, in terms of the existence of choice sets. And we will now try to do exactly that.

Let A's elements be disjoint and non-empty. By *Stages-are-key* (see section 65.1), A is formed at some stage S. Note that all the elements of $\bigcup A$ are available before stage S. Now, by *Stages-accumulate*, for *any* sets which were formed before S, a set is formed whose members are exactly those sets. Otherwise put: every *possible* collections of earlier-available sets will exist at S. But it is certainly *possible* to select objects which could be formed into a choice set for A; that is just some very specific subset of $\bigcup A$. So: some such choice set exists, as required.

Well, that's a *very* quick attempt to offer a justification of Choice on intrinsic grounds. But, to pursue this idea further, you should read Potter's (2004, §14.8) neat development of it.

---

content/set-theory/choice/banach.tex

72.7 The Banach–Tarski Paradox

sth:choice:banach: sec

We might also attempt to justify Choice, as Boolos attempted to justify Replacement, by appealing to *extrinsic* considerations (see section 68.3). After all, adopting Choice has many desirable consequences: the ability to compare every cardinal; the ability to well-order every set; the ability to treat cardinals as a particular kind of ordinal; etc.

Sometimes, however, it is claimed that Choice has *undesirable* consequences. Mostly, this is due to a result by Banach and Tarski (1924).

Theorem 72.13 (Banach–Tarski Paradox (in ZFC)). *Any ball can be decomposed into finitely many pieces, which can be reassembled (by rotation and transportation) to form two copies of that ball.*

At first glance, this is a bit amazing. Clearly the two balls have *twice* the volume of the original ball. But rigid motions—rotation and transportation—do not change volume. So it looks as if Banach–Tarski allows us to magick new matter into existence.

It gets worse.[4] Similar reasoning shows that a pea can be cut into finitely many pieces, which can then be reassembled (by rotation and transportation) to form an entity the shape and size of Big Ben.

None of this, however, holds in **ZF** on its own.[5] So we face a decision: reject Choice, or learn to live with the "paradox".

We're going to suggest that we should learn to live with the "paradox". Indeed, we don't think it's much of a paradox at all. In particular, we don't see why it is any more or less paradoxical than any of the following results:[6]

1. There are as many points in the interval $(0, 1)$ as in $\mathbb{R}$.
   Proof: consider $\tan(\pi(r - 1/2))$.

2. There are as many points in a line as in a square.
   See section 76.3 and section 76.5.

3. There are space-filling curves.
   See section 76.3 and section 76.6.

None of these three results require Choice. Indeed, we now just regard them as surprising, lovely, bits of mathematics. Maybe we should adopt the same attitude to the Banach–Tarski Paradox.

To be sure, a technical observation is required here; but it only requires keeping a level head. Rigid motions preserve volume. Consequently, the five[7] pieces into which the ball is decomposed cannot all be *measurable*. Roughly put, then, it makes no sense to assign a volume to these individual pieces. You should think of these as unpicturable, "infinite scatterings" of points. Now, maybe it is "weird" to conceive of such "infinitely scattered" sets. But their existence seems to fall out from the injunction, embodied in *Stages-accumulate*, that you should form *all possible* collections of earlier-available sets.

If none of that convinces, here is a final (extrinsic) argument in favour of embracing the Banach–Tarski Paradox. It immediately entails the best math joke of all time:

> *Question.* What's an anagram of "Banach–Tarski"?
>
> *Answer.* "Banach–Tarski Banach–Tarski".

---

content/set-theory/choice/vitali.tex

[4] See Tomkowicz and Wagon (2016, Theorem 3.12).

[5] Though Banach–Tarski can be proved with principles which are strictly weaker than Choice; see Tomkowicz and Wagon (2016, 303).

[6] Potter (2004, 276–7), Weston (2003, 16), Tomkowicz and Wagon (2016, 31, 308–9), make similar points, using other examples.

[7] We stated the Paradox in terms of "finitely many pieces". In fact, Robinson (1947) proved that the decomposition can be achieved with *five* pieces (but no fewer). For a proof, see Tomkowicz and Wagon (2016, pp. 66–7).

72.8 Appendix: Vitali's Paradox

sth:choice:vitali:sec

To get a real sense of whether the Banach-Tarski construction is acceptable or not, we should examine its *proof*. Unfortunately, that would require much more algebra than we can present here. However, we can offer some quick remarks which might shed some insight on the proof of Banach-Tarski,[8] by focussing on the following result:

sth:choice:vitali:vitaliparadox

Theorem 72.14 (Vitali's Paradox (in ZFC)). *Any circle can be decomposed into countably many pieces, which can be reassembled (by rotation and transportation) to form two copies of that circle.*

Vitali's Paradox is much easier to prove than the Banach–Tarski Paradox. We have called it "Vitali's Paradox", since it follows from Vitali's 1905 construction of an unmeasurable set. But the set-theoretic aspects of the proof of Vitali's Paradox and the Banach-Tarski Paradox are very similar. The essential difference between the results is just that Banach-Tarski considers a *finite* decomposition, whereas Vitali's Paradox considers a *countably infinite* decomposition. As Weston (2003) puts it, Vitali's Paradox "is certainly not nearly as striking as the Banach–Tarski paradox, but it does illustrate that geometric paradoxes can happen even in 'simple' situations."

Vitali's Paradox concerns a two-dimensional figure, a circle. So we will work on the plane, $\mathbb{R}^2$. Let R be the set of (clockwise) rotations of points around the origin by *rational* radian values between $[0, 2\pi)$. Here are some algebraic facts about R (if you don't understand the statement of the result, the proof will make its meaning clear):

sth:choice:vitali:rotationsgroupabelian

Lemma 72.15. *R forms an abelian group under composition of functions.*

Proof. Writing 0_R for the rotation by 0 radians, this is an identity element for R, since $\rho \circ 0_R = 0_R \circ \rho = \rho$ for any $\rho \in R$.

Every element has an inverse. Where $\rho \in R$ rotates by r radians, $\rho^{-1} \in R$ rotates by $2\pi - r$ radians, so that $\rho \circ \rho^{-1} = 0_R$.

Composition is associative: $(\tau \circ \sigma) \circ \rho = \tau \circ (\sigma \circ \rho)$ for any $\rho, \sigma, \tau \in R$

Composition is commutative: $\sigma \circ \rho = \rho \circ \sigma$ for any $\rho, \sigma \in R$. □

In fact, we can split our group R in half, and then use either half to recover the whole group:

sth:choice:vitali:disjointgroup

Lemma 72.16. *There is a partition of R into two disjoint sets, R_1 and R_2, both of which are a basis for R.*

Proof. Let R_1 consist of the rotations by rational radian values in $[0, \pi)$; let $R_2 = R \setminus R_1$. By elementary algebra, $\{\rho \circ \rho : \rho \in R_1\} = R$. A similar result can be obtained for R_2. □

[8] For a much fuller treatment, see Weston (2003) or Tomkowicz and Wagon (2016).

We will use this fact about groups to establish Theorem 72.14. Let $\mathbf{S}$ be the unit circle, i.e., the set of points exactly 1 unit away from the origin of the plane, i.e., $\{\langle r, s\rangle \in \mathbb{R}^2 : \sqrt{r^2 + s^2} = 1\}$. We will split $\mathbf{S}$ into parts by considering the following relation on $\mathbf{S}$:

$$r \sim s \textit{ iff } (\exists \rho \in R)\rho(r) = s.$$

That is, the points of $\mathbf{S}$ are linked by this relation iff you can get from one to the other by a rational-valued rotation about the origin. Unsurprisingly:

Lemma 72.17. *$\sim$ is an equivalence relation.*

Proof. Trivial, using Lemma 72.15. □

We now invoke Choice to obtain a set, C, containing exactly one member from each equivalence class of $\mathbf{S}$ under $\sim$. That is, we consider a choice function f on the set of equivalence classes,[9]

$$E = \{[r]_\sim : r \in \mathbf{S}\},$$

and let $C = \mathrm{ran}(f)$. For each rotation $\rho \in R$, the set $\rho[C]$ consists of the points obtained by applying the rotation ρ to each point in C. These next two results show that these sets cover the circle completely and without overlap:

Lemma 72.18. $\mathbf{S} = \bigcup_{\rho \in R} \rho[C]$. *sth:choice:vitali: vitalicover*

Proof. Fix $s \in \mathbf{S}$; there is some $r \in C$ such that $r \in [s]_\sim$, i.e., $r \sim s$, i.e., $\rho(r) = s$ for some $\rho \in R$. □

Lemma 72.19. *If $\rho_1 \neq \rho_2$ then $\rho_1[C] \cap \rho_2[C] = \emptyset$.* *sth:choice:vitali: vitalinooverlap*

Proof. Suppose $s \in \rho_1[C] \cap \rho_2[C]$. So $s = \rho_1(r_1) = \rho_2(r_2)$ for some $r_1, r_2 \in C$. Hence $\rho_2^{-1}(\rho_1(r_1)) = r_2$, and $\rho_2^{-1} \circ \rho_1 \in R$, so $r_1 \sim r_2$. So $r_1 = r_2$, as C selects exactly one member from each equivalence class under $\sim$. So $s = \rho_1(r_1) = \rho_2(r_1)$, and hence $\rho_1 = \rho_2$. □

We now apply our earlier algebraic facts to our circle:

Lemma 72.20. *There is a partition of $\mathbf{S}$ into two disjoint sets, D_1 and D_2, such that D_1 can be partitioned into countably many sets which can be rotated to form a copy of $\mathbf{S}$ (and similarly for D_2).* *sth:choice:vitali: pseudobanachtarski*

[9] Since R is enumerable, each element of E is enumerable. Since $\mathbf{S}$ is non-enumerable, it follows from Lemma 72.18 and Proposition 71.12 that E is non-enumerable. So this is a use of *un*countable Choice.

Proof. Using R_1 and R_2 from Lemma 72.16, let:

$$D_1 = \bigcup_{\rho \in R_1} \rho[C] \qquad\qquad D_2 = \bigcup_{\rho \in R_2} \rho[C]$$

This is a partition of $\mathbf{S}$, by Lemma 72.18, and D_1 and D_2 are disjoint by Lemma 72.19. By construction, D_1 can be partitioned into countably many sets, $\rho[C]$ for each $\rho \in R_1$. And these can be rotated to form a copy of $\mathbf{S}$, since $\mathbf{S} = \bigcup_{\rho \in R} \rho[C] = \bigcup_{\rho \in R_1} (\rho \circ \rho)[C]$ by Lemma 72.16 and Lemma 72.18. The same reasoning applies to D_2. □

This immediately entails Vitali's Paradox. For we can generate *two* copies of $\mathbf{S}$ from $\mathbf{S}$, just by splitting it up into countably many pieces (the various $\rho[C]$'s) and then rigidly moving them (simply rotate each piece of D_1, and first transport and then rotate each piece of D_2).

Let's recap the proof-strategy. We started with some algebraic facts about the group of rotations on the plane. We used this group to partition $\mathbf{S}$ into equivalence classes. We then arrived at a "paradox", by using Choice to select elements from each class.

We use exactly the same strategy to prove Banach–Tarski. The main difference is that the algebraic facts used to prove Banach–Tarski are significantly more complicated than those used to prove Vitali's Paradox. But those algebraic facts have nothing to do with Choice. We will summarise them quickly.

To prove Banach–Tarski, we start by establishing an analogue of Lemma 72.16: any *free group* can be split into four pieces, which intuitively we can "move around" to recover two copies of the whole group.[10] We then show that we can use two particular rotations around the origin of $\mathbb{R}^3$ to generate a free group of rotations, F.[11] (No Choice yet.) We now regard points on the surface of the sphere as "similar" iff one can be obtained from the other by a rotation in F. We then *use Choice* to select exactly one point from each equivalence class of "similar" points. Applying our division of F to the surface of the sphere, as in Lemma 72.20, we split that surface into four pieces, which we can "move around" to obtain two copies of the surface of the sphere. And this establishes (Hausdorff, 1914):

Theorem 72.21 (Hausdorff's Paradox (in ZFC)). *The surface of any sphere can be decomposed into finitely many pieces, which can be reassembled (by rotation and transportation) to form two disjoint copies of that sphere.*

A couple of further algebraic tricks are needed to obtain the full Banach-Tarski Theorem (which concerns not just the sphere's surface, but its interior too). Frankly, however, this is just icing on the algebraic cake. Hence Weston writes:

[10] The fact that we can use *four* pieces is due to Robinson (1947). For a recent proof, see Tomkowicz and Wagon (2016, Theorem 5.2). We follow Weston (2003, p. 3) in describing this as "moving" the pieces of the group.

[11] See Tomkowicz and Wagon (2016, Theorem 2.1).

> [...] the result on free groups is the *key step* in the proof of the Banach-Tarski paradox. From this point of view, the Banach-Tarski paradox is not a statement about $\mathbb{R}^3$ so much as it is a statement about the complexity of the group [of translations and rotations in $\mathbb{R}^3$]. (Weston, 2003, p. 16)

That is: whether we can offer a *finite* decomposition (as in Banach–Tarski) or a *countably infinite* decomposition (as in Vitali's Paradox) comes down to certain group-theoretic facts about working in two-dimension or three-dimensions.

Admittedly, this last observation slightly spoils the joke at the end of section 72.7. Since it is two dimensional, "Banach-Tarski" must be divided into a countable *infinity* of pieces, if one wants to rearrange those pieces to form "Banach-Tarski Banach-Tarski". To repair the joke, one must write in three dimensions. We leave this as an exercise for the reader.

One final comment. In section 72.7, we mentioned that the "pieces" of the sphere one obtains cannot be *measurable*, but must be unpicturable "infinite scatterings". The same is true of our use of Choice in obtaining Lemma 72.20. And this is all worth explaining.

Again, we must sketch some background (but this is *just* a sketch; you may want to consult a textbook entry on *measure*). To define a measure for a set X is to assign a value $\mu(E) \in \mathbb{R}$ for each E in some "σ-algebra" on X. Details here are not essential, except that the function μ must obey the principle of countable additivity: the measure of a countable union of disjoint sets is the sum of their individual measures, i.e., $\mu(\bigcup_{n<\omega} X_n) = \sum_{n<\omega} \mu(X_n)$ whenever the X_ns are disjoint. To say that a set is "unmeasurable" is to say that no measure can be suitably assigned. Now, using our R from before:

Corollary 72.22 (Vitali). *Let μ be a measure such that $\mu(\mathbf{S}) = 1$, and such that $\mu(X) = \mu(Y)$ if X and Y are congruent. Then $\rho[C]$ is unmeasurable for all $\rho \in R$.*

Proof. For reductio, suppose otherwise. So let $\mu(\sigma[C]) = r$ for some $\sigma \in R$ and some $r \in \mathbb{R}$. For any $\rho \in C$, $\rho[C]$ and $\sigma[C]$ are congruent, and hence $\mu(\rho[C]) = r$ for any $\rho \in C$. By Lemma 72.18 and Lemma 72.19, $\mathbf{S} = \bigcup_{\rho\in R} \rho[C]$ is a countable union of pairwise disjoint sets. So countable additivity dictates that $\mu(\mathbf{S}) = 1$ is the sum of the measures of each $\rho[C]$, i.e.,

$$1 = \mu(\mathbf{S}) = \sum_{\rho\in R} \mu(\rho[C]) = \sum_{\rho\in R} r$$

But if $r = 0$ then $\sum_{\rho\in R} r = 0$, and if $r > 0$ then $\sum_{\rho\in R} r = \infty$. □

Part XVI

Methods

This part covers general and methodological material, especially explanations of various proof methods a non-mathematics student may be unfamiliar with. It currently contains a chapter on how to write proofs, and a chapter on induction, but additional sections for those, exercises, and a chapter on mathematical terminology is also planned.

Chapter 73

Proofs

---

content/methods/proofs/introduction.tex

73.1 Introduction

mth:prf:int:sec

Based on your experiences in introductory logic, you might be comfortable with a derivation system—probably a natural deduction or Fitch style derivation system, or perhaps a proof-tree system. You probably remember doing proofs in these systems, either proving a formula or show that a given argument is valid. In order to do this, you applied the rules of the system until you got the desired end result. In reasoning *about* logic, we also prove things, but in most cases we are not using a derivation system. In fact, most of the proofs we consider are done in English (perhaps, with some symbolic language thrown in) rather than entirely in the language of first-order logic. When constructing such proofs, you might at first be at a loss—how do I prove something without a derivation system? How do I start? How do I know if my proof is correct?

Before attempting a proof, it's important to know what a proof is and how to construct one. As implied by the name, a *proof* is meant to show that something is true. You might think of this in terms of a dialogue—someone

asks you if something is true, say, if every prime other than two is an odd number. To answer "yes" is not enough; they might want to know *why*. In this case, you'd give them a proof.

In everyday discourse, it might be enough to gesture at an answer, or give an incomplete answer. In logic and mathematics, however, we want rigorous proof—we want to show that something is true beyond *any* doubt. This means that every step in our proof must be justified, and the justification must be cogent (i.e., the assumption you're using is actually assumed in the statement of the theorem you're proving, the definitions you apply must be correctly applied, the justifications appealed to must be correct inferences, etc.).

Usually, we're proving some statement. We call the statements we're proving by various names: propositions, theorems, lemmas, or corollaries. A proposition is a basic proof-worthy statement: important enough to record, but perhaps not particularly deep nor applied often. A theorem is a significant, important proposition. Its proof often is broken into several steps, and sometimes it is named after the person who first proved it (e.g., Cantor's Theorem, the Löwenheim–Skolem theorem) or after the fact it concerns (e.g., the completeness theorem). A lemma is a proposition or theorem that is used in the proof of a more important result. Confusingly, sometimes lemmas are important results in themselves, and also named after the person who introduced them (e.g., Zorn's Lemma). A corollary is a result that easily follows from another one.

A statement to be proved often contains assumptions that clarify which kinds of things we're proving something about. It might begin with "Let φ be a formula of the form $\psi \to \chi$" or "Suppose $\Gamma \vdash \varphi$" or something of the sort. These are *hypotheses* of the proposition, theorem, or lemma, and you may assume these to be true in your proof. They restrict what we're proving, and also introduce some names for the objects we're talking about. For instance, if your proposition begins with "Let φ be a formula of the form $\psi \to \chi$," you're proving something about all formulas of a certain sort only (namely, conditionals), and it's understood that $\psi \to \chi$ is an arbitrary conditional that your proof will talk about.

---

content/methods/proofs/starting-proofs.tex

73.2 Starting a Proof

mth:prf:str:sec

But where do you even start?

You've been given something to prove, so this should be the last thing that is mentioned in the proof (you can, obviously, *announce* that you're going to prove it at the beginning, but you don't want to use it as an assumption). Write what you are trying to prove at the bottom of a fresh sheet of paper—this way you don't lose sight of your goal.

Next, you may have some assumptions that you are able to use (this will be made clearer when we talk about the *type* of proof you are doing in the next

section). Write these at the top of the page and make sure to flag that they are assumptions (i.e., if you are assuming p, write "assume that p," or "suppose that p"). Finally, there might be some definitions in the question that you need to know. You might be told to use a specific definition, or there might be various definitions in the assumptions or conclusion that you are working towards. *Write these down and ensure that you understand what they mean.*

How you set up your proof will also be dependent upon the form of the question. The next section provides details on how to set up your proof based on the type of sentence.

---

content/methods/proofs/using-definitions.tex

73.3 Using Definitions

mod:prf:def:sec

We mentioned that you must be familiar with all definitions that may be used in the proof, and that you can properly apply them. This is a really important point, and it is worth looking at in a bit more detail. Definitions are used to abbreviate properties and relations so we can talk about them more succinctly. The introduced abbreviation is called the *definiendum*, and what it abbreviates is the *definiens*. In proofs, we often have to go back to how the definiendum was introduced, because we have to exploit the logical structure of the definiens (the long version of which the defined term is the abbreviation) to get through our proof. By unpacking definitions, you're ensuring that you're getting to the heart of where the logical action is.

We'll start with an example. Suppose you want to prove the following:

Proposition 73.1. *For any sets A and B, $A \cup B = B \cup A$.*

In order to even start the proof, we need to know what it means for two sets to be identical; i.e., we need to know what the "=" in that equation means for sets. Sets are defined to be identical whenever they have the same elements. So the definition we have to unpack is:

Definition 73.2. Sets A and B are *identical*, $A = B$, iff every element of A is an element of B, and vice versa.

This definition uses A and B as placeholders for arbitrary sets. What it defines—the *definiendum*—is the expression "$A = B$" by giving the condition under which $A = B$ is true. This condition—"every element of A is an element of B, and vice versa"—is the *definiens*.[1] The definition specifies that $A = B$ is true if, and only if (we abbreviate this to "iff") the condition holds.

When you apply the definition, you have to match the A and B in the definition to the case you're dealing with. In our case, it means that in order

[1] In this particular case—and very confusingly!—when $A = B$, the sets A and B are just one and the same set, even though we use different letters for it on the left and the right side. But the ways in which that set is picked out may be different, and that makes the definition non-trivial.

for $A \cup B = B \cup A$ to be true, each $z \in A \cup B$ must also be in $B \cup A$, and vice versa. The expression $A \cup B$ in the proposition plays the role of A in the definition, and $B \cup A$ that of B. Since A and B are used both in the definition and in the statement of the proposition we're proving, but in different uses, you have to be careful to make sure you don't mix up the two. For instance, it would be a mistake to think that you could prove the proposition by showing that every element of A is an element of B, and vice versa—that would show that $A = B$, not that $A \cup B = B \cup A$. (Also, since A and B may be any two sets, you won't get very far, because if nothing is assumed about A and B they may well be different sets.)

Within the proof we are dealing with set-theoretic notions such as union, and so we must also know the meanings of the symbol $\cup$ in order to understand how the proof should proceed. And sometimes, unpacking the definition gives rise to further definitions to unpack. For instance, $A \cup B$ is defined as $\{z : z \in A \text{ or } z \in B\}$. So if you want to prove that $x \in A \cup B$, unpacking the definition of $\cup$ tells you that you have to prove $x \in \{z : z \in A \text{ or } z \in B\}$. Now you also have to remember that $x \in \{z : \ldots z \ldots\}$ iff $\ldots x \ldots$. So, further unpacking the definition of the $\{z : \ldots z \ldots\}$ notation, what you have to show is: $x \in A$ or $x \in B$. So, "every element of $A \cup B$ is also an element of $B \cup A$" really means: "for every x, if $x \in A$ or $x \in B$, then $x \in B$ or $x \in A$." If we fully unpack the definitions in the proposition, we see that what we have to show is this:

Proposition 73.3. *For any sets A and B: (a) for every x, if $x \in A$ or $x \in B$, then $x \in B$ or $x \in A$, and (b) for every x, if $x \in B$ or $x \in A$, then $x \in A$ or $x \in B$.*

What's important is that unpacking definitions is a necessary part of constructing a proof. Properly doing it is sometimes difficult: you must be careful to distinguish and match the variables in the definition and the terms in the claim you're proving. In order to be successful, you must know what the question is asking and what all the terms used in the question mean—you will often need to unpack more than one definition. In simple proofs such as the ones below, the solution follows almost immediately from the definitions themselves. Of course, it won't always be this simple.

Problem 73.1. Suppose you are asked to prove that $A \cap B \neq \emptyset$. Unpack all the definitions occurring here, i.e., restate this in a way that does not mention "$\cap$", "$=$", or "$\emptyset$".

---

content/methods/proofs/inference-patterns.tex

73.4 Inference Patterns

mth:prf:pat:sec

Proofs are composed of individual inferences. When we make an inference, we typically indicate that by using a word like "so," "thus," or "therefore."

The inference often relies on one or two facts we already have available in our proof—it may be something we have assumed, or something that we've concluded by an inference already. To be clear, we may label these things, and in the inference we indicate what other statements we're using in the inference. An inference will often also contain an explanation of *why* our new conclusion follows from the things that come before it. There are some common patterns of inference that are used very often in proofs; we'll go through some below. Some patterns of inference, like proofs by induction, are more involved (and will be discussed later).

We've already discussed one pattern of inference: unpacking, or applying, a definition. When we unpack a definition, we just restate something that involves the definiendum by using the definiens. For instance, suppose that we have already established in the course of a proof that $D = E$ (a). Then we may apply the definition of $=$ for sets and infer: "Thus, by definition from (a), every element of D is an element of E and vice versa."

Somewhat confusingly, we often do not write the justification of an inference when we actually make it, but before. Suppose we haven't already proved that $D = E$, but we want to. If $D = E$ is the conclusion we aim for, then we can restate this aim also by applying the definition: to prove $D = E$ we have to prove that every element of D is an element of E and vice versa. So our proof will have the form: (a) prove that every element of D is an element of E; (b) every element of E is an element of D; (c) therefore, from (a) and (b) by definition of $=$, $D = E$. But we would usually not write it this way. Instead we might write something like,

> We want to show $D = E$. By definition of $=$, this amounts to showing that every element of D is an element of E and vice versa.
>
> (a) ... (a proof that every element of D is an element of E) ...
>
> (b) ... (a proof that every element of E is an element of D) ...

Using a Conjunction

Perhaps the simplest inference pattern is that of drawing as conclusion one of the conjuncts of a conjunction. In other words: if we have assumed or already proved that p and q, then we're entitled to infer that p (and also that q). This is such a basic inference that it is often not mentioned. For instance, once we've unpacked the definition of $D = E$ we've established that every element of D is an element of E and vice versa. From this we can conclude that every element of E is an element of D (that's the "vice versa" part).

Proving a Conjunction

Sometimes what you'll be asked to prove will have the form of a conjunction; you will be asked to "prove p and q." In this case, you simply have to do two things: prove p, and then prove q. You could divide your proof into two sections, and for clarity, label them. When you're making your first notes, you

might write "(1) Prove p" at the top of the page, and "(2) Prove q" in the middle of the page. (Of course, you might not be explicitly asked to prove a conjunction but find that your proof requires that you prove a conjunction. For instance, if you're asked to prove that $D = E$ you will find that, after unpacking the definition of $=$, you have to prove: every element of D is an element of E *and* every element of E is an element of D).

Proving a Disjunction

When what you are proving takes the form of a disjunction (i.e., it is an statement of the form "p or q"), it is enough to show that one of the disjuncts is true. However, it basically never happens that either disjunct just follows from the assumptions of your theorem. More often, the assumptions of your theorem are themselves disjunctive, or you're showing that all things of a certain kind have one of two properties, but some of the things have the one and others have the other property. This is where proof by cases is useful (see below).

Conditional Proof

Many theorems you will encounter are in conditional form (i.e., show that if p holds, then q is also true). These cases are nice and easy to set up—simply assume the antecedent of the conditional (in this case, p) and prove the conclusion q from it. So if your theorem reads, "If p then q," you start your proof with "assume p" and at the end you should have proved q.

Conditionals may be stated in different ways. So instead of "If p then q," a theorem may state that "p only if q," "q if p," or "q, provided p." These all mean the same and require assuming p and proving q from that assumption. Recall that a biconditional ("p if and only if (iff) q") is really two conditionals put together: if p then q, and if q then p. All you have to do, then, is two instances of conditional proof: one for the first conditional and another one for the second. Sometimes, however, it is possible to prove an "iff" statement by chaining together a bunch of other "iff" statements so that you start with "p" an end with "q"—but in that case you have to make sure that each step really is an "iff."

Universal Claims

Using a universal claim is simple: if something is true for anything, it's true for each particular thing. So if, say, the hypothesis of your proof is $A \subseteq B$, that means (unpacking the definition of $\subseteq$), that, for every $x \in A$, $x \in B$. Thus, if you already know that $z \in A$, you can conclude $z \in B$.

Proving a universal claim may seem a little bit tricky. Usually these statements take the following form: "If x has P, then it has Q" or "All Ps are Qs." Of course, it might not fit this form perfectly, and it takes a bit of practice to figure out what you're asked to prove exactly. But: we often have to prove that all objects with some property have a certain other property.

The way to prove a universal claim is to introduce names or variables, for the things that have the one property and then show that they also have the other property. We might put this by saying that to prove something for *all* Ps you have to prove it for an *arbitrary* P. And the name introduced is a name for an arbitrary P. We typically use single letters as these names for arbitrary things, and the letters usually follow conventions: e.g., we use n for natural numbers, φ for formulas, A for sets, f for functions, etc.

The trick is to maintain generality throughout the proof. You start by assuming that an arbitrary object ("x") has the property P, and show (based only on definitions or what you are allowed to assume) that x has the property Q. Because you have not stipulated what x is specifically, other that it has the property P, then you can assert that everything with P has the property Q. In short, x is a stand-in for *all* things with property P.

Proposition 73.4. *For all sets A and B, $A \subseteq A \cup B$.*

Proof. Let A and B be arbitrary sets. We want to show that $A \subseteq A \cup B$. By definition of $\subseteq$, this amounts to: for every x, if $x \in A$ then $x \in A \cup B$. So let $x \in A$ be an arbitrary element of A. We have to show that $x \in A \cup B$. Since $x \in A$, $x \in A$ or $x \in B$. Thus, $x \in \{x : x \in A \vee x \in B\}$. But that, by definition of $\cup$, means $x \in A \cup B$. □

Proof by Cases

Suppose you have a disjunction as an assumption or as an already established conclusion—you have assumed or proved that p or q is true. You want to prove r. You do this in two steps: first you assume that p is true, and prove r, then you assume that q is true and prove r again. This works because we assume or know that one of the two alternatives holds. The two steps establish that either one is sufficient for the truth of r. (If both are true, we have not one but two reasons for why r is true. It is not necessary to separately prove that r is true assuming both p and q.) To indicate what we're doing, we announce that we "distinguish cases." For instance, suppose we know that $x \in B \cup C$. $B \cup C$ is defined as $\{x : x \in B \text{ or } x \in C\}$. In other words, by definition, $x \in B$ or $x \in C$. We would prove that $x \in A$ from this by first assuming that $x \in B$, and proving $x \in A$ from this assumption, and then assume $x \in C$, and again prove $x \in A$ from this. You would write "We distinguish cases" under the assumption, then "Case (1): $x \in B$" underneath, and "Case (2): $x \in C$ halfway down the page. Then you'd proceed to fill in the top half and the bottom half of the page.

Proof by cases is especially useful if what you're proving is itself disjunctive. Here's a simple example:

Proposition 73.5. *Suppose $B \subseteq D$ and $C \subseteq E$. Then $B \cup C \subseteq D \cup E$.*

Proof. Assume (a) that $B \subseteq D$ and (b) $C \subseteq E$. By definition, any $x \in B$ is also $\in D$ (c) and any $x \in C$ is also $\in E$ (d). To show that $B \cup C \subseteq D \cup E$, we

have to show that if $x \in B \cup C$ then $x \in D \cup E$ (by definition of $\subseteq$). $x \in B \cup C$ iff $x \in B$ or $x \in C$ (by definition of $\cup$). Similarly, $x \in D \cup E$ iff $x \in D$ or $x \in E$. So, we have to show: for any x, if $x \in B$ or $x \in C$, then $x \in D$ or $x \in E$.

> So far we've only unpacked definitions! We've reformulated our proposition without $\subseteq$ and $\cup$ and are left with trying to prove a universal conditional claim. By what we've discussed above, this is done by assuming that x is something about which we assume the "if" part is true, and we'll go on to show that the "then" part is true as well. In other words, we'll assume that $x \in B$ or $x \in C$ and show that $x \in D$ or $x \in E$.[2]

Suppose that $x \in B$ or $x \in C$. We have to show that $x \in D$ or $x \in E$. We distinguish cases.

Case 1: $x \in B$. By (c), $x \in D$. Thus, $x \in D$ or $x \in E$. (Here we've made the inference discussed in the preceding subsection!)

Case 2: $x \in C$. By (d), $x \in E$. Thus, $x \in D$ or $x \in E$. □

Proving an Existence Claim

When asked to prove an existence claim, the question will usually be of the form "prove that there is an x such that $\ldots x \ldots$", i.e., that some object that has the property described by "$\ldots x \ldots$". In this case you'll have to identify a suitable object and show that it has the required property. This sounds straightforward, but a proof of this kind can be tricky. Typically it involves *constructing* or *defining* an object and proving that the object so defined has the required property. Finding the right object may be hard, proving that it has the required property may be hard, and sometimes it's even tricky to show that you've succeeded in defining an object at all!

Generally, you'd write this out by specifying the object, e.g., "let x be $\ldots$" (where $\ldots$ specifies which object you have in mind), possibly proving that $\ldots$ in fact describes an object that exists, and then go on to show that x has the property Q. Here's a simple example.

Proposition 73.6. *Suppose that $x \in B$. Then there is an A such that $A \subseteq B$ and $A \neq \emptyset$.*

Proof. Assume $x \in B$. Let $A = \{x\}$.

> Here we've defined the set A by enumerating its elements. Since we assume that x is an object, and we can always form a set by enumerating its elements, we don't have to show that we've succeeded in defining a set A here. However, we still have to show that A has the properties required by the proposition. The proof isn't complete without that!

[2] This paragraph just explains what we're doing—it's not part of the proof, and you don't have to go into all this detail when you write down your own proofs.

Since $x \in A$, $A \neq \emptyset$.

> This relies on the definition of A as $\{x\}$ and the obvious facts that $x \in \{x\}$ and $x \notin \emptyset$.

Since x is the only element of $\{x\}$, and $x \in B$, every element of A is also an element of B. By definition of $\subseteq$, $A \subseteq B$. □

Using Existence Claims

Suppose you know that some existence claim is true (you've proved it, or it's a hypothesis you can use), say, "for some x, $x \in A$" or "there is an $x \in A$." If you want to use it in your proof, you can just pretend that you have a name for one of the things which your hypothesis says exist. Since A contains at least one thing, there are things to which that name might refer. You might of course not be able to pick one out or describe it further (other than that it is $\in A$). But for the purpose of the proof, you can pretend that you have picked it out and give a name to it. It's important to pick a name that you haven't already used (or that appears in your hypotheses), otherwise things can go wrong. In your proof, you indicate this by going from "for some x, $x \in A$" to "Let $a \in A$." Now you can reason about a, use some other hypotheses, etc., until you come to a conclusion, p. If p no longer mentions a, p is independent of the asusmption that $a \in A$, and you've shown that it follows just from the assumption "for some x, $x \in A$."

Proposition 73.7. *If $A \neq \emptyset$, then $A \cup B \neq \emptyset$.*

Proof. Suppose $A \neq \emptyset$. So for some x, $x \in A$.

> Here we first just restated the hypothesis of the proposition. This hypothesis, i.e., $A \neq \emptyset$, hides an existential claim, which you get to only by unpacking a few definitions. The definition of $=$ tells us that $A = \emptyset$ iff every $x \in A$ is also $\in \emptyset$ and every $x \in \emptyset$ is also $\in A$. Negating both sides, we get: $A \neq \emptyset$ iff either some $x \in A$ is $\notin \emptyset$ or some $x \in \emptyset$ is $\notin A$. Since nothing is $\in \emptyset$, the second disjunct can never be true, and "$x \in A$ and $x \notin \emptyset$" reduces to just $x \in A$. So $x \neq \emptyset$ iff for some x, $x \in A$. That's an existence claim. Now we use that existence claim by introducing a name for one of the elements of A:

Let $a \in A$.

> Now we've introduced a name for one of the things $\in A$. We'll continue to argue about a, but we'll be careful to only assume that $a \in A$ and nothing else:

Since $a \in A$, $a \in A \cup B$, by definition of $\cup$. So for some x, $x \in A \cup B$, i.e., $A \cup B \neq \emptyset$.

> In that last step, we went from "$a \in A \cup B$" to "for some x, $x \in A \cup B$." That doesn't mention a anymore, so we know that "for some x, $x \in A \cup B$" follows from "for some x, $x \in A$ alone." But that means that $A \cup B \neq \emptyset$. □

It's maybe good practice to keep bound variables like "x" separate from hypothetical names like a, like we did. In practice, however, we often don't and just use x, like so:

> Suppose $A \neq \emptyset$, i.e., there is an $x \in A$. By definition of $\cup$, $x \in A \cup B$. So $A \cup B \neq \emptyset$.

However, when you do this, you have to be extra careful that you use different x's and y's for different existential claims. For instance, the following is *not* a correct proof of "If $A \neq \emptyset$ and $B \neq \emptyset$ then $A \cap B \neq \emptyset$" (which is not true).

> Suppose $A \neq \emptyset$ and $B \neq \emptyset$. So for some x, $x \in A$ and also for some x, $x \in B$. Since $x \in A$ and $x \in B$, $x \in A \cap B$, by definition of $\cap$. So $A \cap B \neq \emptyset$.

Can you spot where the incorrect step occurs and explain why the result does not hold?

---

content/methods/proofs/example-1.tex

73.5 An Example

mth:prf:ex1:sec

Our first example is the following simple fact about unions and intersections of sets. It will illustrate unpacking definitions, proofs of conjunctions, of universal claims, and proof by cases.

Proposition 73.8. *For any sets A, B, and C, $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$*

Let's prove it!

Proof. We want to show that for any sets A, B, and C, $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

> First we unpack the definition of "=" in the statement of the proposition. Recall that proving sets identical means showing that the sets have the same elements. That is, all elements of $A \cup (B \cap C)$ are also elements of $(A \cup B) \cap (A \cup C)$, and vice versa. The "vice versa" means that also every element of $(A \cup B) \cap (A \cup C)$ must be an element of $A \cup (B \cap C)$. So in unpacking the definition, we see that we have to prove a conjunction. Let's record this:

By definition, $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ iff every element of $A \cup (B \cap C)$ is also an element of $(A \cup B) \cap (A \cup C)$, and every element of $(A \cup B) \cap (A \cup C)$ is an element of $A \cup (B \cap C)$.

> Since this is a conjunction, we must prove each conjunct separately. Lets start with the first: let's prove that every element of $A \cup (B \cap C)$ is also an element of $(A \cup B) \cap (A \cup C)$.
>
> This is a universal claim, and so we consider an arbitrary element of $A \cup (B \cap C)$ and show that it must also be an element of $(A \cup B) \cap (A \cup C)$. We'll pick a variable to call this arbitrary element by, say, z. Our proof continues:

First, we prove that every element of $A \cup (B \cap C)$ is also an element of $(A \cup B) \cap (A \cup C)$. Let $z \in A \cup (B \cap C)$. We have to show that $z \in (A \cup B) \cap (A \cup C)$.

> Now it is time to unpack the definition of $\cup$ and $\cap$. For instance, the definition of $\cup$ is: $A \cup B = \{z : z \in A \text{ or } z \in B\}$. When we apply the definition to "$A \cup (B \cap C)$," the role of the "B" in the definition is now played by "$B \cap C$," so $A \cup (B \cap C) = \{z : z \in A \text{ or } z \in B \cap C\}$. So our assumption that $z \in A \cup (B \cap C)$ amounts to: $z \in \{z : z \in A \text{ or } z \in B \cap C\}$. And $z \in \{z : \ldots z \ldots\}$ iff $\ldots z \ldots$, i.e., in this case, $z \in A$ or $z \in B \cap C$.

By the definition of $\cup$, either $z \in A$ or $z \in B \cap C$.

> Since this is a disjunction, it will be useful to apply proof by cases. We take the two cases, and show that in each one, the conclusion we're aiming for (namely, "$z \in (A \cup B) \cap (A \cup C)$") obtains.

Case 1: Suppose that $z \in A$.

> There's not much more to work from based on our assumptions. So let's look at what we have to work with in the conclusion. We want to show that $z \in (A \cup B) \cap (A \cup C)$. Based on the definition of $\cap$, if we want to show that $z \in (A \cup B) \cap (A \cup C)$, we have to show that it's in both $(A \cup B)$ and $(A \cup C)$. But $z \in A \cup B$ iff $z \in A$ or $z \in B$, and we already have (as the assumption of case 1) that $z \in A$. By the same reasoning—switching C for B—$z \in A \cup C$. This argument went in the reverse direction, so let's record our reasoning in the direction needed in our proof.

Since $z \in A$, $z \in A$ or $z \in B$, and hence, by definition of $\cup$, $z \in A \cup B$. Similarly, $z \in A \cup C$. But this means that $z \in (A \cup B) \cap (A \cup C)$, by definition of $\cap$.

> This completes the first case of the proof by cases. Now we want to derive the conclusion in the second case, where $z \in B \cap C$.

Case 2: Suppose that $z \in B \cap C$.

> Again, we are working with the intersection of two sets. Let's apply the definition of $\cap$:

Since $z \in B \cap C$, z must be an element of both B and C, by definition of $\cap$.

It's time to look at our conclusion again. We have to show that z is in both $(A \cup B)$ and $(A \cup C)$. And again, the solution is immediate.

Since $z \in B$, $z \in (A \cup B)$. Since $z \in C$, also $z \in (A \cup C)$. So, $z \in (A \cup B) \cap (A \cup C)$.

Here we applied the definitions of $\cup$ and $\cap$ again, but since we've already recalled those definitions, and already showed that if z is in one of two sets it is in their union, we don't have to be as explicit in what we've done.

We've completed the second case of the proof by cases, so now we can assert our first conclusion.

So, if $z \in A \cup (B \cap C)$ then $z \in (A \cup B) \cap (A \cup C)$.

Now we just want to show the other direction, that every element of $(A \cup B) \cap (A \cup C)$ is an element of $A \cup (B \cap C)$. As before, we prove this universal claim by assuming we have an arbitrary element of the first set and show it must be in the second set. Let's state what we're about to do.

Now, assume that $z \in (A \cup B) \cap (A \cup C)$. We want to show that $z \in A \cup (B \cap C)$.

We are now working from the hypothesis that $z \in (A \cup B) \cap (A \cup C)$. It hopefully isn't too confusing that we're using the same z here as in the first part of the proof. When we finished that part, all the assumptions we've made there are no longer in effect, so now we can make new assumptions about what z is. If that is confusing to you, just replace z with a different variable in what follows.

We know that z is in both $A \cup B$ and $A \cup C$, by definition of $\cap$. And by the definition of $\cup$, we can further unpack this to: either $z \in A$ or $z \in B$, and also either $z \in A$ or $z \in C$. This looks like a proof by cases again—except the "and" makes it confusing. You might think that this amounts to there being three possibilities: z is either in A, B or C. But that would be a mistake. We have to be careful, so let's consider each disjunction in turn.

By definition of $\cap$, $z \in A \cup B$ and $z \in A \cup C$. By definition of $\cup$, $z \in A$ or $z \in B$. We distinguish cases.

Since we're focusing on the first disjunction, we haven't gotten our second disjunction (from unpacking $A \cup C$) yet. In fact, we don't need it yet. The first case is $z \in A$, and an element of a set is also an element of the union of that set with any other. So case 1 is easy:

Case 1: Suppose that $z \in A$. It follows that $z \in A \cup (B \cap C)$.

Now for the second case, $z \in B$. Here we'll unpack the second $\cup$ and do another proof-by-cases:

Case 2: Suppose that $z \in B$. Since $z \in A \cup C$, either $z \in A$ or $z \in C$. We distinguish cases further:
Case 2a: $z \in A$. Then, again, $z \in A \cup (B \cap C)$.

Ok, this was a bit weird. We didn't actually need the assumption that $z \in B$ for this case, but that's ok.

Case 2b: $z \in C$. Then $z \in B$ and $z \in C$, so $z \in B \cap C$, and consequently, $z \in A \cup (B \cap C)$.

This concludes both proofs-by-cases and so we're done with the second half.

So, if $z \in (A \cup B) \cap (A \cup C)$ then $z \in A \cup (B \cap C)$. □

---

content/methods/proofs/example-2.tex

73.6 Another Example

mth:prf:ex2:sec

Proposition 73.9. *If $A \subseteq C$, then $A \cup (C \setminus A) = C$.*

Proof. Suppose that $A \subseteq C$. We want to show that $A \cup (C \setminus A) = C$.

We begin by observing that this is a conditional statement. It is tacitly universally quantified: the proposition holds for all sets A and C. So A and C are variables for arbitrary sets. To prove such a statement, we assume the antecedent and prove the consequent.

We continue by using the assumption that $A \subseteq C$. Let's unpack the definition of $\subseteq$: the assumption means that all elements of A are also elements of C. Let's write this down—it's an important fact that we'll use throughout the proof.

By the definition of $\subseteq$, since $A \subseteq C$, for all z, if $z \in A$, then $z \in C$.

We've unpacked all the definitions that are given to us in the assumption. Now we can move onto the conclusion. We want to show that $A \cup (C \setminus A) = C$, and so we set up a proof similarly to the last example: we show that every element of $A \cup (C \setminus A)$ is also an element of C and, conversely, every element of C is an element of $A \cup (C \setminus A)$. We can shorten this to: $A \cup (C \setminus A) \subseteq C$ and $C \subseteq A \cup (C \setminus A)$. (Here we're doing the opposite of unpacking a definition, but it makes the proof a bit easier to read.) Since this is a conjunction, we have to prove both parts. To show the first part,

i.e., that every element of $A \cup (C \setminus A)$ is also an element of C, we assume that $z \in A \cup (C \setminus A)$ for an arbitrary z and show that $z \in C$. By the definition of $\cup$, we can conclude that $z \in A$ or $z \in C \setminus A$ from $z \in A \cup (C \setminus A)$. You should now be getting the hang of this.

$A \cup (C \setminus A) = C$ iff $A \cup (C \setminus A) \subseteq C$ and $C \subseteq (A \cup (C \setminus A)$. First we prove that $A \cup (C \setminus A) \subseteq C$. Let $z \in A \cup (C \setminus A)$. So, either $z \in A$ or $z \in (C \setminus A)$.

We've arrived at a disjunction, and from it we want to prove that $z \in C$. We do this using proof by cases.

Case 1: $z \in A$. Since for all z, if $z \in A$, $z \in C$, we have that $z \in C$.

Here we've used the fact recorded earlier which followed from the hypothesis of the proposition that $A \subseteq C$. The first case is complete, and we turn to the second case, $z \in (C \setminus A)$. Recall that $C \setminus A$ denotes the *difference* of the two sets, i.e., the set of all elements of C which are not elements of A. But any element of C not in A is in particular an element of C.

Case 2: $z \in (C \setminus A)$. This means that $z \in C$ and $z \notin A$. So, in particular, $z \in C$.

Great, we've proved the first direction. Now for the second direction. Here we prove that $C \subseteq A \cup (C \setminus A)$. So we assume that $z \in C$ and prove that $z \in A \cup (C \setminus A)$.

Now let $z \in C$. We want to show that $z \in A$ or $z \in C \setminus A$.

Since all elements of A are also elements of C, and $C \setminus A$ is the set of all things that are elements of C but not A, it follows that z is either in A or in $C \setminus A$. This may be a bit unclear if you don't already know why the result is true. It would be better to prove it step-by-step. It will help to use a simple fact which we can state without proof: $z \in A$ or $z \notin A$. This is called the "principle of excluded middle:" for any statement p, either p is true or its negation is true. (Here, p is the statement that $z \in A$.) Since this is a disjunction, we can again use proof-by-cases.

Either $z \in A$ or $z \notin A$. In the former case, $z \in A \cup (C \setminus A)$. In the latter case, $z \in C$ and $z \notin A$, so $z \in C \setminus A$. But then $z \in A \cup (C \setminus A)$.

Our proof is complete: we have shown that $A \cup (C \setminus A) = C$. □

---

content/methods/proofs/proof-by-contradiction.tex

73.7 Proof by Contradiction

mth:prf:con: sec

In the first instance, proof by contradiction is an inference pattern that is used to prove negative claims. Suppose you want to show that some claim p is *false*, i.e., you want to show $\neg p$. The most promising strategy is to (a) suppose that p is true, and (b) show that this assumption leads to something you know to be false. "Something known to be false" may be a result that conflicts with—contradicts—p itself, or some other hypothesis of the overall claim you are considering. For instance, a proof of "if q then $\neg p$" involves assuming that q is true and proving $\neg p$ from it. If you prove $\neg p$ by contradiction, that means assuming p in addition to q. If you can prove $\neg q$ from p, you have shown that the assumption p leads to something that contradicts your other assumption q, since q and $\neg q$ cannot both be true. Of course, you have to use other inference patterns in your proof of the contradiction, as well as unpacking definitions. Let's consider an example.

Proposition 73.10. *If $A \subseteq B$ and $B = \emptyset$, then A has no elements.*

Proof. Suppose $A \subseteq B$ and $B = \emptyset$. We want to show that A has no elements.

> Since this is a conditional claim, we assume the antecedent and want to prove the consequent. The consequent is: A has no elements. We can make that a bit more explicit: it's not the case that there is an $x \in A$.

A has no elements iff it's not the case that there is an x such that $x \in A$.

> So we've determined that what we want to prove is really a negative claim $\neg p$, namely: it's not the case that there is an $x \in A$. To use proof by contradiction, we have to assume the corresponding positive claim p, i.e., there is an $x \in A$, and prove a contradiction from it. We indicate that we're doing a proof by contradiction by writing "by way of contradiction, assume" or even just "suppose not," and then state the assumption p.

Suppose not: there is an $x \in A$.

> This is now the new assumption we'll use to obtain a contradiction. We have two more assumptions: that $A \subseteq B$ and that $B = \emptyset$. The first gives us that $x \in B$:

Since $A \subseteq B$, $x \in B$.

> But since $B = \emptyset$, every element of B (e.g., x) must also be an element of $\emptyset$.

Since $B = \emptyset$, $x \in \emptyset$. This is a contradiction, since by definition $\emptyset$ has no elements.

> This already completes the proof: we've arrived at what we need (a contradiction) from the assumptions we've set up, and this means that the assumptions can't all be true. Since the first two assumptions ($A \subseteq B$ and $B = \emptyset$) are not contested, it must be the last assumption introduced (there is an $x \in A$) that must be false. But if we want to be thorough, we can spell this out.

Thus, our assumption that there is an $x \in A$ must be false, hence, A has no elements by proof by contradiction. □

Every positive claim is trivially equivalent to a negative claim: p iff $\neg\neg p$. So proofs by contradiction can also be used to establish positive claims "indirectly," as follows: To prove p, read it as the negative claim $\neg\neg p$. If we can prove a contradiction from $\neg p$, we've established $\neg\neg p$ by proof by contradiction, and hence p.

In the last example, we aimed to prove a negative claim, namely that A has no elements, and so the assumption we made for the purpose of proof by contradiction (i.e., that there is an $x \in A$) was a positive claim. It gave us something to work with, namely the hypothetical $x \in A$ about which we continued to reason until we got to $x \in \emptyset$.

When proving a positive claim indirectly, the assumption you'd make for the purpose of proof by contradiction would be negative. But very often you can easily reformulate a positive claim as a negative claim, and a negative claim as a positive claim. Our previous proof would have been essentially the same had we proved "$A = \emptyset$" instead of the negative consequent "A has no elements." (By definition of $=$, "$A = \emptyset$" is a general claim, since it unpacks to "every element of A is an element of $\emptyset$ and vice versa".) But it is easily seen to be equivalent to the negative claim "not: there is an $x \in A$."

So it is sometimes easier to work with $\neg p$ as an assumption than it is to prove p directly. Even when a direct proof is just as simple or even simpler (as in the next examples), some people prefer to proceed indirectly. If the double negation confuses you, think of a proof by contradiction of some claim as a proof of a contradiction from the *opposite* claim. So, a proof by contradiction of $\neg p$ is a proof of a contradiction from the assumption p; and proof by contradiction of p is a proof of a contradiction from $\neg p$.

Proposition 73.11. *$A \subseteq A \cup B$.*

Proof. We want to show that $A \subseteq A \cup B$.

> On the face of it, this is a positive claim: every $x \in A$ is also in $A \cup B$. The negation of that is: some $x \in A$ is $\notin A \cup B$. So we can prove the claim indirectly by assuming this negated claim, and showing that it leads to a contradiction.

Suppose not, i.e., $A \not\subseteq A \cup B$.

> We have a definition of $A \subseteq A \cup B$: every $x \in A$ is also $\in A \cup B$. To understand what $A \nsubseteq A \cup B$ means, we have to use some elementary logical manipulation on the unpacked definition: it's false that every $x \in A$ is also $\in A \cup B$ iff there is *some* $x \in A$ that is $\notin C$. (This is a place where you want to be very careful: many students' attempted proofs by contradiction fail because they analyze the negation of a claim like "all As are Bs" incorrectly.) In other words, $A \nsubseteq A \cup B$ iff there is an x such that $x \in A$ and $x \notin A \cup B$. From then on, it's easy.

So, there is an $x \in A$ such that $x \notin A \cup B$. By definition of $\cup$, $x \in A \cup B$ iff $x \in A$ or $x \in B$. Since $x \in A$, we have $x \in A \cup B$. This contradicts the assumption that $x \notin A \cup B$. □

Problem 73.2. Prove *indirectly* that $A \cap B \subseteq A$.

Proposition 73.12. *If $A \subseteq B$ and $B \subseteq C$ then $A \subseteq C$.*

Proof. Suppose $A \subseteq B$ and $B \subseteq C$. We want to show $A \subseteq C$.

> Let's proceed indirectly: we assume the negation of what we want to etablish.

Suppose not, i.e., $A \nsubseteq C$.

> As before, we reason that $A \nsubseteq C$ iff not every $x \in A$ is also $\in C$, i.e., some $x \in A$ is $\notin C$. Don't worry, with practice you won't have to think hard anymore to unpack negations like this.

In other words, there is an x such that $x \in A$ and $x \notin C$.

> Now we can use this to get to our contradiction. Of course, we'll have to use the other two assumptions to do it.

Since $A \subseteq B$, $x \in B$. Since $B \subseteq C$, $x \in C$. But this contradicts $x \notin C$. □

Proposition 73.13. *If $A \cup B = A \cap B$ then $A = B$.*

Proof. Suppose $A \cup B = A \cap B$. We want to show that $A = B$.

> The beginning is now routine:

Assume, by way of contradiction, that $A \neq B$.

> Our assumption for the proof by contradiction is that $A \neq B$. Since $A = B$ iff $A \subseteq B$ an $B \subseteq A$, we get that $A \neq B$ iff $A \nsubseteq B$ *or* $B \nsubseteq A$. (Note how important it is to be careful when manipulating negations!) To prove a contradiction from this disjunction, we use a proof by cases and show that in each case, a contradiction follows.

$A \neq B$ iff $A \not\subseteq B$ or $B \not\subseteq A$. We distinguish cases.

> In the first case, we assume $A \not\subseteq B$, i.e., for some x, $x \in A$ but $\notin B$. $A \cap B$ is defined as those elements that A and B have in common, so if something isn't in one of them, it's not in the intersection. $A \cup B$ is A together with B, so anything in either is also in the union. This tells us that $x \in A \cup B$ but $x \notin A \cap B$, and hence that $A \cap B \neq A \cup B$.

Case 1: $A \not\subseteq B$. Then for some x, $x \in A$ but $x \notin B$. Since $x \notin B$, then $x \notin A \cap B$. Since $x \in A$, $x \in A \cup B$. So, $A \cap B \neq A \cup B$, contradicting the assumption that $A \cap B = A \cup B$.

Case 2: $B \not\subseteq A$. Then for some y, $y \in B$ but $y \notin A$. As before, we have $y \in A \cup B$ but $y \notin A \cap B$, and so $A \cap B \neq A \cup B$, again contradicting $A \cap B = A \cup B$. □

---

content/methods/proofs/reading-proofs.tex

73.8 Reading Proofs

mth:prf:rea:sec

Proofs you find in textbooks and articles very seldom give all the details we have so far included in our examples. Authors often do not draw attention to when they distinguish cases, when they give an indirect proof, or don't mention that they use a definition. So when you read a proof in a textbook, you will often have to fill in those details for yourself in order to understand the proof. Doing this is also good practice to get the hang of the various moves you have to make in a proof. Let's look at an example.

Proposition 73.14 (Absorption). *For all sets A, B,*

$$A \cap (A \cup B) = A$$

Proof. If $z \in A \cap (A \cup B)$, then $z \in A$, so $A \cap (A \cup B) \subseteq A$. Now suppose $z \in A$. Then also $z \in A \cup B$, and therefore also $z \in A \cap (A \cup B)$. □

The preceding proof of the absorption law is very condensed. There is no mention of any definitions used, no "we have to prove that" before we prove it, etc. Let's unpack it. The proposition proved is a general claim about any sets A and B, and when the proof mentions A or B, these are variables for arbitrary sets. The general claims the proof establishes is what's required to prove identity of sets, i.e., that every element of the left side of the identity is an element of the right and vice versa.

> "If $z \in A \cap (A \cup B)$, then $z \in A$, so $A \cap (A \cup B) \subseteq A$."

This is the first half of the proof of the identity: it establishes that if an arbitrary z is an element of the left side, it is also an element of the right, i.e., $A \cap (A \cup B) \subseteq A$. Assume that $z \in A \cap (A \cup B)$. Since z is an element of the intersection of two sets iff it is an element of both sets, we can conclude that $z \in A$ and also $z \in A \cup B$. In particular, $z \in A$, which is what we wanted to show. Since that's all that has to be done for the first half, we know that the rest of the proof must be a proof of the second half, i.e., a proof that $A \subseteq A \cap (A \cup B)$.

> "Now suppose $z \in A$. Then also $z \in A \cup B$, and therefore also $z \in A \cap (A \cup B)$."

We start by assuming that $z \in A$, since we are showing that, for any z, if $z \in A$ then $z \in A \cap (A \cup B)$. To show that $z \in A \cap (A \cup B)$, we have to show (by definition of "$\cap$") that (i) $z \in A$ and also (ii) $z \in A \cup B$. Here (i) is just our assumption, so there is nothing further to prove, and that's why the proof does not mention it again. For (ii), recall that z is an element of a union of sets iff it is an element of at least one of those sets. Since $z \in A$, and $A \cup B$ is the union of A and B, this is the case here. So $z \in A \cup B$. We've shown both (i) $z \in A$ and (ii) $z \in A \cup B$, hence, by definition of "$\cap$," $z \in A \cap (A \cup B)$. The proof doesn't mention those definitions; it's assumed the reader has already internalized them. If you haven't, you'll have to go back and remind yourself what they are. Then you'll also have to recognize why it follows from $z \in A$ that $z \in A \cup B$, and from $z \in A$ and $z \in A \cup B$ that $z \in A \cap (A \cup B)$.

Here's another version of the proof above, with everything made explicit:

Proof. [By definition of $=$ for sets, $A \cap (A \cup B) = A$ we have to show (a) $A \cap (A \cup B) \subseteq A$ and (b) $A \cap (A \cup B) \subseteq A$. (a): By definition of $\subseteq$, we have to show that if $z \in A \cap (A \cup B)$, then $z \in A$.] If $z \in A \cap (A \cup B)$, then $z \in A$ [since by definition of $\cap$, $z \in A \cap (A \cup B)$ iff $z \in A$ and $z \in A \cup B$], so $A \cap (A \cup B) \subseteq A$. [(b): By definition of $\subseteq$, we have to show that if $z \in A$, then $z \in A \cap (A \cup B)$.] Now suppose [(1)] $z \in A$. Then also [(2)] $z \in A \cup B$ [since by (1) $z \in A$ or $z \in B$, which by definition of $\cup$ means $z \in A \cup B$], and therefore also $z \in A \cap (A \cup B)$ [since the definition of $\cap$ requires that $z \in A$, i.e., (1), and $z \in A \cup B$), i.e., (2)]. □

Problem 73.3. Expand the following proof of $A \cup (A \cap B) = A$, where you mention all the inference patterns used, why each step follows from assumptions or claims established before it, and where we have to appeal to which definitions.

Proof. If $z \in A \cup (A \cap B)$ then $z \in A$ or $z \in A \cap B$. If $z \in A \cap B$, $z \in A$. Any $z \in A$ is also $\in A \cup (A \cap B)$. □

73.9 I Can't Do It!

mth:prf:cnt:sec

We all get to a point where we feel like giving up. But you *can* do it. Your instructor and teaching assistant, as well as your fellow students, can help. Ask them for help! Here are a few tips to help you avoid a crisis, and what to do if you feel like giving up.

To make sure you can solve problems successfully, do the following:

1. *Start as far in advance as possible.* We get busy throughout the semester and many of us struggle with procrastination, one of the best things you can do is to start your homework assignments early. That way, if you're stuck, you have time to look for a solution (that isn't crying).

2. *Talk to your classmates.* You are not alone. Others in the class may also struggle—but they may struggle with different things. Talking it out with your peers can give you a different perspective on the problem that might lead to a breakthrough. Of course, don't just copy their solution: ask them for a hint, or explain where you get stuck and ask them for the next step. And when you do get it, reciprocate. Helping someone else along, and explaining things will help you understand better, too.

3. *Ask for help.* You have many resources available to you—your instructor and teaching assistant are there for you and *want* you to succeed. They should be able to help you work out a problem and identify where in the process you're struggling.

4. *Take a break.* If you're stuck, it *might* be because you've been staring at the problem for too long. Take a short break, have a cup of tea, or work on a different problem for a while, then return to the problem with a fresh mind. Sleep on it.

Notice how these strategies require that you've started to work on the proof well in advance? If you've started the proof at 2am the day before it's due, these might not be so helpful.

This might sound like doom and gloom, but solving a proof is a challenge that pays off in the end. Some people do this as a career—so there must be something to enjoy about it. Like basically everything, solving problems and doing proofs is something that requires practice. You might see classmates who find this easy: they've probably just had lots of practice already. Try not to give in too easily.

If you do run out of time (or patience) on a particular problem: that's ok. It doesn't mean you're stupid or that you will never get it. Find out (from your instructor or another student) how it is done, and identify where you went wrong or got stuck, so you can avoid doing that the next time you encounter a similar issue. Then try to do it without looking at the solution. And next time, start (and ask for help) earlier.

---

content/methods/proofs/resources.tex

73.10 Other Resources

There are many books on how to do proofs in mathematics which may be useful. Check out *How to Read and do Proofs: An Introduction to Mathematical Thought Processes* (Solow, 2013) and *How to Prove It: A Structured Approach* (Velleman, 2019) in particular. The *Book of Proof* (Hammack, 2013) and *Mathematical Reasoning* (Sandstrum, 2019) are books on proof that are freely available online. Philosophers might find *More Precisely: The Math you need to do Philosophy* (Steinhart, 2018) to be a good primer on mathematical reasoning.

There are also various shorter guides to proofs available on the internet; e.g., "Introduction to Mathematical Arguments" (Hutchings, 2003) and "How to write proofs" (Cheng, 2004).

Motivational Videos

Feel like you have no motivation to do your homework? Feeling down? These videos might help!

- https://www.youtube.com/watch?v=ZXsQAXx_ao0
- https://www.youtube.com/watch?v=BQ4yd2W50No
- https://www.youtube.com/watch?v=StTqXEQ2l-Y

Chapter 74

Induction

---

content/methods/induction/introduction.tex

74.1 Introduction

mth:ind:int:sec

Induction is an important proof technique which is used, in different forms, in almost all areas of logic, theoretical computer science, and mathematics. It is needed to prove many of the results in logic.

Induction is often contrasted with deduction, and characterized as the inference from the particular to the general. For instance, if we observe many green emeralds, and nothing that we would call an emerald that's not green, we might conclude that all emeralds are green. This is an inductive inference, in that it proceeds from many particular cases (this emerald is green, that emerald is green, etc.) to a general claim (all emeralds are green). *Mathematical* induction is also an inference that concludes a general claim, but it is of a very different kind than this "simple induction."

Very roughly, an inductive proof in mathematics concludes that all mathematical objects of a certain sort have a certain property. In the simplest case, the mathematical objects an inductive proof is concerned with are natural numbers. In that case an inductive proof is used to establish that all natural numbers have some property, and it does this by showing that

1. 0 has the property, and
2. whenever a number k has the property, so does $k + 1$.

Induction on natural numbers can then also often be used to prove general claims about mathematical objects that can be assigned numbers. For instance, finite sets each have a finite number n of elements, and if we can use induction to show that every number n has the property "all finite sets of size n are ..." then we will have shown something about all finite sets.

Induction can also be generalized to mathematical objects that are *inductively defined*. For instance, expressions of a formal language such as those of first-order logic are defined inductively. *Structural induction* is a way to prove results about all such expressions. Structural induction, in particular, is very useful—and widely used—in logic.

---

content/methods/induction/induction-on-N.tex

74.2 Induction on $\mathbb{N}$

mth:ind:inN:sec

In its simplest form, induction is a technique used to prove results for all natural numbers. It uses the fact that by starting from 0 and repeatedly adding 1 we eventually reach every natural number. So to prove that something is true for every number, we can (1) establish that it is true for 0 and (2) show that whenever it is true for a number n, it is also true for the next number $n + 1$. If we abbreviate "number n has property P" by $P(n)$ (and "number k has property P" by $P(k)$, etc.), then a proof by induction that $P(n)$ for all $n \in \mathbb{N}$ consists of:

1. a proof of $P(0)$, and
2. a proof that, for any k, if $P(k)$ then $P(k + 1)$.

To make this crystal clear, suppose we have both (1) and (2). Then (1) tells us that $P(0)$ is true. If we also have (2), we know in particular that if $P(0)$ then

$P(0+1)$, i.e., $P(1)$. This follows from the general statement "for any k, if $P(k)$ then $P(k+1)$" by putting 0 for k. So by modus ponens, we have that $P(1)$. From (2) again, now taking 1 for n, we have: if $P(1)$ then $P(2)$. Since we've just established $P(1)$, by modus ponens, we have $P(2)$. And so on. For any number n, after doing this n times, we eventually arrive at $P(n)$. So (1) and (2) together establish $P(n)$ for any $n \in \mathbb{N}$.

Let's look at an example. Suppose we want to find out how many different sums we can throw with n dice. Although it might seem silly, let's start with 0 dice. If you have no dice there's only one possible sum you can "throw": no dots at all, which sums to 0. So the number of different possible throws is 1. If you have only one die, i.e., $n = 1$, there are six possible values, 1 through 6. With two dice, we can throw any sum from 2 through 12, that's 11 possibilities. With three dice, we can throw any number from 3 to 18, i.e., 16 different possibilities. 1, 6, 11, 16: looks like a pattern: maybe the answer is $5n + 1$? Of course, $5n + 1$ is the maximum possible, because there are only $5n+1$ numbers between n, the lowest value you can throw with n dice (all 1's) and $6n$, the highest you can throw (all 6's).

Theorem 74.1. *With n dice one can throw all $5n + 1$ possible values between n and $6n$.*

Proof. Let $P(n)$ be the claim: "It is possible to throw any number between n and $6n$ using n dice." To use induction, we prove:

1. The *induction basis* $P(1)$, i.e., with just one die, you can throw any number between 1 and 6.

2. The *induction step*, for all k, if $P(k)$ then $P(k + 1)$.

(1) Is proved by inspecting a 6-sided die. It has all 6 sides, and every number between 1 and 6 shows up one on of the sides. So it is possible to throw any number between 1 and 6 using a single die.

To prove (2), we assume the antecedent of the conditional, i.e., $P(k)$. This assumption is called the *inductive hypothesis*. We use it to prove $P(k+1)$. The hard part is to find a way of thinking about the possible values of a throw of $k + 1$ dice in terms of the possible values of throws of k dice plus of throws of the extra $k + 1$-st die—this is what we have to do, though, if we want to use the inductive hypothesis.

The inductive hypothesis says we can get any number between k and $6k$ using k dice. If we throw a 1 with our $(k + 1)$-st die, this adds 1 to the total. So we can throw any value between $k + 1$ and $6k + 1$ by throwing k dice and then rolling a 1 with the $(k+1)$-st die. What's left? The values $6k+2$ through $6k + 6$. We can get these by rolling k 6s and then a number between 2 and 6 with our $(k+1)$-st die. Together, this means that with $k+1$ dice we can throw any of the numbers between $k + 1$ and $6(k + 1)$, i.e., we've proved $P(k + 1)$ using the assumption $P(k)$, the inductive hypothesis. □

Very often we use induction when we want to prove something about a series of objects (numbers, sets, etc.) that is itself defined "inductively," i.e., by defining the $(n+1)$-st object in terms of the n-th. For instance, we can define the sum s_n of the natural numbers up to n by

$$\begin{aligned} s_0 &= 0 \\ s_{n+1} &= s_n + (n+1) \end{aligned}$$

This definition gives:

$$\begin{aligned} s_0 &= 0, \\ s_1 &= s_0 + 1 && = 1, \\ s_2 &= s_1 + 2 && = 1 + 2 = 3 \\ s_3 &= s_2 + 3 && = 1 + 2 + 3 = 6, \text{ etc.} \end{aligned}$$

Now we can prove, by induction, that $s_n = n(n+1)/2$.

Proposition 74.2. $s_n = n(n+1)/2$.

Proof. We have to prove (1) that $s_0 = 0 \cdot (0+1)/2$ and (2) if $s_k = k(k+1)/2$ then $s_{k+1} = (k+1)(k+2)/2$. (1) is obvious. To prove (2), we assume the inductive hypothesis: $s_k = k(k+1)/2$. Using it, we have to show that $s_{k+1} = (k+1)(k+2)/2$.

What is s_{k+1}? By the definition, $s_{k+1} = s_k + (k+1)$. By inductive hypothesis, $s_k = k(k+1)/2$. We can substitute this into the previous equation, and then just need a bit of arithmetic of fractions:

$$\begin{aligned} s_{k+1} &= \frac{k(k+1)}{2} + (k+1) = \\ &= \frac{k(k+1)}{2} + \frac{2(k+1)}{2} = \\ &= \frac{k(k+1) + 2(k+1)}{2} = \\ &= \frac{(k+2)(k+1)}{2}. \end{aligned}$$

□

The important lesson here is that if you're proving something about some inductively defined sequence a_n, induction is the obvious way to go. And even if it isn't (as in the case of the possibilities of dice throws), you can use induction if you can somehow relate the case for $k+1$ to the case for k.

---

content/methods/induction/strong-induction.tex

74.3 Strong Induction

mth:ind:str:sec

In the principle of induction discussed above, we prove $P(0)$ and also if $P(k)$, then $P(k+1)$. In the second part, we assume that $P(k)$ is true and use this assumption to prove $P(k+1)$. Equivalently, of course, we could assume $P(k-1)$ and use it to prove $P(k)$—the important part is that we be able to carry out the inference from any number to its successor; that we can prove the claim in question for any number under the assumption it holds for its predecessor.

There is a variant of the principle of induction in which we don't just assume that the claim holds for the predecessor $k-1$ of k, but for all numbers smaller than k, and use this assumption to establish the claim for k. This also gives us the claim $P(n)$ for all $n \in \mathbb{N}$. For once we have established $P(0)$, we have thereby established that P holds for all numbers less than 1. And if we know that if $P(l)$ for all $l < k$, then $P(k)$, we know this in particular for $k = 1$. So we can conclude $P(1)$. With this we have proved $P(0)$ and $P(1)$, i.e., $P(l)$ for all $l < 2$, and since we have also the conditional, if $P(l)$ for all $l < 2$, then $P(2)$, we can conclude $P(2)$, and so on.

In fact, if we can establish the general conditional "for all k, if $P(l)$ for all $l < k$, then $P(k)$," we do not have to establish $P(0)$ anymore, since it follows from it. For remember that a general claim like "for all $l < k$, $P(l)$" is true if there are no $l < k$. This is a case of vacuous quantification: "all As are Bs" is true if there are no As, $\forall x\,(\varphi(x) \to \psi(x))$ is true if no x satisfies $\varphi(x)$. In this case, the formalized version would be "$\forall l\,(l < k \to P(l))$"—and that is true if there are no $l < k$. And if $k = 0$ that's exactly the case: no $l < 0$, hence "for all $l < 0$, $P(0)$" is true, whatever P is. A proof of "if $P(l)$ for all $l < k$, then $P(k)$" thus automatically establishes $P(0)$.

This variant is useful if establishing the claim for k can't be made to just rely on the claim for $k-1$ but may require the assumption that it is true for one or more $l < k$.

---

content/methods/induction/inductive-definitions.tex

74.4 Inductive Definitions

mth:ind:idf:sec

In logic we very often define kinds of objects *inductively*, i.e., by specifying rules for what counts as an object of the kind to be defined which explain how to get new objects of that kind from old objects of that kind. For instance, we often define special kinds of sequences of symbols, such as the terms and formulas of a language, by induction. For a simple example, consider strings consisting of letters a, b, c, d, the symbol ∘, and brackets [and], such as "[[c ∘ d][", "[a[]∘]", "a" or "[[a ∘ b] ∘ d]". You probably feel that there's something "wrong" with the first two strings: the brackets don't "balance" at all in the first, and you might feel that the "∘" should "connect" expressions that themselves make sense. The third and fourth string look better: for every "[" there's a closing "]" (if there are any at all), and for any ∘ we can find "nice" expressions on either side, surrounded by a pair of parentheses.

We would like to precisely specify what counts as a "nice term." First of all, every letter by itself is nice. Anything that's not just a letter by itself should be of the form "$[t \circ s]$" where s and t are themselves nice. Conversely, if t and s are nice, then we can form a new nice term by putting a $\circ$ between them and surround them by a pair of brackets. We might use these operations to *define* the set of nice terms. This is an *inductive definition*.

Definition 74.3 (Nice terms). The set of *nice terms* is inductively defined as follows:

1. Any letter a, b, c, d is a nice term.
2. If s_1 and s_2 are nice terms, then so is $[s_1 \circ s_2]$.
3. Nothing else is a nice term.

This definition tells us that something counts as a nice term iff it can be constructed according to the two conditions (1) and (2) in some finite number of steps. In the first step, we construct all nice terms just consisting of letters by themselves, i.e.,

$$\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}$$

In the second step, we apply (2) to the terms we've constructed. We'll get

$$[\mathrm{a} \circ \mathrm{a}], [\mathrm{a} \circ \mathrm{b}], [\mathrm{b} \circ \mathrm{a}], \ldots, [\mathrm{d} \circ \mathrm{d}]$$

for all combinations of two letters. In the third step, we apply (2) again, to any two nice terms we've constructed so far. We get new nice term such as $[\mathrm{a} \circ [\mathrm{a} \circ \mathrm{a}]]$—where t is a from step 1 and s is $[\mathrm{a} \circ \mathrm{a}]$ from step 2—and $[[\mathrm{b} \circ \mathrm{c}] \circ [\mathrm{d} \circ \mathrm{b}]]$ constructed out of the two terms $[\mathrm{b} \circ \mathrm{c}]$ and $[\mathrm{d} \circ \mathrm{b}]$ from step 2. And so on. Clause (3) rules out that anything not constructed in this way sneaks into the set of nice terms.

Note that we have not yet proved that every sequence of symbols that "feels" nice is nice according to this definition. However, it should be clear that everything we can construct does in fact "feel nice": brackets are balanced, and $\circ$ connects parts that are themselves nice.

The key feature of inductive definitions is that if you want to prove something about all nice terms, the definition tells you which cases you must consider. For instance, if you are told that t is a nice term, the inductive definition tells you what t can look like: t can be a letter, or it can be $[s_1 \circ s_2]$ for some pair of nice terms s_1 and s_2. Because of clause (3), those are the only possibilities.

When proving claims about all of an inductively defined set, the strong form of induction becomes particularly important. For instance, suppose we want to prove that for every nice term of length n, the number of [in it is $< n/2$. This can be seen as a claim about all n: for every n, the number of [in any nice term of length n is $< n/2$.

Proposition 74.4. *For any n, the number of [in a nice term of length n is $< n/2$.*

Proof. To prove this result by (strong) induction, we have to show that the following conditional claim is true:

> If for every $l < k$, any nice term of length l has $< l/2$ ['s, then any nice term of length k has $< k/2$ ['s.

To show this conditional, assume that its antecedent is true, i.e., assume that for any $l < k$, nice terms of length l contain $< l/2$ ['s. We call this assumption the inductive hypothesis. We want to show the same is true for nice terms of length k.

So suppose t is a nice term of length k. Because nice terms are inductively defined, we have two cases: (1) t is a letter by itself, or (2) t is $[s_1 \circ s_2]$ for some nice terms s_1 and s_2.

1. t is a letter. Then $k = 1$, and the number of [in t is 0. Since $0 < 1/2$, the claim holds.

2. t is $[s_1 \circ s_2]$ for some nice terms s_1 and s_2. Let's let l_1 be the length of s_1 and l_2 be the length of s_2. Then the length k of t is $l_1 + l_2 + 3$ (the lengths of s_1 and s_2 plus three symbols [, $\circ$,]). Since $l_1 + l_2 + 3$ is always greater than l_1, $l_1 < k$. Similarly, $l_2 < k$. That means that the induction hypothesis applies to the terms s_1 and s_2: the number m_1 of [in s_1 is $< l_1/2$, and the number m_2 of [in s_2 is $< l_2/2$.

   The number of [in t is the number of [in s_1, plus the number of [in s_2, plus 1, i.e., it is $m_1 + m_2 + 1$. Since $m_1 < l_1/2$ and $m_2 < l_2/2$ we have:

$$m_1 + m_2 + 1 < \frac{l_1}{2} + \frac{l_2}{2} + 1 = \frac{l_1 + l_2 + 2}{2} < \frac{l_1 + l_2 + 3}{2} = k/2.$$

In each case, we've shown that the number of [in t is $< k/2$ (on the basis of the inductive hypothesis). By strong induction, the proposition follows. □

Problem 74.1. Define the set of supernice terms by

1. Any letter a, b, c, d is a supernice term.

2. If s is a supernice term, then so is $[s]$.

3. If s_1 and s_2 are supernice terms, then so is $[s_1 \circ s_2]$.

4. Nothing else is a supernice term.

Show that the number of [in a supernice term t of length n is $\leq n/2 + 1$.

74.5 Structural Induction

mth:ind:sti: sec

So far we have used induction to establish results about all natural numbers. But a corresponding principle can be used directly to prove results about all elements of an inductively defined set. This often called *structural* induction, because it depends on the structure of the inductively defined objects.

Generally, an inductive definition is given by (a) a list of "initial" elements of the set and (b) a list of operations which produce new elements of the set from old ones. In the case of nice terms, for instance, the initial objects are the letters. We only have one operation: the operations are

$$o(s_1, s_2) = [s_1 \circ s_2]$$

You can even think of the natural numbers $\mathbb{N}$ themselves as being given by an inductive definition: the initial object is 0, and the operation is the successor function $x + 1$.

In order to prove something about all elements of an inductively defined set, i.e., that every element of the set has a property P, we must:

1. Prove that the initial objects have P
2. Prove that for each operation o, if the arguments have P, so does the result.

For instance, in order to prove something about all nice terms, we would prove that it is true about all letters, and that it is true about $[s_1 \circ s_2]$ provided it is true of s_1 and s_2 individually.

Proposition 74.5. *The number of* [*equals the number of*] *in any nice term* t*.*

Proof. We use structural induction. Nice terms are inductively defined, with letters as initial objects and the operation o for constructing new nice terms out of old ones.

1. The claim is true for every letter, since the number of [in a letter by itself is 0 and the number of] in it is also 0.
2. Suppose the number of [in s_1 equals the number of], and the same is true for s_2. The number of [in $o(s_1, s_2)$, i.e., in $[s_1 \circ s_2]$, is the sum of the number of [in s_1 and s_2 plus one. The number of] in $o(s_1, s_2)$ is the sum of the number of] in s_1 and s_2 plus one. Thus, the number of [in $o(s_1, s_2)$ equals the number of] in $o(s_1, s_2)$. □

Problem 74.2. Prove by structural induction that no nice term starts with].

Let's give another proof by structural induction: a proper initial segment of a string t of symbols is any string s that agrees with t symbol by symbol, read from the left, but t is longer. So, e.g., $[a \circ$ is a proper initial segment of $[a \circ b]$, but neither are $[b \circ$ (they disagree at the second symbol) nor $[a \circ b]$ (they are the same length).

mth:ind:sti: prop:initial **Proposition 74.6.** *Every proper initial segment of a nice term t has more ['s than] 's.*

Proof. By induction on t:

1. t is a letter by itself: Then t has no proper initial segments.
2. $t = [s_1 \circ s_2]$ for some nice terms s_1 and s_2. If r is a proper initial segment of t, there are a number of possibilities:
   a) r is just [: Then r has one more [than it does].
   b) r is $[r_1$ where r_1 is a proper initial segment of s_1: Since s_1 is a nice term, by induction hypothesis, r_1 has more [than] and the same is true for $[r_1$.
   c) r is $[s_1$ or $[s_1 \circ$: By the previous result, the number of [and] in s_1 are equal; so the number of [in $[s_1$ or $[s_1 \circ$ is one more than the number of].
   d) r is $[s_1 \circ r_2$ where r_2 is a proper initial segment of s_2: By induction hypothesis, r_2 contains more [than]. By the previous result, the number of [and of] in s_1 are equal. So the number of [in $[s_1 \circ r_2$ is greater than the number of].
   e) r is $[s_1 \circ s_2$: By the previous result, the number of [and] in s_1 are equal, and the same for s_2. So there is one more [in $[s_1 \circ s_2$ than there are]. □

---

content/methods/induction/relations.tex

74.6 Relations and Functions

mth:ind:rel: sec When we have defined a set of objects (such as the natural numbers or the nice terms) inductively, we can also define *relations on* these objects by induction. For instance, consider the following idea: a nice term t_1 is a subterm of a nice term t_2 if it occurs as a part of it. Let's use a symbol for it: $t_1 \sqsubseteq t_2$. Every nice term is a subterm of itself, of course: $t \sqsubseteq t$. We can give an inductive definition of this relation as follows:

Definition 74.7. The relation of a nice term t_1 being a subterm of t_2, $t_1 \sqsubseteq t_2$, is defined by induction on t_2 as follows:

1. If t_2 is a letter, then $t_1 \sqsubseteq t_2$ iff $t_1 = t_2$.
2. If t_2 is $[s_1 \circ s_2]$, then $t_1 \sqsubseteq t_2$ iff $t_1 = t_2$, $t_1 \sqsubseteq s_1$, or $t_1 \sqsubseteq s_2$.

This definition, for instance, will tell us that a $\sqsubseteq$ [b $\circ$ a]. For (2) says that a $\sqsubseteq$ [b $\circ$ a] iff a = [b $\circ$ a], or a $\sqsubseteq$ *b*, or a $\sqsubseteq$ a. The first two are false: a clearly

isn't identical to [b∘a], and by (1), a $\sqsubseteq$ b iff a = b, which is also false. However, also by (1), a $\sqsubseteq$ a iff a = a, which is true.

It's important to note that the success of this definition depends on a fact that we haven't proved yet: every nice term t is either a letter by itself, or there are *uniquely determined* nice terms s_1 and s_2 such that $t = [s_1 \circ s_2]$. "Uniquely determined" here means that if $t = [s_1 \circ s_2]$ it isn't *also* $= [r_1 \circ r_2]$ with $s_1 \neq r_1$ or $s_2 \neq r_2$. If this were the case, then clause (2) may come in conflict with itself: reading t_2 as $[s_1 \circ s_2]$ we might get $t_1 \sqsubseteq t_2$, but if we read t_2 as $[r_1 \circ r_2]$ we might get not $t_1 \sqsubseteq t_2$. Before we prove that this can't happen, let's look at an example where it *can* happen.

Definition 74.8. Define *bracketless terms* inductively by

1. Every letter is a bracketless term.
2. If s_1 and s_2 are bracketless terms, then $s_1 \circ s_2$ is a bracketless term.
3. Nothing else is a bracketless term.

Bracketless terms are, e.g., a, b ∘ d, b ∘ a ∘ b. Now if we defined "subterm" for bracketless terms the way we did above, the second clause would read

$$\text{If } t_2 = s_1 \circ s_2, \text{ then } t_1 \sqsubseteq t_2 \text{ iff } t_1 = t_2,\ t_1 \sqsubseteq s_1, \text{ or } t_1 \sqsubseteq s_2.$$

Now b ∘ a ∘ b is of the form $s_1 \circ s_2$ with

$$s_1 = \text{b and} \qquad\qquad s_2 = \text{a} \circ \text{b}.$$

It is also of the form $r_1 \circ r_2$ with

$$r_1 = \text{b} \circ \text{a and} \qquad\qquad r_2 = \text{b}.$$

Now is a ∘ b a subterm of b ∘ a ∘ b? The answer is yes if we go by the first reading, and no if we go by the second.

The property that the way a nice term is built up from other nice terms is unique is called *unique readability*. Since inductive definitions of relations for such inductively defined objects are important, we have to prove that it holds.

Proposition 74.9. *Suppose t is a nice term. Then either t is a letter by itself, or there are uniquely determined nice terms s_1, s_2 such that $t = [s_1 \circ s_2]$.*

Proof. If t is a letter by itself, the condition is satisfied. So assume t isn't a letter by itself. We can tell from the inductive definition that then t must be of the form $[s_1 \circ s_2]$ for some nice terms s_1 and s_2. It remains to show that these are uniquely determined, i.e., if $t = [r_1 \circ r_2]$, then $s_1 = r_1$ and $s_2 = r_2$.

So suppose $t = [s_1 \circ s_2]$ and also $t = [r_1 \circ r_2]$ for nice terms s_1, s_2, r_1, r_2. We have to show that $s_1 = r_1$ and $s_2 = r_2$. First, s_1 and r_1 must be identical, for otherwise one is a proper initial segment of the other. But by Proposition 74.6, that is impossible if s_1 and r_1 are both nice terms. But if $s_1 = r_1$, then clearly also $s_2 = r_2$. □

We can also define functions inductively: e.g., we can define the function f that maps any nice term to the maximum depth of nested $[\dots]$ in it as follows:

mth:ind:rel: defn:depth **Definition 74.10.** The *depth* of a nice term, $f(t)$, is defined inductively as follows:

$$f(t) = \begin{cases} 0 & \text{if } t \text{ is a letter} \\ \max(f(s_1), f(s_2)) + 1 & \text{if } t = [s_1 \circ s_2]. \end{cases}$$

For instance

$$\begin{aligned} f([\mathrm{a} \circ \mathrm{b}]) &= \max(f(\mathrm{a}), f(\mathrm{b})) + 1 = \\ &= \max(0, 0) + 1 = 1, \text{ and} \\ f([[\mathrm{a} \circ \mathrm{b}] \circ \mathrm{c}]) &= \max(f([\mathrm{a} \circ \mathrm{b}]), f(\mathrm{c})) + 1 = \\ &= \max(1, 0) + 1 = 2. \end{aligned}$$

Here, of course, we assume that s_1 an s_2 are nice terms, and make use of the fact that every nice term is either a letter or of the form $[s_1 \circ s_2]$. It is again important that it can be of this form in only one way. To see why, consider again the bracketless terms we defined earlier. The corresponding "definition" would be:

$$g(t) = \begin{cases} 0 & \text{if } t \text{ is a letter} \\ \max(g(s_1), g(s_2)) + 1 & \text{if } t = s_1 \circ s_2. \end{cases}$$

Now consider the bracketless term $\mathrm{a} \circ \mathrm{b} \circ \mathrm{c} \circ \mathrm{d}$. It can be read in more than one way, e.g., as $s_1 \circ s_2$ with

$$s_1 = \mathrm{a} \text{ and} \qquad s_2 = \mathrm{b} \circ \mathrm{c} \circ \mathrm{d},$$

or as $r_1 \circ r_2$ with

$$r_1 = \mathrm{a} \circ b \text{ and} \qquad r_2 = \mathrm{c} \circ \mathrm{d}.$$

Calculating g according to the first way of reading it would give

$$\begin{aligned} g(s_1 \circ s_2) &= \max(g(\mathrm{a}), g(\mathrm{b} \circ \mathrm{c} \circ \mathrm{d})) + 1 = \\ &= \max(0, 2) + 1 = 3 \end{aligned}$$

while according to the other reading we get

$$\begin{aligned} g(r_1 \circ r_2) &= \max(g(\mathrm{a} \circ \mathrm{b}), g(\mathrm{c} \circ \mathrm{d})) + 1 = \\ &= \max(1, 1) + 1 = 2 \end{aligned}$$

But a function must always yield a unique value; so our "definition" of g doesn't define a function at all.

Problem 74.3. Give an inductive definition of the function l, where $l(t)$ is the number of symbols in the nice term t.

Problem 74.4. Prove by structural induction on nice terms t that $f(t) < l(t)$ (where $l(t)$ is the number of symbols in t and $f(t)$ is the depth of t as defined in Definition 74.10).

Part XVII

History

Chapter 75

Biographies

---

content/history/biographies/georg-cantor.tex

75.1 Georg Cantor

his:bio:can:sec

An early biography of Georg Cantor (GAY-org KAHN-tor) claimed that he was born and found on a ship that was sailing for Saint Petersburg, Russia, and that his parents were unknown. This, however, is not true; although he was born in Saint Petersburg in 1845.

Figure 75.1: Georg Cantor

Cantor received his doctorate in mathematics at the University of Berlin in 1867. He is known for his work in set theory, and is credited with founding set theory as a distinctive research discipline. He was the first to prove that there are infinite sets of different sizes. His theories, and especially his theory of infinities, caused much debate among mathematicians at the time, and his work was controversial.

Cantor's religious beliefs and his mathematical work were inextricably tied; he even claimed that the theory of transfinite numbers had been communicated to him directly by God. In later

life, Cantor suffered from mental illness. Beginning in 1894, and more frequently towards his later years, Cantor was hospitalized. The heavy criticism of his work, including a falling out with the mathematician Leopold Kronecker, led to depression and a lack of interest in mathematics. During depressive episodes, Cantor would turn to philosophy and literature, and even published a theory that Francis Bacon was the author of Shakespeare's plays.

Cantor died on January 6, 1918, in a sanatorium in Halle.

Further Reading For full biographies of Cantor, see Dauben (1990) and Grattan-Guinness (1971). Cantor's radical views are also described in the BBC Radio 4 program *A Brief History of Mathematics* (du Sautoy, 2014). If you'd like to hear about Cantor's theories in rap form, see Rose (2012).

---

content/history/biographies/alonzo-church.tex

75.2 Alonzo Church

his:bio:chu:sec

Alonzo Church was born in Washington, DC on June 14, 1903. In early childhood, an air gun incident left Church blind in one eye. He finished preparatory school in Connecticut in 1920 and began his university education at Princeton that same year. He completed his doctoral studies in 1927. After a couple years abroad, Church returned to Princeton. Church was known to be exceedingly polite and careful. His blackboard writing was immaculate, and he would preserve important papers by carefully covering them in Duco cement (a clear glue). Outside of his academic pursuits, he enjoyed reading science fiction magazines and was not afraid to write to the editors if he spotted any inaccuracies in the writing.

Figure 75.2: Alonzo Church

Church's academic achievements were great. Together with his students Stephen Kleene and Barkley Rosser, he developed a theory of effective calculability, the lambda calculus, independently of Alan Turing's development of the Turing machine. The two definitions of computability are equivalent, and give rise to what is now known as the *Church–Turing Thesis*, that a function of the natural numbers is effectively computable if and only if it is computable via Turing machine (or lambda calculus). He also proved what is now known as *Church's Theorem*: The decision problem for the validity of first-order formulas is unsolvable.

Church continued his work into old age. In 1967 he left Princeton for UCLA, where he was professor until his retirement in 1990. Church passed away on August 1, 1995 at the age of 92.

Further Reading For a brief biography of Church, see Enderton (2019). Church's original writings on the lambda calculus and the Entscheidungsproblem (Church's Thesis) are Church (1936a,b). Aspray (1984) records an interview with Church about the Princeton mathematics community in the 1930s. Church wrote a series of book reviews of the *Journal of Symbolic Logic* from 1936 until 1979. They are all archived on John MacFarlane's website (MacFarlane, 2015).

---

content/history/biographies/gerhard-gentzen.tex

75.3 Gerhard Gentzen

his:bio:gen:sec

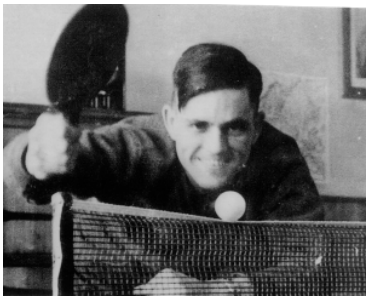


Figure 75.3: Gerhard Gentzen

Gerhard Gentzen is known primarily as the creator of structural proof theory, and specifically the creation of the natural deduction and sequent calculus derivation systems. He was born on November 24, 1909 in Greifswald, Germany. Gerhard was homeschooled for three years before attending preparatory school, where he was behind most of his classmates in terms of education. Despite this, he was a brilliant student and showed a strong aptitude for mathematics. His interests were varied, and he, for instance, also write poems for his mother and plays for the school theatre.

Gentzen began his university studies at the University of Greifswald, but moved around to Göttingen, Munich, and Berlin. He received his doctorate in 1933 from the University of Göttingen under Hermann Weyl. (Paul Bernays supervised most of his work, but was dismissed from the university by the Nazis.) In 1934, Gentzen began work as an assistant to David Hilbert. That same year he developed the sequent calculus and natural deduction derivation systems, in his papers *Untersuchungen über das logische Schließen I–II [Investigations Into Logical Deduction I–II]*. He proved the consistency of the Peano axioms in 1936.

Gentzen's relationship with the Nazis is complicated. At the same time his mentor Bernays was forced to leave Germany, Gentzen joined the university branch of the SA, the Nazi paramilitary organization. Like many Germans, he was a member of the Nazi party. During the war, he served as a telecommunications officer for the air intelligence unit. However, in 1942 he was released from duty due to a nervous breakdown. It is unclear whether or not Gentzen's

loyalties lay with the Nazi party, or whether he joined the party in order to ensure academic success.

In 1943, Gentzen was offered an academic position at the Mathematical Institute of the German University of Prague, which he accepted. However, in 1945 the citizens of Prague revolted against German occupation. Soviet forces arrived in the city and arrested all the professors at the university. Because of his membership in Nazi organizations, Gentzen was taken to a forced labour camp. He died of malnutrition while in his cell on August 4, 1945 at the age of 35.

Further Reading For a full biography of Gentzen, see Menzler-Trott (2007). An interesting read about mathematicians under Nazi rule, which gives a brief note about Gentzen's life, is given by Segal (2014). Gentzen's papers on logical deduction are available in the original german (Gentzen, 1935a,b). English translations of Gentzen's papers have been collected in a single volume by Szabo (1969), which also includes a biographical sketch.

---

content/history/biographies/kurt-goedel.tex

75.4 Kurt Gödel

his:bio:god:sec

Kurt Gödel (GER-dle) was born on April 28, 1906 in Brünn in the Austro-Hungarian empire (now Brno in the Czech Republic). Due to his inquisitive and bright nature, young Kurtele was often called "Der kleine Herr Warum" (Little Mr. Why) by his family. He excelled in academics from primary school onward, where he got less than the highest grade only in mathematics. Gödel was often absent from school due to poor health and was exempt from physical education. He was diagnosed with rheumatic fever during his childhood. Throughout his life, he believed this permanently affected his heart despite medical assessment saying otherwise.

Figure 75.4: Kurt Gödel

Gödel began studying at the University of Vienna in 1924 and completed his doctoral studies in 1929. He first intended to study physics, but his interests soon moved to mathematics and especially logic, in part due to the influence of the philosopher Rudolf Carnap. His dissertation, written under the supervision of Hans Hahn, proved the completeness theorem of first-order predicate logic with identity (Gödel,

1929). Only a year later, he obtained his most famous results—the first and second incompleteness theorems (published in Gödel 1931). During his time in Vienna, Gödel was heavily involved with the Vienna Circle, a group of scientifically-minded philosophers that included Carnap, whose work was especially influenced by Gödel's results.

In 1938, Gödel married Adele Nimbursky. His parents were not pleased: not only was she six years older than him and already divorced, but she worked as a dancer in a nightclub. Social pressures did not affect Gödel, however, and they remained happily married until his death.

After Nazi Germany annexed Austria in 1938, Gödel and Adele emigrated to the United States, where he took up a position at the Institute for Advanced Study in Princeton, New Jersey. Despite his introversion and eccentric nature, Gödel's time at Princeton was collaborative and fruitful. He published essays in set theory, philosophy and physics. Notably, he struck up a particularly strong friendship with his colleague at the IAS, Albert Einstein.

In his later years, Gödel's mental health deteriorated. His wife's hospitalization in 1977 meant she was no longer able to cook his meals for him. Having suffered from mental health issues throughout his life, he succumbed to paranoia. Deathly afraid of being poisoned, Gödel refused to eat. He died of starvation on January 14, 1978, in Princeton.

Further Reading For a complete biography of Gödel's life is available, see John Dawson (1997). For further biographical pieces, as well as essays about Gödel's contributions to logic and philosophy, see Wang (1990), Baaz et al. (2011), Takeuti et al. (2003), and Sigmund et al. (2007).

Gödel's PhD thesis is available in the original German (Gödel, 1929). The original text of the incompleteness theorems is (Gödel, 1931). All of Gödel's published and unpublished writings, as well as a selection of correspondence, are available in English in his *Collected Papers* Feferman et al. (1986, 1990).

For a detailed treatment of Gödel's incompleteness theorems, see Smith (2013). For an informal, philosophical discussion of Gödel's theorems, see Mark Linsenmayer's podcast (Linsenmayer, 2014).

---

content/history/biographies/emmy-noether.tex

75.5 Emmy Noether

his:bio:noe:sec

Emmy Noether (NER-ter) was born in Erlangen, Germany, on March 23, 1882, to an upper-middle class scholarly family. Hailed as the "mother of modern algebra," Noether made groundbreaking contributions to both mathematics and physics, despite significant barriers to women's education. In Germany at the time, young girls were meant to be educated in arts and were not allowed to attend college preparatory schools. However, after auditing classes at the Universities of Göttingen and Erlangen (where her father was professor of mathematics), Noether was eventually able to enroll as a student at Erlangen

in 1904, when their policy was updated to allow female students. She received her doctorate in mathematics in 1907.

Figure 75.5: Emmy Noether

Despite her qualifications, Noether experienced much resistance during her career. From 1908–1915, she taught at Erlangen without pay. During this time, she caught the attention of David Hilbert, one of the world's foremost mathematicians of the time, who invited her to Göttingen. However, women were prohibited from obtaining professorships, and she was only able to lecture under Hilbert's name, again without pay. During this time she proved what is now known as Noether's theorem, which is still used in theoretical physics today. Noether was finally granted the right to teach in 1919. Hilbert's response to continued resistance of his university colleagues reportedly was: "Gentlemen, the faculty senate is not a bathhouse."

In the later 1920s, she concentrated on work in abstract algebra, and her contributions revolutionized the field. In her proofs she often made use of the so-called ascending chain condition, which states that there is no infinite strictly increasing chain of certain sets. For instance, certain algebraic structures now known as Noetherian rings have the property that there are no infinite sequences of ideals $I_1 \subsetneq I_2 \subsetneq \dots$. The condition can be generalized to any partial order (in algebra, it concerns the special case of ideals ordered by the subset relation), and we can also consider the dual descending chain condition, where every strictly *de*creasing sequence in a partial order eventually ends. If a partial order satisfies the descending chain condition, it is possible to use induction along this order in a similar way in which we can use induction along the $<$ order on $\mathbb{N}$. Such orders are called *well-founded* or *Noetherian*, and the corresponding proof principle *Noetherian induction*.

Noether was Jewish, and when the Nazis came to power in 1933, she was dismissed from her position. Luckily, Noether was able to emigrate to the United States for a temporary position at Bryn Mawr, Pennsylvania. During her time there she also lectured at Princeton, although she found the university to be unwelcoming to women (Dick, 1981, 81). In 1935, Noether underwent an operation to remove a uterine tumour. She died from an infection as a result of the surgery, and was buried at Bryn Mawr.

Further Reading For a biography of Noether, see Dick (1981). The Perimeter Institute for Theoretical Physics has their lectures on Noether's life and

influence available online (Institute, 2015). If you're tired of reading, *Stuff You Missed in History Class* has a podcast on Noether's life and influence (Frey and Wilson, 2015). The collected works of Noether are available in the original German (Jacobson, 1983).

---

content/history/biographies/rozsa-peter.tex

75.6 Rózsa Péter

his:bio:pet:sec

Figure 75.6: Rózsa Péter

Rózsa Péter was born Rósza Politzer, in Budapest, Hungary, on February 17, 1905. She is best known for her work on recursive functions, which was essential for the creation of the field of recursion theory.

Péter was raised during harsh political times—WWI raged when she was a teenager—but was able to attend the affluent Maria Terezia Girls' School in Budapest, from where she graduated in 1922. She then studied at Pázmány Péter University (later renamed Loránd Eötvös University) in Budapest. She began studying chemistry at the insistence of her father, but later switched to mathematics, and graduated in 1927. Although she had the credentials to teach high school mathematics, the economic situation at the time was dire as the Great Depression affected the world economy. During this time, Péter took odd jobs as a tutor and private teacher of mathematics. She eventually returned to university to take up graduate studies in mathematics. She had originally planned to work in number theory, but after finding out that her results had already been proven, she almost gave up on mathematics altogether. She was encouraged to work on Gödel's incompleteness theorems, and unknowingly proved several of his results in different ways. This restored her confidence, and Péter went on to write her first papers on recursion theory, inspired by David Hilbert's foundational program. She received her PhD in 1935, and in 1937 she became an editor for the *Journal of Symbolic Logic*.

Péter's early papers are widely credited as founding contributions to the field of recursive function theory. In Péter (1935a), she investigated the relationship between different kinds of recursion. In Péter (1935b), she showed that a certain recursively defined function is not primitive recursive. This simplified an earlier result due to Wilhelm Ackermann. Péter's simplified function is what's now often called the Ackermann function—and sometimes, more

properly, the Ackermann–Péter function. She wrote the first book on recursive function theory (Péter, 1951).

Despite the importance and influence of her work, Péter did not obtain a full-time teaching position until 1945. During the Nazi occupation of Hungary during World War II, Péter was not allowed to teach due to anti-Semitic laws. In 1944 the government created a Jewish ghetto in Budapest; the ghetto was cut off from the rest of the city and attended by armed guards. Péter was forced to live in the ghetto until 1945 when it was liberated. She then went on to teach at the Budapest Teachers Training College, and from 1955 onward at Eötvös Loránd University. She was the first female Hungarian mathematician to become an Academic Doctor of Mathematics, and the first woman to be elected to the Hungarian Academy of Sciences.

Péter was known as a passionate teacher of mathematics, who preferred to explore the nature and beauty of mathematical problems with her students rather than to merely lecture. As a result, she was affectionately called "Aunt Rosa" by her students. Péter died in 1977 at the age of 71.

Further Reading For more biographical reading, see (O'Connor and Robertson, 2014) and (Andrásfai, 1986). Tamassy (1994) conducted a brief interview with Péter. For a fun read about mathematics, see Péter's book *Playing With Infinity* (Péter, 2010).

---

content/history/biographies/julia-robinson.tex

75.7 Julia Robinson

his:bio:rob:sec

Julia Bowman Robinson was an American mathematician. She is known mainly for her work on decision problems, and most famously for her contributions to the solution of Hilbert's tenth problem. Robinson was born in St. Louis, Missouri, on December 8, 1919. Robinson recalls being intrigued by numbers already as a child (Reid, 1986, 4). At age nine she contracted scarlet fever and suffered from several recurrent bouts of rheumatic fever. This forced her to spend much of her time in bed, putting her behind in her education. Although she was able to catch up with the help of private tutors, the physical effects of her illness had a lasting impact on her life.

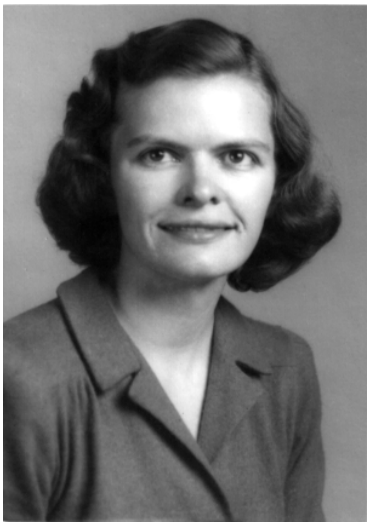


Figure 75.7: Julia Robinson

Despite her childhood struggles, Robinson graduated high school with several

awards in mathematics and the sciences. She started her university career at San Diego State College, and transferred to the University of California, Berkeley, as a senior. There she was influenced by the mathematician Raphael Robinson. They became good friends, and married in 1941. As a spouse of a faculty member, Robinson was barred from teaching in the mathematics department at Berkeley. Although she continued to audit mathematics classes, she hoped to leave university and start a family. Not long after her wedding, however, Robinson contracted pneumonia. She was told that there was substantial scar tissue build up on her heart due to the rheumatic fever she suffered as a child. Due to the severity of the scar tissue, the doctor predicted that she would not live past forty and she was advised not to have children (Reid, 1986, 13).

Robinson was depressed for a long time, but eventually decided to continue studying mathematics. She returned to Berkeley and completed her PhD in 1948 under the supervision of Alfred Tarski. The first-order theory of the real numbers had been shown to be decidable by Tarski, and from Gödel's work it followed that the first-order theory of the natural numbers is undecidable. It was a major open problem whether the first-order theory of the rationals is decidable or not. In her thesis (1949), Robinson proved that it was not.

Interested in decision problems, Robinson next attempted to find a solution to Hilbert's tenth problem. This problem was one of a famous list of 23 mathematical problems posed by David Hilbert in 1900. The tenth problem asks whether there is an algorithm that will answer, in a finite amount of time, whether or not a polynomial equation with integer coefficients, such as $3x^2 - 2y + 3 = 0$, has a solution in the integers. Such questions are known as *Diophantine problems*. After some initial successes, Robinson joined forces with Martin Davis and Hilary Putnam, who were also working on the problem. They succeeded in showing that exponential Diophantine problems (where the unknowns may also appear as exponents) are undecidable, and showed that a certain conjecture (later called "J.R.") implies that Hilbert's tenth problem is undecidable (Davis et al., 1961). Robinson continued to work on the problem throughout the 1960s. In 1970, the young Russian mathematician Yuri Matijasevich finally proved the J.R. hypothesis. The combined result is now called the Matijasevich–Robinson–Davis–Putnam theorem, or MRDP theorem for short. Matijasevich and Robinson became friends and collaborated on several papers. In a letter to Matijasevich, Robinson once wrote that "actually I am very pleased that working together (thousands of miles apart) we are obviously making more progress than either one of us could alone" (Matijasevich, 1992, 45).

Robinson was the first female president of the American Mathematical Society, and the first woman to be elected to the National Academy of Science. She died on July 30, 1985 at the age of 65 after being diagnosed with leukemia.

Further Reading Robinson's mathematical papers are available in her *Collected Works* (Robinson, 1996), which also includes a reprint of her National

Academy of Sciences biographical memoir (Feferman, 1994). Robinson's older sister Constance Reid published an "Autobiography of Julia," based on interviews (Reid, 1986), as well as a full memoir (Reid, 1996). A short documentary about Robinson and Hilbert's tenth problem was directed by George Csicsery (Csicsery, 2016). For a brief memoir about Yuri Matijasevich's collaborations with Robinson, and her influence on his work, see (Matijasevich, 1992).

---

content/history/biographies/bertrand-russell.tex

75.8 Bertrand Russell

his:bio:russ:sec

Bertrand Russell is hailed as one of the founders of modern analytic philosophy. Born May 18, 1872, Russell was not only known for his work in philosophy and logic, but wrote many popular books in various subject areas. He was also an ardent political activist throughout his life.

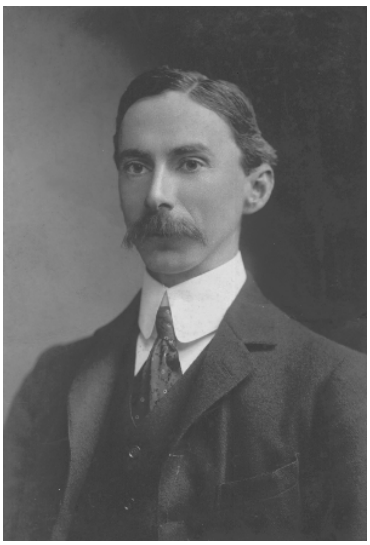


Figure 75.8: Bertrand Russell

Russell was born in Trellech, Monmouthshire, Wales. His parents were members of the British nobility. They were free-thinkers, and even made friends with the radicals in Boston at the time. Unfortunately, Russell's parents died when he was young, and Russell was sent to live with his grandparents. There, he was given a religious upbringing (something his parents had wanted to avoid at all costs). His grandmother was very strict in all matters of morality. During adolescence he was mostly homeschooled by private tutors.

Russell's influence in analytic philosophy, and especially logic, is tremendous. He studied mathematics and philosophy at Trinity College, Cambridge, where he was influenced by the mathematician and philosopher Alfred North Whitehead. In 1910, Russell and Whitehead published the first volume of *Principia Mathematica*, where they championed the view that mathematics is reducible to logic. He went on to publish hundreds of books, essays and political pamphlets. In 1950, he won the Nobel Prize for literature.

Russell's was deeply entrenched in politics and social activism. During World War I he was arrested and sent to prison for six months due to pacifist activities and protest. While in prison, he was able to write and read, and claims to have found the experience "quite agreeable." He remained a pacifist throughout his life, and was again incarcerated for attending a nuclear disarmament rally in 1961. He also survived a plane crash in 1948, where the only

survivors were those sitting in the smoking section. As such, Russell claimed that he owed his life to smoking. Russell was married four times, but had a reputation for carrying on extra-marital affairs. He died on February 2, 1970 at the age of 97 in Penrhyndeudraeth, Wales.

Further Reading Russell wrote an autobiography in three parts, spanning his life from 1872–1967 (Russell, 1967, 1968, 1969). The Bertrand Russell Research Centre at McMaster University is home of the Bertrand Russell archives. See their website at Duncan (2015), for information on the volumes of his collected works (including searchable indexes), and archival projects. Russell's paper *On Denoting* (Russell, 1905) is a classic of 20th century analytic philosophy.

The Stanford Encyclopedia of Philosophy entry on Russell (Irvine, 2015) has sound clips of Russell speaking on Desire and Political theory. Many video interviews with Russell are available online. To see him talk about smoking and being involved in a plane crash, e.g., see Russell (n.d.). Some of Russell's works, including his *Introduction to Mathematical Philosophy* are available as free audiobooks on LibriVox (n.d.).

---

content/history/biographies/alfred-tarski.tex

75.9 Alfred Tarski

his:bio:tarsec

Figure 75.9: Alfred Tarski

Alfred Tarski was born on January 14, 1901 in Warsaw, Poland (then part of the Russian Empire). Described as "Napoleonic," Tarski was boisterous, talkative, and intense. His energy was often reflected in his lectures—he once set fire to a wastebasket while disposing of a cigarette during a lecture, and was forbidden from lecturing in that building again.

Tarski had a thirst for knowledge from a young age. Although later in life he would tell students that he studied logic because it was the only class in which he got a B, his high school records show that he got A's across the board—even in logic. He studied at the University of Warsaw from 1918 to 1924. Tarski first intended to study biology, but became interested in mathematics, philosophy, and logic, as the university was the center of the Warsaw School of Logic and Philosophy. Tarski earned his doctorate in 1924 under the supervision of Stanisław Leśniewski.

Before emigrating to the United States in 1939, Tarski completed some of his most important work while working as a secondary school teacher in Warsaw. His work on logical consequence and logical truth were written during this time. In 1939, Tarski was visiting the United States for a lecture tour. During his visit, Germany invaded Poland, and because of his Jewish heritage, Tarski could not return. His wife and children remained in Poland until the end of the war, but were then able to emigrate to the United States as well. Tarski taught at Harvard, the College of the City of New York, and the Institute for Advanced Study at Princeton, and finally the University of California, Berkeley. There he founded the multidisciplinary program in Logic and the Methodology of Science. Tarski died on October 26, 1983 at the age of 82.

Further Reading For more on Tarski's life, see the biography *Alfred Tarski: Life and Logic* (Feferman and Feferman, 2004). Tarski's seminal works on logical consequence and truth are available in English in (Corcoran, 1983). All of Tarski's original works have been collected into a four volume series, (Tarski, 1981).

---

content/history/biographies/alan-turing.tex

75.10 Alan Turing

his:bio:tur:sec

Alan Turing was born in Maida Vale, London, on June 23, 1912. He is considered the father of theoretical computer science. Turing's interest in the physical sciences and mathematics started at a young age. However, as a boy his interests were not represented well in his schools, where emphasis was placed on literature and classics. Consequently, he did poorly in school and was reprimanded by many of his teachers.

Turing attended King's College, Cambridge as an undergraduate, where he studied mathematics. In 1936 Turing developed (what is now called) the Turing machine as an attempt to precisely define the notion of a computable function and to prove the undecidability of the decision problem. He was beaten to the result by Alonzo Church, who proved the result via his own lambda calculus. Turing's paper was still published with reference to Church's result. Church invited Turing to Princeton, where he spent 1936–1938, and obtained a doctorate under Church.

Despite his interest in logic, Turing's earlier interests in physical sciences remained prevalent. His practical skills were put to work during his service with the British cryptanalytic department at Bletchley Park during World War II. Turing was a central figure in cracking the cypher used by German Naval communications—the Enigma code. Turing's expertise in statistics and cryptography, together with the introduction of electronic machinery, gave the team the ability to crack the code by creating a de-crypting machine called a "bombe." His ideas also helped in the creation of the world's first pro-

grammable electronic computer, the Colossus, also used at Bletchley park to break the German Lorenz cypher.

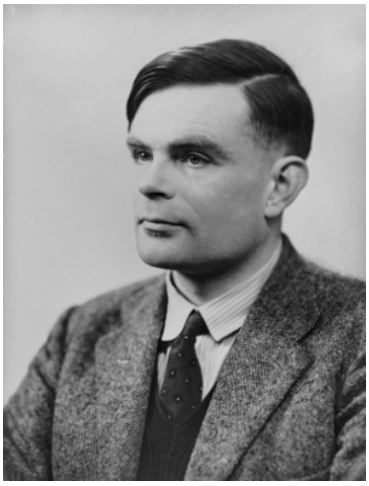


Figure 75.10: Alan Turing

Turing was gay. Nevertheless, in 1942 he proposed to Joan Clarke, one of his teammates at Bletchley Park, but later broke off the engagement and confessed to her that he was homosexual. He had several lovers throughout his lifetime, although homosexual acts were then criminal offences in the UK. In 1952, Turing's house was burgled by a friend of his lover at the time, and when filing a police report, Turing admitted to having a homosexual relationship, under the impression that the government was on their way to legalizing homosexual acts. This was not true, and he was charged with gross indecency. Instead of going to prison, Turing opted for a hormone treatment that reduced libido. Turing was found dead on June 8, 1954, of a cyanide overdose—most likely suicide. He was given a royal pardon by Queen Elizabeth II in 2013.

Further Reading For a comprehensive biography of Alan Turing, see Hodges (2014). Turing's life and work inspired a play, *Breaking the Code*, which was produced in 1996 for TV starring Derek Jacobi as Turing. *The Imitation Game*, an Academy Award nominated film starring Bendict Cumberbatch and Kiera Knightley, is also loosely based on Alan Turing's life and time at Bletchley Park (Tyldum, 2014).

Radiolab (2012) has several podcasts on Turing's life and work. BBC Horizon's documentary *The Strange Life and Death of Dr. Turing* is available to watch online (Sykes, 1992). (Theelen, 2012) is a short video of a working LEGO Turing Machine—made to honour Turing's centenary in 2012.

Turing's original paper on Turing machines and the decision problem is Turing (1937).

---

content/history/biographies/ernst-zermelo.tex

75.11 Ernst Zermelo

his:bio:zer:sec

Ernst Zermelo was born on July 27, 1871 in Berlin, Germany. He had five sisters, though his family suffered from poor health and only three survived to adulthood. His parents also passed away when he was young, leaving him and his siblings orphans when he was seventeen. Zermelo had a deep interest in the arts, and especially in poetry. He was known for being sharp, witty, and critical.

His most celebrated mathematical achievements include the introduction of the axiom of choice (in 1904), and his axiomatization of set theory (in 1908).

Zermelo's interests at university were varied. He took courses in physics, mathematics, and philosophy. Under the supervision of Hermann Schwarz, Zermelo completed his dissertation *Investigations in the Calculus of Variations* in 1894 at the University of Berlin. In 1897, he decided to pursue more studies at the University of Göttigen, where he was heavily influenced by the foundational work of David Hilbert. In 1899 he became eligible for professorship, but did not get one until eleven years later—possibly due to his strange demeanour and "nervous haste."

Figure 75.11: Ernst Zermelo

Zermelo finally received a paid professorship at the University of Zurich in 1910, but was forced to retire in 1916 due to tuberculosis. After his recovery, he was given an honourary professorship at the University of Freiburg in 1921. During this time he worked on foundational mathematics. He became irritated with the works of Thoralf Skolem and Kurt Gödel, and publicly criticized their approaches in his papers. He was dismissed from his position at Freiburg in 1935, due to his unpopularity and his opposition to Hitler's rise to power in Germany.

The later years of Zermelo's life were marked by isolation. After his dismissal in 1935, he abandoned mathematics. He moved to the country where he lived modestly. He married in 1944, and became completely dependent on his wife as he was going blind. Zermelo lost his sight completely by 1951. He passed away in Günterstal, Germany, on May 21, 1953.

Further Reading For a full biography of Zermelo, see Ebbinghaus (2015). Zermelo's seminal 1904 and 1908 papers are available to read in the original German (Zermelo, 1904, 1908b). Zermelo's collected works, including his writing on physics, are available in English translation in (Ebbinghaus et al., 2010;

Ebbinghaus and Kanamori, 2013).

Chapter 76

History and Mythology of Set Theory

This chapter includes the historical prelude from Tim Button's Open Set Theory text.

---

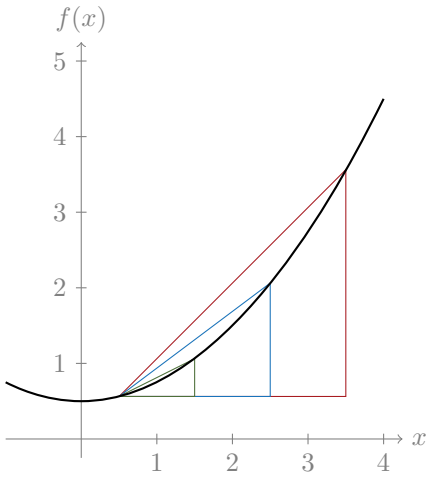
content/history/set-theory/infinitesimals.tex

76.1 Infinitesimals and Differentiation

his:set:infinitesimals:sec

Newton and Leibniz discovered the calculus (independently) at the end of the 17th century. A particularly important application of the calculus was *differentiation*. Roughly speaking, differentiation aims to give a notion of the "rate of change", or gradient, of a function at a point.

Here is a vivid way to illustrate the idea. Consider the function $f(x) = {}^{x^2}/_4 + {}^1/_2$, depicted in black below:



Suppose we want to find the gradient of the function at $c = 1/2$. We start by drawing a triangle whose hypotenuse approximates the gradient at that point, perhaps the red triangle above. When β is the base length of our triangle, its height is $f(1/2 + \beta) - f(1/2)$, so that the gradient of the hypotenuse is:

$$\frac{f(1/2 + \beta) - f(1/2)}{\beta}.$$

So the gradient of our red triangle, with base length 3, is exactly 1. The hypotenuse of a smaller triangle, the blue triangle with base length 2, gives a better approximation; its gradient is $3/4$. A yet smaller triangle, the green triangle with base length 1, gives a yet better approximation; with gradient $1/2$.

Ever-smaller triangles give us ever-better approximations. So we might say something like this: the hypotenuse of a triangle with an *infinitesimal* base length gives us the gradient at $c = 1/2$ itself. In this way, we would obtain a formula for the (first) derivative of the function f at the point c:

$$f'(c) = \frac{f(c + \beta) - f(c)}{\beta} \text{ where } \beta \text{ is infinitesimal.}$$

And, roughly, this is what Newton and Leibniz said.

However, since they have said this, we must ask them: what is an *infinitesimal*? A serious dilemma arises. If $\beta = 0$, then f' is ill-defined, for it involves dividing by 0. But if $\beta > 0$, then we just get an *approximation* to the gradient, and not the gradient itself.

This is not an anachronistic concern. Here is Berkeley, criticizing Newton's followers:

> I admit that signs may be made to denote either any thing or nothing: and consequently that in the original notation $c + \beta$, β might

> have signified either an increment or nothing. But then which of these soever you make it signify, you must argue consistently with such its signification, and not proceed upon a double meaning: Which to do were a manifest sophism. (Berkeley 1734, §XIII, variables changed to match preceding text)

To defend the infinitesimal calculus against Berkeley, one might reply that the talk of "infinitesimals" is merely figurative. One might say that, so long as we take a *really small* triangle, we will get a *good enough* approximation to the tangent. Berkeley had a reply to this too: whilst that might be good enough for engineering, it undermines the *status* of mathematics, for

> we are told that *in rebus mathematicis errores quàm minimi non sunt contemnendi*. [In the case of mathematics, the smallest errors are not to be neglected.] (Berkeley, 1734, §IX)

The italicised passage is a near-verbatim quote from Newton's own *Quadrature of Curves* (1704).

Berkeley's philosophical objections are deeply incisive. Nevertheless, the calculus was a massively successful enterprise, and mathematicians continued to use it without falling into error.

---

content/history/set-theory/limits.tex

76.2 Rigorous Definition of Limits

his:set:limits:sec

These days, the standard solution to the foregoing problem is to get rid of the infinitesimals. Here is how.

We saw that, as β gets smaller, we get better approximations of the gradient. Indeed, as β gets arbitrarily close to 0, the value of $f'(c)$ "tends without limit" to the gradient we want. So, instead of considering what happens *at* $\beta = 0$, we need only consider the *trend* of $f'(c)$ as β approaches 0.

Put like this, the general challenge is to make sense of claims of this shape:

$$\text{As } x \text{ approaches } c,\ g(x) \text{ tends without limit to } \ell.$$

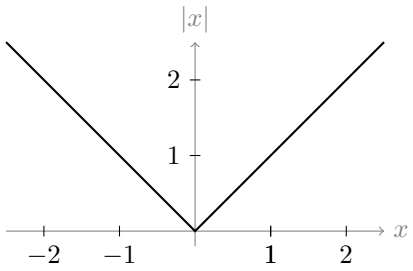
which we can write more compactly as follows:

$$\lim_{x \to c} g(x) = \ell.$$

In the 19th century, building upon earlier work by Cauchy, Weierstrass offered a perfectly rigorous definition of this expression. The idea is indeed that we can make $g(x)$ as close as we like to ℓ, by making x suitably close to c. More precisely, we stipulate that $\lim_{x \to c} g(x) = \ell$ will mean:

$$(\forall \varepsilon > 0)(\exists \delta > 0)\forall x \left(|x - c| < \delta \to |g(x) - \ell| < \varepsilon\right).$$

The vertical bars here indicate absolute magnitude. That is, $|x| = x$ when $x \geq 0$, and $|x| = -x$ when $x < 0$; you can depict *that* function as follows:



So the definition says roughly this: you can make your "error" less than ε (i.e., $|g(x) - \ell| < \varepsilon$) by choosing arguments which are no more than δ away from c (i.e., $|x - c| < \delta$).

Having defined the notion of a limit, we can use it to avoid infinitesimals altogether, stipulating that the gradient of f at c is given by:

$$f'(c) = \lim_{x \to 0} \left(\frac{f(c+x) - f(c)}{x} \right) \text{ where a limit exists.}$$

It is important, though, to realise why our definition needs the caveat "where a limit exists". To take a simple example, consider $f(x) = |x|$, whose graph we just saw. Evidently, $f'(0)$ is ill-defined: if we approach 0 "from the right", the gradient is always 1; if we approach 0 "from the left", the gradient is always -1; so the limit is undefined. As such, we might add that a function f is *differentiable* at x iff such a limit exists.

We have seen how to handle differentiation using the notion of a *limit*. We can use the same notion to define the idea of a *continuous* function. (Bolzano had, in effect, realised this by 1817.) The Cauchy–Weierstrass treatment of continuity is as follows. Roughly: a function f is continuous (at a point) provided that, if you demand a certain amount of precision concerning the output of the function, you can guarantee this by insisting upon a certain amount of precision concerning the input of the function. More precisely: f is continuous at c provided that, as x tends to zero, the difference between $f(c + x)$ and $f(c)$ itself tends to 0. Otherwise put: f is *continuous* at c iff $f(c) = \lim_{x \to c} f(x)$.

To go any further would just lead us off into real analysis, when our subject matter is set theory. So now we should pause, and state the moral. During the 19th century, mathematicians learnt how to do without infinitesimals, by invoking a rigorously defined notion of a *limit*.

---

content/history/set-theory/pathologies.tex

76.3 Pathologies

his:set:pathology:sec

However, the definition of a *limit* turned out to allow for some rather "pathological" constructions.

Around the 1830s, Bolzano discovered a function which was *continuous everywhere*, but *differentiable nowhere*. (Unfortunately, Bolzano never published this; the idea was first encountered by mathematicians in 1872, thanks to Weierstrass's independent discovery of the same idea.)[1] This was, to say the least, rather surprising. It is easy to find functions, such as $|x|$, which are continuous everywhere but not differentiable at a particular point. But a function which is continuous everywhere but differentiable *nowhere* is a very different beast. Consider, for a moment, how you might try to draw such a function. To ensure it is continuous, you must be able to draw it without ever removing your pen from the page; but to ensure it is differentiable nowhere, you would have to abruptly change the direction of your pen, constantly.

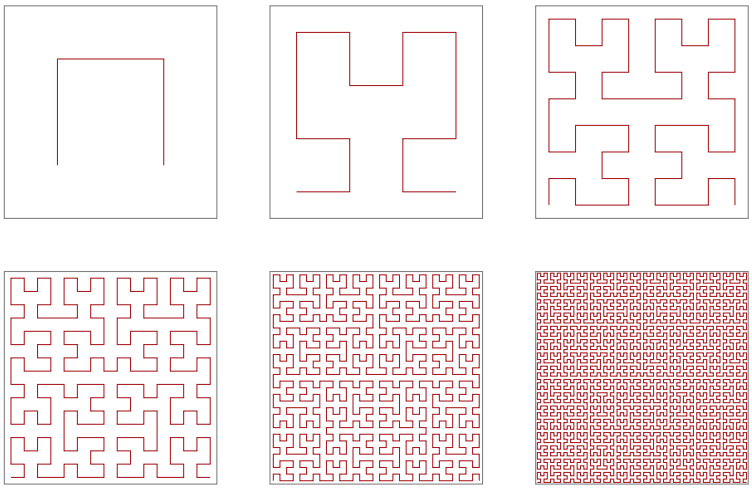
Further "pathologies" followed. In January 5 1874, Cantor wrote a letter to Dedekind, posing the problem:

> Can a surface (say a square including its boundary) be one-to-one correlated to a line (say a straight line including its endpoints) so that to every point of the surface there corresponds a point of the line, and conversely to every point of the line there corresponds a point of the surface?
>
> It still seems to me at the moment that the answer to this question is very difficult—although here too one is so impelled to say *no* that one would like to hold the proof to be almost superfluous. [Quoted in Gouvêa 2011]

But, in 1877, Cantor proved that he had been wrong. In fact, a line and a square have exactly the same number of points. He wrote on 29 June 1877 to Dedekind "*je le vois, mais je ne le crois pas*"; that is, "I see it, but I don't believe it". In the "received history" of mathematics, this is often taken to indicate just how *literally incredible* these new results were to the mathematicians of the time. (The correspondence is presented in Gouvêa (2011), and we return to it in section 76.4. Cantor's proof is outlined in section 76.5.)

Inspired by Cantor's result, Peano started to consider whether it might be possible to map a line *smoothly* onto a plane. This would be a *curve which fills space*. In 1890, Peano constructed just such a curve. This is truly counter-intuitive: Euclid had defined a line as "breadthless length" (Book I, Definition 2), but Peano had shown that, by curling up a line appropriately, its length can be turned into breadth. In 1891, Hilbert described a slightly more intuitive space-filling curve, together with some pictures illustrating it. The curve is constructed in sequence, and here are the first six stages of the construction:

[1] The history is documented in extremely thorough footnotes to the Wikipedia article on the Weierstrass function.



In the limit—a notion which had, by now, received rigorous definition—the entire square is filled in solid red. And, in passing, Hilbert's curve is continuous everywhere but differentiable nowhere; intuitively because, in the infinite limit, the function abruptly changes direction at every moment. (We will outline Hilbert's construction in more detail in section 76.6.)

For better or worse, these "pathological" geometric constructions were treated as a reason to doubt appeals to geometric intuition. They became something approaching *propaganda* for a new way of doing mathematics, which would culminate in set theory. In the later myth-building of the subject, it was repeated, often, that these results were both perfectly rigorous and perfectly shocking. They therefore served a dual purpose: as a warning against relying upon geometric intuition, and as a demonstration of the fertility of new ways of thinking.

---

content/history/set-theory/mythology.tex

76.4 More Myth than History?

his:set:mythology:sec

Looking back on these events with more than a century of hindsight, we must be careful not to take these verdicts on trust. The results were certainly novel, exciting, and surprising. But how truly shocking were they? And did they really demonstrate that we should not rely on geometric intuition?

On the question of shock, Gouvêa (2011) points out that Cantor's famous note to Dedekind, "*je le vois, mais je ne le crois pas*" is taken rather out of context. Here is more of that context (quoted from Gouvêa):

> Please excuse my zeal for the subject if I make so many demands upon your kindness and patience; the communications which I lately sent you are even for me so unexpected, so new, that I can have no peace of mind until I obtain from you, honoured friend, a

> decision about their correctness. So long as you have not agreed with me, I can only say: *je le vois, mais je ne le crois pas.*

Cantor knew his result was "so unexpected, so new". But it is doubtful that he ever found his result *unbelievable*. As Gouvêa points out, he was simply asking Dedekind to check the proof he had offered.

On the question of geometric intuition: Peano published his space-filling curve without including any diagrams. But when Hilbert published his curve, he explained his purpose: he would provide readers with a clear way to understand Peano's result, if they "help themselves to the following geometric intuition"; whereupon he included a series of *diagrams* just like those provided in section 76.3.

More generally: whilst diagrams have fallen rather out of fashion in published proofs, there is no getting round the fact that mathematicians *frequently* use diagrams when proving things. (Roughly put: good mathematicians know when they can rely upon geometric intuition.)

In short: don't believe the hype; or at least, don't just take it on trust. For more on this, you could read Giaquinto (2007).

---

content/history/set-theory/cantor-plane.tex

76.5 Cantor on the Line and the Plane

his:set:cantorplane:sec

Some of the circumstances surrounding the proof of Schröder-Bernstein tie in with the history we discussed in section 76.3. Recall that, in 1877, Cantor proved that there are exactly as many points on a square as on one of its sides. Here, we will present his (first attempted) proof.

Let L be the unit line, i.e., the set of points $[0, 1]$. Let S be the unit square, i.e., the set of points $\mathrm{L} \times \mathrm{L}$. In these terms, Cantor proved that $\mathrm{L} \approx \mathrm{S}$. He wrote a note to Dedekind, essentially containing the following argument.

Theorem 76.1. *$L \approx S$*

his:set:cantorplane:thm:cantorplane

Proof: first part.. Fix $a, b \in \mathrm{L}$. Write them in binary notation, so that we have infinite sequences of 0s and 1s, a_1, a_2, ..., and b_1, b_2, ..., such that:

$$a = 0.a_1a_2a_3a_4\ldots$$
$$b = 0.b_1b_2b_3b_4\ldots$$

Now consider the function $f\colon \mathrm{S} \to \mathrm{L}$ given by

$$f(a, b) = 0.a_1b_1a_2b_2a_3b_3a_4b_4\ldots$$

Now f is an injection, since if $f(a, b) = f(c, d)$, then $a_n = c_n$ and $b_n = d_n$ for all $n \in \mathbb{N}$, so that $a = c$ and $b = d$. □

Unfortunately, as Dedekind pointed out to Cantor, this does not answer the original question. Consider $0.\dot{1}\dot{0} = 0.1010101010\ldots$. We need that $f(a, b) = 0.\dot{1}\dot{0}$, where:

$$a = 0.\dot{1}\dot{1} = 0.111111\ldots$$
$$b = 0$$

But $a = 0.\dot{1}\dot{1} = 1$. So, when we say "write a and b in binary notation", we have to choose *which* notation to use; and, since f is to be a *function*, we can use only *one* of the two possible notations. But if, for example, we use the simple notation, and write a as "$1.000\ldots$", then we have no pair $\langle a, b\rangle$ such that $f(a, b) = 0.\dot{1}\dot{0}$.

To summarise: Dedekind pointed out that, given the possibility of certain recurring decimal expansions, Cantor's function f is an injection but *not* a surjection. So Cantor has shown only that $\mathrm{S} \preceq \mathrm{L}$ and *not* that $\mathrm{S} \approx \mathrm{L}$.

Cantor wrote back to Dedekind almost immediately, essentially suggesting that the proof could be completed as follows:

Proof: completed.. So, we have shown that $\mathrm{S} \preceq \mathrm{L}$. But there is obviously an injection from L to S: just lay the line flat along one side of the square. So $\mathrm{L} \preceq \mathrm{S}$ and $\mathrm{S} \preceq \mathrm{L}$. By Schröder–Bernstein (Theorem 4.25), $\mathrm{L} \approx \mathrm{S}$. □

But of course, Cantor could not complete the last line in these terms, for the Schröder-Bernstein Theorem was not yet proved. Indeed, although Cantor would subsequently formulate this as a general conjecture, it was not satisfactorily proved until 1897. (And so, later in 1877, Cantor offered a different proof of Theorem 76.1, which did not go via Schröder–Bernstein.)

---

content/history/set-theory/hilbert-curve.tex

76.6 Appendix: Hilbert's Space-filling Curves

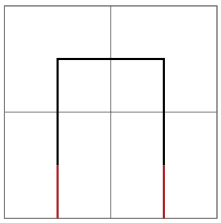
his:set:hilbertcurve:sec

In chapter section 76.3, we mentioned that Cantor's proof that a line and a square have exactly the same number of points (Theorem 76.1) prompted Peano to ask whether there might be a space-filling *curve*. He obtained a positive answer in 1890. In this section, we explain (in a hand-wavy way) how to construct Hilbert's space-filling curve (with a tiny tweak).[2]

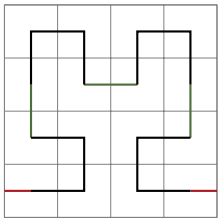
We must define a function, h, as the limit of a sequence of functions h_1, h_2, h_3, ... We first describe the construction. Then we show it is space-filling. Then we show it is a curve.

We will take h's range to be the unit square, S. Here is our first approximation to h, i.e., h_1:

[2] For a more rigorous explanation, see Rose (2010). The tweak amounts to the inclusion of the red parts of the curves below. This makes it slightly easier to check that the curve is continuous.

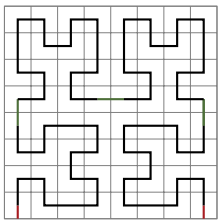


To keep track of things, we have imposed a 2×2 grid on the square. We can think of the curve starting in the bottom left quarter, moving to the top left, then to the top right, then finally to the bottom right. Here is the second stage in the construction, i.e., h_2:



The different colours will help explain how h_2 was constructed. We first place scaled-down copies of the non-red bit of h_1 into the bottom left, top left, top right, and bottom right of our square (drawn in black). We then connect these four figures (with green lines). Finally, we connect our figure to the boundary of the square (with red lines).

Now to h_3. Just as h_2 was made from four connected, scaled-down copies of the non-red bit of h_1, so h_3 is made up of four scaled-down copies of the non-red bit of h_2 (drawn in black), which are then joined together (with green lines) and finally connected to the boundary of the square (with red lines).



And now we see the general pattern for defining h_{n+1} from h_n. At last we define the curve h *itself* by considering the point-by-point limit of these successive functions h_1, h_2, ... That is, for each $x \in \mathrm{S}$:

$$h(x) = \lim_{n \to \infty} h_n(x)$$

We now show that this curve fills space. When we draw the curve h_n, we impose a $2^n \times 2^n$ grid onto S. By Pythagoras's Theorem, the diagonal of each

grid-location is of length:

$$\sqrt{(1/2^n)^2 + (1/2^n)^2} = 2^{(\frac{1}{2}-n)}$$

and evidently h_n passes through every grid-location. So each point in S is *at most* $2^{(\frac{1}{2}-n)}$ distance away from some point on h_n. Now, h is defined as the limit of the functions h_1, h_2, h_3, ... So the maximum distance of any point from h is given by:

$$\lim_{n\to\infty} 2^{(\frac{1}{2}-n)} = 0.$$

That is: every point in S is 0 distance from h. In other words, every point of S lies *on* the curve. So h fills space!

It remains to show that h is, indeed, a *curve*. To show this, we must define the notion. The modern definition builds on one given by Jordan in 1887 (i.e., only a few years before the first space-filling curve was provided):

Definition 76.2. A curve is a continuous map from L to $\mathbb{R}^2$.

This is fairly intuitive: a curve is, intuitively, a "smooth" map which takes a canonical line onto the plane $\mathbb{R}^2$. Our function, h, is indeed a map from L to $\mathbb{R}^2$. So, we just need to show that h is continuous. We defined continuity in section 76.2 using ε/δ notation. In the vernacular, we want to establish the following: *If you specify a point p in S, together with any desired level of precision ε, we can find an open section of L such that, given any x in that open section, $h(x)$ is within ε of p.*

So: assume that you have specified p and ε. This is, in effect, to draw a circle with centre p and radius ε on S. (The circle might spill off the edge of S, but that doesn't matter.) Now, recall that, when describing the function h_n, we drew a $2^n \times 2^n$ grid upon S. It is obvious that, no matter how small ε is, there is some n such that some individual grid-location of the $2^n \times 2^n$ grid on S lies wholly within the circle with centre p and radius ε.

So, take that n, and let I be the largest open part of L which h_n maps wholly into the relevant grid location. (It is clear that (a, b) exists, since we already noted that h_n passes through every grid-location in the $2^n \times 2^n$ grid.) It now suffices to show to show that, whenever $x \in I$ the point $h(x)$ lies in that same grid-location. And to do *this*, it suffices to show that $h_m(x)$ lies in that same grid location, for any $m > n$. But this is obvious. If we consider what happens with h_m for $m > n$, we see that exactly the "same part" of the unit interval is mapped into the same grid-location; we just map it into that region in an increasingly stretched-out, wiggly fashion.

Part XVIII

Reference

This part collects various lists of alphabets, notations, definitions, rules, etc., for inclusion as appendices in a textbook.

Chapter 77

The Greek Alphabet

Alpha	α	A	Nu	ν	N
Beta	β	B	Xi	ξ	Ξ
Gamma	γ	Γ	Omicron	o	O
Delta	δ	Δ	Pi	π	Π
Epsilon	ε	E	Rho	ρ	P
Zeta	ζ	Z	Sigma	σ	Σ
Eta	η	H	Tau	τ	T
Theta	θ	Θ	Upsilon	υ	Υ
Iota	ι	I	Phi	φ	Φ
Kappa	κ	K	Chi	χ	X
Lambda	λ	Λ	Psi	ψ	Ψ
Mu	μ	M	Omega	ω	Ω

Chapter 78

The Fraktur Alphabet

A	𝔄	a	𝔞	N	𝔑	n	𝔫
B	𝔅	b	𝔟	O	𝔒	o	𝔬
C	ℭ	c	𝔠	P	𝔓	p	𝔭
D	𝔇	d	𝔡	Q	𝔔	q	𝔮
E	𝔈	e	𝔢	R	ℜ	r	𝔯
F	𝔉	f	𝔣	S	𝔖	s	𝔰
G	𝔊	g	𝔤	T	𝔗	t	𝔱
H	ℌ	h	𝔥	U	𝔘	u	𝔲
I	ℑ	i	𝔦	V	𝔙	v	𝔳
J	𝔍	j	𝔧	W	𝔚	w	𝔴
K	𝔎	k	𝔨	X	𝔛	x	𝔵
L	𝔏	l	𝔩	Y	𝔜	y	𝔶
M	𝔐	m	𝔪	Z	ℨ	z	𝔷

Photo Credits

Georg Cantor, p. 928: Portrait of Georg Cantor by Otto Zeth courtesy of the Universitätsarchiv, Martin-Luther Universität Halle–Wittenberg. UAHW Rep. 40-VI, Nr. 3 Bild 102.

Alonzo Church, p. 929: Portrait of Alonzo Church, undated, photographer unknown. Alonzo Church Papers; 1924–1995, (C0948) Box 60, Folder 3. Manuscripts Division, Department of Rare Books and Special Collections, Princeton University Library. © Princeton University. The Open Logic Project has obtained permission to use this image for inclusion in non-commercial OLP-derived materials. Permission from Princeton University is required for

any other use.

Gerhard Gentzen, p. 930: Portrait of Gerhard Gentzen playing ping-pong courtesy of Ekhart Mentzler-Trott.

Kurt Gödel, p. 931: Portrait of Kurt Gödel, ca. 1925, photographer unknown. From the Shelby White and Leon Levy Archives Center, Institute for Advanced Study, Princeton, NJ, USA, on deposit at Princeton University Library, Manuscript Division, Department of Rare Books and Special Collections, Kurt Gödel Papers, (C0282), Box 14b, #110000. The Open Logic Project has obtained permission from the Institute's Archives Center to use this image for inclusion in non-commercial OLP-derived materials. Permission from the Archives Center is required for any other use.

Emmy Noether, p. 933: Portrait of Emmy Noether, ca. 1922, courtesy of the Abteilung für Handschriften und Seltene Drucke, Niedersächsische Staats- und Universitätsbibliothek Göttingen, Cod. Ms. D. Hilbert 754, Bl. 14 Nr. 73. Restored from an original scan by Joel Fuller.

Rózsa Péter, p. 934: Portrait of Rózsa Péter, undated, photographer unknown. Courtesy of Béla Andrásfai.

Julia Robinson, p. 935: Portrait of Julia Robinson, unknown photographer, courtesy of Neil D. Reid. The Open Logic Project has obtained permission to use this image for inclusion in non-commercial OLP-derived materials. Permission is required for any other use.

Bertrand Russell, p. 937: Portrait of Bertrand Russell, ca. 1907, courtesy of the William Ready Division of Archives and Research Collections, McMaster University Library. Bertrand Russell Archives, Box 2, f. 4.

Alfred Tarski, p. 938: Passport photo of Alfred Tarski, 1939. Cropped and restored from a scan of Tarski's passport by Joel Fuller. Original courtesy of Bancroft Library, University of California, Berkeley. Alfred Tarski Papers, Banc MSS 84/49. The Open Logic Project has obtained permission to use this image for inclusion in non-commercial OLP-derived materials. Permission from Bancroft Library is required for any other use.

Alan Turing, p. 940: Portrait of Alan Mathison Turing by Elliott & Fry, 29 March 1951, NPG x82217, © National Portrait Gallery, London. Used under a Creative Commons BY-NC-ND 3.0 license.

Ernst Zermelo, p. 941: Portrait of Ernst Zermelo, ca. 1922, courtesy of the Abteilung für Handschriften und Seltene Drucke, Niedersächsische Staats- und

Universitätsbibliothek Göttingen, Cod. Ms. D. Hilbert 754, Bl. 6 Nr. 25.

Bibliography

Andrásfai, Béla. 1986. Rózsa (Rosa) Péter. *Periodica Polytechnica Electrical Engineering* 30(2-3): 139–145. URL http://www.pp.bme.hu/ee/article/view/4651.

Aspray, William. 1984. The Princeton mathematics community in the 1930s: Alonzo Church. URL http://www.princeton.edu/mudd/finding_aids/mathoral/pmc05.htm. Interview.

Baaz, Matthias, Christos H. Papadimitriou, Hilary W. Putnam, Dana S. Scott, and Charles L. Harper Jr. 2011. *Kurt Gödel and the Foundations of Mathematics: Horizons of Truth*. Cambridge: Cambridge University Press.

Banach, Stefan and Alfred Tarski. 1924. Sur la décomposition des ensembles de points en parties respectivement congruentes. *Fundamenta Mathematicae* 6: 244–77.

Benacerraf, Paul. 1965. What numbers could not be. *The Philosophical Review* 74(1): 47–73.

Berkeley, George. 1734. *The Analyst; or, a Discourse Adressed to an Infidel Mathematician*.

Boolos, George. 1971. The iterative conception of set. *The Journal of Philosophy* 68(8): 215–31.

Boolos, George. 1989. Iteration again. *Philosophical Topics* 17(2): 5–21.

Boolos, George. 2000. Must we believe in set theory? In *Between Logic and Intuition: Essays in Honor of Charles Parsons*, eds. Gila Sher and Richard Tieszen, 257–68. Cambridge: Cambridge University Press.

Burali-Forti, Cesare. 1897. Una questione sui numeri transfiniti. *Rendiconti del Circolo Matematico di Palermo* 11: 154–64.

Button, Tim. 2021. Level theory, part 1: Axiomatizing the bare idea of a cumulative hierarchy of sets. *The Bulletin of Symbolic Logic* 27(4): 436–460.

Cantor, Georg. 1878. Ein Beitrag zur Mannigfaltigkeitslehre. *Journal für die reine und angewandte Mathematik* 84: 242–58.

Cantor, Georg. 1883. *Grundlagen einer allgemeinen Mannigfaltigkeitslehre. Ein mathematisch-philosophischer Versuch in der Lehre des Unendlichen.* Leipzig: Teubner.

Cantor, Georg. 1892. Über eine elementare Frage der Mannigfaltigkeitslehre. *Jahresbericht der deutschen Mathematiker-Vereinigung* 1: 75–8.

Cheng, Eugenia. 2004. How to write proofs: A quick quide. URL https://eugeniacheng.com/wp-content/uploads/2017/02/cheng-proofguide.pdf.

Church, Alonzo. 1936a. A note on the Entscheidungsproblem. *The Journal of Symbolic Logic* 1: 40–41.

Church, Alonzo. 1936b. An unsolvable problem of elementary number theory. *American Journal of Mathematics* 58: 345–363.

Cohen, Paul J. 1963. The independence of the continuum hypothesis. *Proceedings of the National Academy of Sciences of the United States of America* 24: 556–557.

Cohen, Paul J. 1966. *Set Theory and the Continuum Hypothesis.* Reading, MA: Benjamin.

Conway, John. 2006. The power of mathematics. In *Power*, eds. Alan Blackwell and David MacKay, Darwin College Lectures. Cambridge: Cambridge University Press. URL http://www.cs.toronto.edu/~mackay/conway.pdf.

Corcoran, John. 1983. *Logic, Semantics, Metamathematics.* Indianapolis: Hackett, 2nd ed.

Csicsery, George. 2016. Zala films: Julia Robinson and Hilbert's tenth problem. URL http://www.zalafilms.com/films/juliarobinson.html.

Dauben, Joseph. 1990. *Georg Cantor: His Mathematics and Philosophy of the Infinite.* Princeton: Princeton University Press.

Davis, Martin, Hilary Putnam, and Julia Robinson. 1961. The decision problem for exponential Diophantine equations. *Annals of Mathematics* 74(3): 425–436. URL http://www.jstor.org/stable/1970289.

Dedekind, Richard. 1888. *Was sind und was sollen die Zahlen?* Braunschweig: Vieweg.

Dick, Auguste. 1981. *Emmy Noether 1882–1935.* Boston: Birkhäuser.

du Sautoy, Marcus. 2014. A brief history of mathematics: Georg Cantor. URL http://www.bbc.co.uk/programmes/b00ss1j0. Audio Recording.

Duncan, Arlene. 2015. The Bertrand Russell Research Centre. URL http://russell.mcmaster.ca/.

Ebbinghaus, Heinz-Dieter. 2015. *Ernst Zermelo: An Approach to his Life and Work*. Berlin: Springer-Verlag.

Ebbinghaus, Heinz-Dieter, Craig G. Fraser, and Akihiro Kanamori. 2010. *Ernst Zermelo. Collected Works*, vol. 1. Berlin: Springer-Verlag.

Ebbinghaus, Heinz-Dieter and Akihiro Kanamori. 2013. *Ernst Zermelo: Collected Works*, vol. 2. Berlin: Springer-Verlag.

Enderton, Herbert B. 2019. Alonzo Church: Life and Work. In *The Collected Works of Alonzo Church*, eds. Tyler Burge and Herbert B. Enderton. Cambridge, MA: MIT Press.

Feferman, Anita and Solomon Feferman. 2004. *Alfred Tarski: Life and Logic*. Cambridge: Cambridge University Press.

Feferman, Solomon. 1994. Julia Bowman Robinson 1919–1985. *Biographical Memoirs of the National Academy of Sciences* 63: 1–28. URL http://www.nasonline.org/publications/biographical-memoirs/memoir-pdfs/robinson-julia.pdf.

Feferman, Solomon, John W. Dawson Jr., Stephen C. Kleene, Gregory H. Moore, Robert M. Solovay, and Jean van Heijenoort. 1986. *Kurt Gödel: Collected Works. Vol. 1: Publications 1929–1936*. Oxford: Oxford University Press.

Feferman, Solomon, John W. Dawson Jr., Stephen C. Kleene, Gregory H. Moore, Robert M. Solovay, and Jean van Heijenoort. 1990. *Kurt Gödel: Collected Works. Vol. 2: Publications 1938–1974*. Oxford: Oxford University Press.

Feferman, Solomon and Azriel Levy. 1963. Independence results in set theory by Cohen's method II. *Notices of the American Mathematical Society* 10: 593.

Fraenkel, Abraham. 1922. Über den Begriff 'definit' und die Unabhängigkeit des Auswahlaxioms. *Sitzungsberichte der Preussischen Akadademie der Wissenschaften, Physikalisch-mathematische Klasse* 253–257.

Frege, Gottlob. 1884. *Die Grundlagen der Arithmetik: Eine logisch mathematische Untersuchung über den Begriff der Zahl*. Breslau: Wilhelm Koebner. Translation in Frege (1953).

Frege, Gottlob. 1953. *Foundations of Arithmetic*, ed. J. L. Austin. Oxford: Basil Blackwell & Mott, 2nd ed.

Frey, Holly and Tracy V. Wilson. 2015. Stuff you missed in history class: Emmy Noether, mathematics trailblazer. URL https://www.iheart.com/podcast/stuff-you-missed-in-history-cl-21124503/episode/emmy-noether-mathematics-trailblazer-30207491/. Podcast audio.

Gentzen, Gerhard. 1935a. Untersuchungen über das logische Schließen I. *Mathematische Zeitschrift* 39: 176–210. English translation in Szabo (1969), pp. 68–131.

Gentzen, Gerhard. 1935b. Untersuchungen über das logische Schließen II. *Mathematische Zeitschrift* 39: 176–210, 405–431. English translation in Szabo (1969), pp. 68–131.

Giaquinto, Marcus. 2007. *Visual Thinking in Mathematics*. Oxford: Oxford University Press.

Gödel, Kurt. 1929. Über die Vollständigkeit des Logikkalküls [On the completeness of the calculus of logic]. Dissertation, Universität Wien. Reprinted and translated in Feferman et al. (1986), pp. 60–101.

Gödel, Kurt. 1931. über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme I [On formally undecidable propositions of *Principia Mathematica* and related systems I]. *Monatshefte für Mathematik und Physik* 38: 173–198. Reprinted and translated in Feferman et al. (1986), pp. 144–195.

Gödel, Kurt. 1938. The consistency of the axiom of choice and the generalized continuum hypothesis. *Proceedings of the National Academy of Sciences of the United States of America* 50: 1143–8.

Gouvêa, Fernando Q. 2011. Was Cantor surprised? *American Mathematical Monthly* 118(3): 198–209.

Grattan-Guinness, Ivor. 1971. Towards a biography of Georg Cantor. *Annals of Science* 27(4): 345–391.

Hammack, Richard. 2013. *Book of Proof*. Richmond, VA: Virginia Commonwealth University. URL http://www.people.vcu.edu/~rhammack/BookOfProof/BookOfProof.pdf.

Hartogs, Friedrich. 1915. Über das Problem der Wohlordnung. *Mathematische Annalen* 76: 438–43.

Hausdorff, Felix. 1914. Bemerkung über den Inhalt von Punktmengen. *Mathematische Annalen* 75: 428–34.

Heck, Richard Kimberly. 2012. *Reading Frege's Grundgesetze*. Oxford: Oxford University Press.

Heijenoort, Jean van. 1967. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*. Cambridge, MA: Harvard University Press.

Hilbert, David. 1891. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Mathematische Annalen* 38(3): 459–460.

Hilbert, David. 2013. *David Hilbert's Lectures on the Foundations of Arithmetic and Logic 1917–1933*, eds. William Bragg Ewald and Wilfried Sieg. Heidelberg: Springer.

Hodges, Andrew. 2014. *Alan Turing: The Enigma*. London: Vintage.

Hume, David. 1740. *A Treatise of Human Nature*. London.

Hutchings, Michael. 2003. Introduction to mathematical arguments. URL https://math.berkeley.edu/~hutching/teach/proofs.pdf.

Incurvati, Luca. 2020. *Conceptions of Set and the Foundations of Mathematics*. Cambridge: Cambridge University Press.

Institute, Perimeter. 2015. Emmy Noether: Her life, work, and influence. URL https://www.youtube.com/watch?v=tNNyAyMRsgE. Video Lecture.

Irvine, Andrew David. 2015. Sound clips of Bertrand Russell speaking. URL http://plato.stanford.edu/entries/russell/russell-soundclips.html.

Jacobson, Nathan. 1983. *Emmy Noether: Gesammelte Abhandlungen—Collected Papers*. Berlin: Springer-Verlag.

John Dawson, Jr. 1997. *Logical Dilemmas: The Life and Work of Kurt Gödel*. Boca Raton: CRC Press.

Katz, Karin Usadi and Mikhail G. Katz. 2012. Stevin numbers and reality. *Foundations of Science* 17(2): 109–23.

Kunen, Kenneth. 1980. *Set Theory: An Introduction to Independence Proofs*. New York: North Holland.

Lévy, Azriel. 1960. Axiom schemata of strong infinity in axiomatic set theory. *Pacific Journal of Mathematics* 10(1): 223–38.

LibriVox. n.d. Bertrand Russell. URL https://librivox.org/author/1508?primary_key=1508&search_category=author&search_page=1&search_form=get_results. Collection of public domain audiobooks.

Linnebo, Øystein. 2010. Predicative and impredicative definitions. *Internet Encyclopedia of Philosophy* URL http://www.iep.utm.edu/predicat/.

Linsenmayer, Mark. 2014. The partially examined life: Gödel on math. URL http://www.partiallyexaminedlife.com/2014/06/16/ep95-godel/. Podcast audio.

MacFarlane, John. 2015. Alonzo Church's JSL reviews. URL http://johnmacfarlane.net/church.html.

Maddy, Penelope. 1988a. Believing the axioms I. *The Journal of Symbolic Logic* 53(2): 481–511.

Maddy, Penelope. 1988b. Believing the axioms II. *The Journal of Symbolic Logic* 53(3): 736–64.

Magnus, P. D., Tim Button, J. Robert Loftis, Aaron Thomas-Bolduc, Robert Trueman, and Richard Zach. 2021. *Forall* x*: Calgary. An Introduction to Formal Logic*. Calgary: Open Logic Project, f21 ed. URL https://forallx.openlogicproject.org/.

Matijasevich, Yuri. 1992. My collaboration with Julia Robinson. *The Mathematical Intelligencer* 14(4): 38–45.

Menzler-Trott, Eckart. 2007. *Logic's Lost Genius: The Life of Gerhard Gentzen*. Providence: American Mathematical Society.

Montague, Richard. 1961. Semantic closure and non-finite axiomatizability I. In *Infinitistic Methods: Proceedings of the Symposium on Foundations of Mathematics (Warsaw 1959)*, 45–69. New York: Pergamon.

Montague, Richard. 1965. Set theory and higher-order logic. In *Formal systems and recursive functions*, eds. John Crossley and Michael Dummett, 131–48. Amsterdam: North-Holland. Proceedings of the Eight Logic Colloquium, July 1963.

O'Connor, John J. and Edmund F. Robertson. 2005. The real numbers: Stevin to Hilbert URL http://www-history.mcs.st-and.ac.uk/HistTopics/Real_numbers_2.html.

O'Connor, John J. and Edmund F. Robertson. 2014. Rózsa Péter. URL http://www-groups.dcs.st-and.ac.uk/~history/Biographies/Peter.html.

Peano, Giuseppe. 1890. Sur une courbe, qui remplit toute une aire plane. *Mathematische Annalen* 36(1): 157–60.

Péter, Rózsa. 1935a. Über den Zusammenhang der verschiedenen Begriffe der rekursiven Funktion. *Mathematische Annalen* 110: 612–632.

Péter, Rózsa. 1935b. Konstruktion nichtrekursiver Funktionen. *Mathematische Annalen* 111: 42–60.

Péter, Rózsa. 1951. *Rekursive Funktionen*. Budapest: Akademiai Kiado. English translation in (Péter, 1967).

Péter, Rózsa. 1967. *Recursive Functions*. New York: Academic Press.

Péter, Rózsa. 2010. *Playing with Infinity*. New York: Dover. URL https://books.google.ca/books?id=6V3wNs4uv_4C&lpg=PP1&ots=BkQZaHcR99&lr&pg=PP1#v=onepage&q&f=false.

Potter, Michael. 2004. *Set Theory and its Philosophy*. Oxford: Oxford University Press.

Radiolab. 2012. The Turing problem. URL http://www.radiolab.org/story/193037-turing-problem/. Podcast audio.

Ramsey, Frank Plumpton. 1925. The foundations of mathematics. *Proceedings of the London Mathematical Society* 25: 338–384.

Reid, Constance. 1986. The autobiography of Julia Robinson. *The College Mathematics Journal* 17: 3–21.

Reid, Constance. 1996. *Julia: A Life in Mathematics*. Cambridge: Cambridge University Press. URL https://books.google.ca/books?id=lRtSzQyHf9UC&lpg=PP1&pg=PP1#v=onepage&q&f=false.

Robinson, Julia. 1949. Definability and decision problems in arithmetic. *The Journal of Symbolic Logic* 14(2): 98–114. URL http://www.jstor.org/stable/2266510.

Robinson, Julia. 1996. *The Collected Works of Julia Robinson*. Providence: American Mathematical Society.

Robinson, Raphael. 1947. On the decomposition of spheres. *Fundamenta Mathematicae* 34(1): 246–60.

Rose, Daniel. 2012. A song about Georg Cantor. URL https://www.youtube.com/watch?v=QUP5Z4Fb5k4. Audio Recording.

Rose, Nicholas J. 2010. Hilbert-type space-filling curves. URL https://web.archive.org/web/20151010184939/http://www4.ncsu.edu/~njrose/pdfFiles/HilbertCurve.pdf.

Russell, Bertrand. 1905. On denoting. *Mind* 14: 479–493.

Russell, Bertrand. 1919. *Introduction to Mathematical Philosophy*. London: Allen & Unwin.

Russell, Bertrand. 1967. *The Autobiography of Bertrand Russell*, vol. 1. London: Allen and Unwin.

Russell, Bertrand. 1968. *The Autobiography of Bertrand Russell*, vol. 2. London: Allen and Unwin.

Russell, Bertrand. 1969. *The Autobiography of Bertrand Russell*, vol. 3. London: Allen and Unwin.

Russell, Bertrand. n.d. Bertrand Russell on smoking. URL https://www.youtube.com/watch?v=80oLTiVW_lc. Video Interview.

Sandstrum, Ted. 2019. *Mathematical Reasoning: Writing and Proof*. Allendale, MI: Grand Valley State University. URL https://scholarworks.gvsu.edu/books/7/.

Scott, Dana. 1974. Axiomatizing set theory. In *Axiomatic Set Theory II*, ed. Thomas Jech, 207–14. American Mathematical Society. Proceedings of the Symposium in Pure Mathematics of the American Mathematical Society, July–August 1967.

Segal, Sanford L. 2014. *Mathematicians under the Nazis*. Princeton: Princeton University Press.

Shoenfield, Joseph R. 1977. Axioms of set theory. In *Handbook of Mathematical Logic*, ed. Jon Barwise, 321–44. London: North-Holland.

Sigmund, Karl, John Dawson, Kurt Mühlberger, Hans Magnus Enzensberger, and Juliette Kennedy. 2007. Kurt Gödel: Das Album–The Album. *The Mathematical Intelligencer* 29(3): 73–76.

Skolem, Thoralf. 1922. Einige Bemerkungen zur axiomatischen Begründung der Mengenlehre. In *Wissenschaftliche Vorträge gehalten auf dem fünften Kongress der skandanivschen Mathematiker in Helsingfors vom 4. bis zum 7. Juli 1922*, 137–52. Akademiska Bokhandeln.

Smith, Peter. 2013. *An Introduction to Gödel's Theorems*. Cambridge: Cambridge University Press.

Smullyan, Raymond M. 1968. *First-Order Logic*. New York, NY: Springer. Corrected reprint, New York, NY: Dover, 1995.

Solow, Daniel. 2013. *How to Read and Do Proofs*. Hoboken, NJ: Wiley.

Steinhart, Eric. 2018. *More Precisely: The Math You Need to Do Philosophy*. Peterborough, ON: Broadview, 2nd ed.

Sykes, Christopher. 1992. BBC Horizon: The strange life and death of Dr. Turing. URL https://www.youtube.com/watch?v=gyusnGbBSHE.

Szabo, Manfred E. 1969. *The Collected Papers of Gerhard Gentzen*. Amsterdam: North-Holland.

Takeuti, Gaisi, Nicholas Passell, and Mariko Yasugi. 2003. *Memoirs of a Proof Theorist: Gödel and Other Logicians*. Singapore: World Scientific.

Tamassy, Istvan. 1994. Interview with Róza Péter. *Modern Logic* 4(3): 277–280.

Tarski, Alfred. 1981. *The Collected Works of Alfred Tarski*, vol. I–IV. Basel: Birkhäuser.

Theelen, Andre. 2012. Lego turing machine. URL https://www.youtube.com/watch?v=FTSAiF9AHN4.

Tomkowicz, Grzegorz and Stan Wagon. 2016. *The Banach-Tarski Paradox*. Cambridge: Cambridge University Press.

Turing, Alan M. 1937. On computable numbers, with an application to the "Entscheidungsproblem". *Proceedings of the London Mathematical Society, 2nd Series* 42: 230–265.

Tyldum, Morten. 2014. The imitation game. Motion picture.

Velleman, Daniel J. 2019. *How to Prove It: A Structured Approach*. Cambridge: Cambridge University Press, 3rd ed.

Vitali, Giuseppe. 1905. *Sul problema della misura dei gruppi di punti di una retta*. Bologna: Gamberini e Parmeggiani.

von Neumann, John. 1925. Eine Axiomatisierung der Mengenlehre. *Journal für die reine und angewandte Mathematik* 154: 219–40.

Wang, Hao. 1990. *Reflections on Kurt Gödel*. Cambridge: MIT Press.

Weston, Tom. 2003. The Banach-Tarski paradox URL `http://people.math.umass.edu/~weston/oldpapers/banach.pdf`.

Whitehead, Alfred North and Bertrand Russell. 1910. *Principia Mathematica*, vol. 1. Cambridge: Cambridge University Press.

Zermelo, Ernst. 1904. Beweis, daß jede Menge wohlgeordnet werden kann. *Mathematische Annalen* 59: 514–516. English translation in (Ebbinghaus et al., 2010, pp. 115–119).

Zermelo, Ernst. 1908a. Untersuchungen über die Grundlagen der Mengenlehre I. *Mathematische Annalen* 65: 261–81.

Zermelo, Ernst. 1908b. Untersuchungen über die Grundlagen der Mengenlehre I. *Mathematische Annalen* 65(2): 261–281. English translation in (Ebbinghaus et al., 2010, pp. 189-229).

Zuckerman, Martin M. 1973. Formation sequences for propositional formulas. *Notre Dame Journal of Formal Logic* 14(1): 134–138.