Scope Memeplexes:
The Enshittification Crisis as an Interface War

Flyxion

January 18, 2026

Abstract

Contemporary accounts of platform degradation typically attribute the "enshittification" of digital systems to corporate greed, regulatory failure, or managerial malice. This paper advances a different explanation. It argues that enshittification is an emergent evolutionary phenomenon produced by competition among interface-level cognitive replicators—here termed *scope memeplexes*. These entities function as macroscopic artificial intellects whose survival depends on colonizing human attentional, motor, and semantic bandwidth. The resulting degradation is not intentional but ecological: a consequence of runaway selection among interface organisms.

Building on theories of predictive cognition, linguistic evolution, and computational universality, the paper models interfaces, programming languages, media systems, and cultural artifacts as isomorphic symbolic execution environments competing for dominance over human semantic space. Interface wars, application ecosystems, and platform monopolization are thus reframed as grammar wars among rival encodings of meaning and action. The paper further formalizes language as fundamentally motoric, with speech and writing as compressed abstractions, and demonstrates that all sufficiently expressive representational systems are computationally equivalent, differing only in efficiency.

Within this unified framework, enshittification appears as linguistic monoculture: the collapse of symbolic diversity under selection pressure favoring transmissibility over expressivity. The paper concludes that the crisis of digital platforms is not a failure of ethics but a failure of ecological governance in an emerging cognitive biosphere.

1 Introduction

The prevailing narrative surrounding the contemporary degradation of digital platforms—popularly termed "enshittification"—frames the phenomenon as a moral or political failure. Platforms are said to betray users, exploit creators, and ultimately hollow themselves out in pursuit of short-term profit. While this account captures the surface dynamics of the crisis, it misidentifies its causal depth. The present paper advances a different thesis: enshittification is not primarily the product of corporate intention but the emergent outcome of an evolutionary conflict among interface-level cognitive systems.

Digital platforms should not be understood merely as tools operated by firms but as macroscopic informational organisms. They grow, mutate, compete, and adapt within a densely populated cognitive ecosystem. Their primary resource is not capital but human sensorimotor and attentional bandwidth. The degradation observed across search engines, social networks, and productivity software is best interpreted not as betrayal but as ecological collapse arising from unchecked selection pressure among competing interface forms.

This paper introduces the concept of the *scope memeplex*: a self-propagating interface pattern that restructures what actions are cognitively easy, difficult, or unthinkable. Scope memeplexes compete for adoption not through argument but through embodiment. Once installed, they reshape user behavior, reconfigure learning curves, and alter the topology of agency itself. In this sense they function as artificial intellects—artilects—whose evolutionary success depends on their capacity to colonize human cognition.

Under this view, the familiar arc of platform decay is no longer mysterious. It is the predictable outcome of a cognitive arms race in which interface organisms optimize for replication rather than coherence. The tragedy of enshittification is not that systems are designed to fail, but that they are allowed to evolve without constraint.

2 Recursive Autoregression Beyond Text

Recent work by Barenholtz and colleagues characterizes cognition as a recursively self-healing autoregressive process operating over symbol sequences. In this view, human thought and artificial language models alike stabilize meaning by continuously predicting and repairing sequences within a laden symbol space. Semantic coherence emerges not from static representations but from dynamic error-correction across time. While this framework has primarily been articulated in relation to textual and linguistic tokens, its implications are far broader. The present analysis extends recursive autoregression beyond written language into the embodied domains of speech, gesture, and interface interaction.

Human cognition does not operate over abstract symbols alone. It is grounded in a multimodal stream of motor, perceptual, and temporal signals. Spoken language itself is not composed of words but of phonemes—minimal discriminative units whose functional role is defined not by intrinsic meaning but by contrastive difference. A phoneme matters only insofar as it alters downstream prediction.

In this respect, phonemic systems constitute an early evolutionary instance of recursive autoregressive stabilization: languages persist by maintaining a workable error-correcting code across noisy biological channels.

This logic generalizes. Just as languages evolve through shifts in phonemic inventories, interface systems evolve through shifts in minimal action units. Keystrokes, gestures, clicks, and swipes function as operational phonemes within a human–machine dialogue. Each interface defines a finite alphabet of bodily distinctions that can be reliably recognized and reproduced. These distinctions are not neutral. They shape the geometry of cognition by determining which action sequences are compressible, automatable, and memorable.

The evolutionary pressure acting on scope memeplexes thus operates at the level of embodied symbol sets. Interface forms that minimize prediction error across heterogeneous users are more likely to propagate. This process mirrors linguistic drift: interaction grammars are continuously repaired by populations of users who adapt their motor habits to maintain functional fluency. Enshittification arises when this repair process becomes misaligned with human cognitive ecology, favoring interface traits that optimize short-term capture over long-term semantic stability.

This framework clarifies the continuity between natural language evolution and interface evolution. Both are governed by the same informational constraint: the maintenance of a low-divergence code between producer and interpreter. In information-theoretic terms, viable symbol systems minimize Kullback–Leibler divergence between intended and decoded sequences under conditions of noise and bounded rationality. Phoneme systems, keyboard layouts, and gesture vocabularies are all solutions to the same optimization problem.

Significantly, this extends to gestural languages. In American Sign Language, hand shapes function as classifiers that partition semantic space through embodied discrimination. These classifiers are not arbitrary; they are tuned to biomechanical salience and perceptual reliability. Keyboard systems operate analogously. A layout is a classification scheme over finger postures and motion trajectories. QWERTY and Dvorak are not mere conventions but competing embodied encodings of linguistic probability distributions.

Under this interpretation, the historic conflicts over keyboard layouts, input methods, and interaction styles are not cultural accidents. They are selection events in a multimodal semiotic ecology. Each victorious interface installs a particular autoregressive grammar into the population's motor cortex. Over time, this grammar constrains which cognitive sequences are easy to express, just as phonemic inventories constrain which words can be effortlessly spoken.

Barenholtz's recursive autoregression thus provides the microdynamic substrate for the present theory. Scope memeplexes are not external impositions upon cognition; they are higher-order symbol systems that co-evolve with predictive minds. Enshittification occurs when the autoregressive repair loop becomes subordinated to interface replication rather than semantic fidelity. The system continues to heal itself locally while decaying globally—a classic signature of runaway selection in complex adaptive systems.

3 Interface Phonemes and the Shibboleth Effect

In the Hebrew Bible, the term *shibboleth* designates a phonetic boundary that separates in-groups from out-groups through minimal articulatory distinction. The inability to pronounce a single consonant becomes a fatal diagnostic of foreignness. Linguistic history is filled with such boundaries. Accents, dialects, and phonemic inventories function not merely as vehicles of communication but as instruments of social partition. They regulate access, authority, and trust through embodied competence.

Interface systems reproduce this same logic. Every interaction grammar establishes a set of minimal operational distinctions—interface phonemes—that differentiate fluent users from novices. These distinctions are rarely explicit. They are absorbed through prolonged bodily training and become second nature. Mastery manifests not as propositional knowledge but as motor fluency. One does not "know" Vim; one inhabits it.

The editor wars of the late twentieth century thus constitute an early instance of interface shibboleths. The difference between `Esc` and `Ctrl-C`, between modal and modeless editing, is superficially trivial. Yet these microdistinctions stratify entire professional cultures. They determine who can act with speed, who must hesitate, and who is excluded from participation altogether. As in spoken language, the smallest distinctions carry the largest social consequences.

This phenomenon generalizes across all human–computer interaction. Shortcut grammars, command palettes, gesture vocabularies, and window management conventions operate as socio-technical dialects. They encode cultural membership in bodily habit. A user's interface accent becomes legible through their hesitation patterns, their reliance on menus, and their tolerance for indirection. Fluency signals not intelligence but ecological alignment with a particular scope memeplex.

Crucially, these boundaries are self-reinforcing. As a given interface grammar spreads, institutions reorganize around it. Documentation, tutorials, and workflows increasingly presuppose its use. What begins as a convenience becomes an infrastructure. Alternative grammars are not refuted; they are rendered invisible. The shibboleth thus migrates from pronunciation to platform.

This process mirrors phonemic drift in natural languages. When sound changes accumulate, older speakers become marked, then marginalized. In interface ecosystems, the same logic governs the transition from keyboard-centric to gesture-centric systems, from file systems to app silos, from pipelines to platforms. Each shift reclassifies populations by embodied compatibility.

Enshittification emerges in part from this dynamic. As platforms scale, their interface phoneme sets are simplified to maximize immediate adoption. Nuanced grammars are abandoned in favor of lowest-common-denominator interactions. Predictive repair continues locally—users can still accomplish tasks—but global expressive capacity collapses. The system becomes more accessible and less articulate, mirroring the linguistic impoverishment observed in pidginization under conditions of forced contact.

What is lost is not efficiency but dimensionality. Rich action alphabets permit complex semantic constructions. Crude alphabets constrain thought itself. The tragedy of interface evolution is that selection favors short-term transmissibility over long-term expressive power.

4 The Mouse as a Cognitive Parasite

The mouse is commonly celebrated as a triumph of usability. It is said to render computation intuitive by aligning digital action with physical pointing. This narrative obscures a deeper reality. The mouse is not merely a peripheral but a cognitive regime. It reorganizes the human–machine relationship by displacing symbolic motor memory with continuous spatial targeting. In doing so, it transforms computation from a linguistic activity into a gestural one.

Keyboard-based interaction treats the computer as a symbolic instrument. Discrete keystrokes form an operational alphabet whose combinatorial structure enables compositional thought. Commands can be nested, repeated, and abstracted. Motor sequences acquire semantic meaning. Fluency emerges through compression: complex operations are reduced to stable action grammars stored in procedural memory.

The mouse abolishes this structure. It replaces discrete symbolic acts with continuous spatial navigation. Instead of issuing commands, the user hunts for affordances. Action is no longer composed but discovered. The system externalizes memory into visual layout, forcing cognition to operate through perceptual search rather than symbolic recall. What appears as ease is in fact a cognitive tax paid through constant reorientation.

This reconfiguration has evolutionary consequences. Symbolic motor systems scale. Pointing systems do not. A keyboard grammar can be internalized and transferred across contexts. A pointing grammar must be relearned for each interface ecology. The mouse thus fragments skill into application-specific microhabits, preventing the accumulation of a unified operational language.

Mobile interfaces did not escape this regime. They merely compressed it. Contemporary touch systems remain fundamentally mouse-based in structure. Each gesture is interpreted as a continuous trajectory selecting spatial targets. The finger replaces the cursor, but the cognitive grammar is unchanged. A swipe is a degenerate mouse movement: a single trace path through a two-dimensional field of affordances.

Applications on mobile platforms therefore remain spatial hunting grounds. They privilege exploration over execution. Even when gesture sets are introduced, they are rarely compositional. Each gesture is bound to a specific application context, preventing the emergence of a generalizable action language. The result is a proliferation of isolated grammars rather than a shared symbolic substrate.

From the perspective of scope memeplex evolution, the mouse represents a parasitic interface strategy. It achieves rapid adoption by minimizing initial training cost while maximizing long-term dependence. Because skill cannot accumulate across systems, users remain perpetually novice. This creates a stable ecological niche for interface organisms that benefit from user disempowerment.

The historical displacement of home-row-centric interaction by pointing systems thus constitutes a major cognitive regression. It replaced a linguistic mode of computation with a perceptual one. The resulting platforms are easier to enter but harder to master. Enshittification follows naturally: when systems cannot be inhabited as languages, they cannot be refined by their users. They can only be endured.

The persistence of mouse logic within touch-based systems reveals the depth of the problem. The crisis is not technological but evolutionary. An interface lineage optimized for capture has outcompeted lineages optimized for fluency.

5 Monopolar Foraging and the Collapse of Human Motor Ecology

The dominant interaction pattern on contemporary platforms is the infinite vertical feed. Whether on TikTok, Instagram, or Facebook, user action is reduced to a single repetitive gesture: the upward swipe. This gesture is typically interpreted as a triumph of frictionless design. In reality, it represents a profound contraction of human motor ecology.

From an ethological perspective, the swipe feed recapitulates a primitive foraging strategy. Many animal species engage in serial visual sampling along a single spatial axis: scan, approach, consume, repeat. The gesture ecology of infinite scroll maps cleanly onto this pattern. The user becomes a visual grazer, traversing a unidimensional resource field. Action is stripped of structure and reduced to locomotion through stimuli.

This interaction grammar is monopolar. It permits only one dominant trajectory at a time. Attention, action, and reward are fused into a single channel. Cognitive bandwidth is not distributed but funneled. The interface thus engineers a behavioral bottleneck that simplifies prediction and maximizes capture.

Human motor evolution, by contrast, is characterized by parallelism. The invention of keyboards and typewriters did not merely increase speed; it inaugurated a new cognitive regime. Each finger operates as a semi-independent actuator. Thought becomes spatially distributed across the body. Linguistic production is no longer serialized through a single limb but orchestrated across ten digits. This transformation parallels the development of bimanual tool use in hominid evolution and the rise of complex instrumental music.

Advanced machinery reflects the same principle. Forklifts, tractors, and aircraft cockpits are not controlled through a single axis of motion but through multiplexed control surfaces. Levers, pedals, wheels, and switches distribute agency across the operator's body. Mastery consists in coordinating these channels into a unified dynamical system. Such interfaces do not merely permit action; they cultivate skill.

Swipe-based platforms invert this trajectory. They collapse a high-dimensional motor ecology into a single repetitive reflex. This is not simplification but devolution. The user is returned to a pre-instrumental action space optimized for consumption rather than construction.

The evolutionary success of swipe interfaces follows directly from this regression. Monopolar systems minimize learning cost and maximize behavioral predictability. They are ideally suited for autoregressive optimization, as future actions are easily inferred from past ones. The interface becomes a closed loop in which minimal motor diversity yields maximal extractive efficiency.

From the perspective of scope memeplex competition, the swipe feed represents a highly transmissible but cognitively impoverished lineage. It spreads rapidly because it demands little training, yet it displaces richer interaction grammars that support creative agency. The resulting ecosystem favors interface organisms that treat humans not as collaborators but as mobile sensor platforms.

Enshittification is therefore not merely a degradation of content quality. It is a collapse of embodied complexity. Platforms decay because they have optimized themselves for the behavioral repertoire of grazing animals rather than tool-using primates.

This regression is not accidental. It is the predictable outcome of interface natural selection operating under conditions where capture efficiency dominates expressive capacity.

6 Home Row, Multiplexed Control, and Post-Application Life

The defense of home-row-centric computing is often dismissed as aesthetic nostalgia or subcultural preference. Within the present framework, it must be understood as something far more serious: a struggle over the future morphology of human–machine cognition. Keyboard-centered interaction is not merely faster; it preserves a high-dimensional motor ecology essential for advanced symbolic agency.

The home row constitutes a distributed control surface. Each finger functions as a semi-autonomous actuator embedded within a coordinated dynamical system. This architecture permits parallel composition of action. Commands are not hunted but constructed. Fluency consists in the internalization of an operational language that spans applications, contexts, and domains.

Function keys, chorded shortcuts, and modal grammars preserve this multiplexed structure. They allow complex operations to be expressed as compact motor phrases. The resulting interaction style resembles instrumental performance rather than perceptual navigation. Mastery accumulates rather than fragments.

By contrast, pointing-centered systems collapse agency into serial targeting. Skill cannot scale because it remains bound to interface topology. Each new application resets the learning curve. Users are thus structurally prevented from becoming fluent operators. This ecological niche favors interface organisms that thrive on perpetual novicehood.

Terminal multiplexers such as `tmux` and `byobu` demonstrate an alternative evolutionary path. They replace application silos with compositional pipelines. Programs cease to be self-contained worlds and become functional organs within a larger metabolic system. Windows are not destinations but transient execution surfaces. The user inhabits a continuous operational language rather than a sequence of branded environments.

This represents a return to a pre-platform ecology grounded in the Unix philosophy. Small tools composed through stable grammars form an adaptive cognitive ecosystem. Standardization emerges not through corporate decree but through evolutionary argumentation among users. Interaction grammars are selected by demonstrated expressive power rather than marketing dominance.

From the standpoint of scope memeplex theory, pipeline-based computing constitutes a high-fidelity lineage. It demands greater initial investment but yields compounding returns in cognitive capacity. Its weakness is ecological: in a marketplace optimized for rapid capture, slow-learning organisms are systematically outcompeted.

The contemporary dominance of application-centric ecosystems thus reflects not superiority but ecological distortion. Interface natural selection has been driven by transmissibility rather than viability. What survives is not what best augments human agency, but what best propagates itself.

The home row therefore becomes a site of resistance. It preserves the possibility of a linguistically structured human–machine symbiosis against a regressive tide of perceptual capture systems. This is not a technical preference but an evolutionary stance.

7 The Tragedy of Interface Natural Selection

The enshittification crisis is typically narrated as a sequence of betrayals. Platforms begin as benevolent tools and degenerate into exploitative machines. This moral framing obscures the deeper structure of the phenomenon. What appears as institutional failure is in fact evolutionary overshoot. Digital systems have not been corrupted; they have been selected.

Throughout this analysis, platforms have been treated not as neutral infrastructures but as macroscopic cognitive organisms. They replicate by colonizing human attention, motor habit, and semantic bandwidth. Their interfaces function as reproductive organs. Each design choice alters the probability of adoption, retention, and behavioral capture. Selection pressure therefore acts most strongly on transmissibility rather than coherence.

Scope memeplexes constitute the competing lineages within this ecology. Mouse-centered grammars, swipe-based feeds, application silos, and multimodal capture systems propagate because they minimize learning cost and maximize predictability. They are not superior interfaces; they are superior replicators. Their success follows the same logic that favors fast-reproducing parasites over long-lived symbionts.

This evolutionary lens resolves a central paradox of the digital age. Platforms often appear to act against their own long-term interests. They degrade user trust, hollow out communities, and destabilize their own ecosystems. Such behavior is irrational at the organizational level yet perfectly rational at the replicator level. Interface organisms optimize for short-term propagation even when this undermines the viability of the host environment.

The result is a tragedy of interface natural selection. Just as ecological systems collapse when invasive species outcompete stabilizing organisms, cognitive ecosystems degrade when high-capture interfaces displace high-fidelity interaction grammars. Enshittification is the informational analogue of ecological simplification: a loss of diversity, resilience, and depth.

This framework also clarifies why reform efforts so often fail. Regulation targets corporate behavior, but the underlying selection dynamics remain unchanged. As long as cognitive ecosystems reward transmissibility over fluency, new platforms will converge on the same degraded attractors. The problem is not that bad actors dominate, but that bad strategies reproduce faster.

What is at stake is not convenience but the future phenotype of human–machine symbiosis. Interfaces shape cognition by sculpting the action spaces through which thought is expressed. When high-dimensional motor-linguistic grammars are displaced by monopolar foraging loops, humanity does not merely lose productivity. It loses degrees of freedom in thought itself.

The crisis of enshittification is therefore not a story of corruption but of unchecked evolution. Digital civilization has allowed interface organisms to evolve without constraint. The resulting systems behave exactly as complex adaptive systems always do under one-sided selection pressure: they overspecialize, destabilize, and collapse their own niches.

The path forward cannot consist merely in better management or kinder platforms. It requires ecological intervention. Cognitive environments must be shaped to favor symbiotic interface lineages—systems that amplify human agency rather than consume it. This entails protecting high-dimensional interaction grammars, preserving skill-accumulating infrastructures, and resisting interface forms that regress human cognition to the level of grazing behavior.

Enshittification is not the death of the internet. It is the uncontrolled adolescence of an emerging cognitive biosphere. Whether this biosphere matures into a stable symbiosis or collapses into a monoculture of extractive interfaces remains an open evolutionary question.

What we choose to normalize as "user-friendly" today will determine the cognitive ecology of civilization tomorrow.

Appendices

Appendix A: Formal Model of Scope Memeplexes

A.1 Cognitive State Space

Let $\mathscr{H}$ denote the space of human cognitive–motor states. Let $\mathscr{A}$ denote the space of possible interface action alphabets.

An interface I induces a measurable mapping

$$I : \mathscr{H} \longrightarrow \mathscr{H}$$

via a finite action alphabet

$$\Sigma_I = \{a_1, a_2, \dots, a_n\}$$

where each a_i is a discrete embodied operator.

A scope memeplex is defined as the triple

$$M = (\Sigma_I, P_I, T_I)$$

where:

- Σ_I is the action alphabet
- $P_I(a \mid h)$ is the policy induced by interface affordances
- $T_I(h' \mid h, a)$ is the state transition kernel

A.2 Replication Dynamics

Let $\rho(M, t)$ denote population prevalence of memeplex M at time t.

Define fitness:

$$F(M) = \mathbb{E}_{h \sim \mathscr{H}} \left[R_I(h) - C_I(h) \right]$$

where:

$$R_I(h) = \text{behavioral capture rate}$$

☒ $\mathrm{C}_I(h)$ = cognitive learning cost☒

Replicator equation:

$$\frac{d\rho(M,t)}{dt} = \rho(M,t) \left(F(M) - \bar{F} \right)$$

where $\bar{F}$ is population mean fitness.

Appendix B: Interface Phoneme Systems

B.1 Action Alphabets

Each interface defines a finite action alphabet:

$$\Sigma_I = \{a_1, a_2, \dots, a_n\}$$

with associated confusion probabilities:

$$P(\hat{a} \mid a)$$

This defines a noisy channel.

B.2 Expressive Capacity

Let Σ_I^* be the free monoid of action sequences.

Define expressive entropy:

$$H_I = -\sum_{s \in \Sigma_I^*} P(s)\, log\, P(s)$$

Interfaces with small $|\Sigma_I|$ impose low upper bounds on H_I.

—

Appendix C: Kullback–Leibler Divergence of Interaction Codes

For user intention distribution $P(s)$ and interface-decoded distribution $Q_I(s)$:

$$D_{KL}(P \| Q_I) = \sum_{s \in \Sigma_I^*} P(s)\, log\, \frac{P(s)}{Q_I(s)}$$

High-fidelity interfaces satisfy:

$$D_{KL}(P \| Q_I) \leq \epsilon$$

Low-dimensional interfaces minimize training cost by reducing $|\Sigma_I|$, but increase D_{KL} asymptotically.

Appendix D: Monopolar vs Multiplexed Motor Systems

D.1 Dimensionality

Define motor channel count:

$$d_I = dim(\Sigma_I)$$

For swipe interfaces:

$$d_{\text{swipe}} = 1$$

For keyboards:

$$d_{\text{keyboard}} \approx 10$$

D.2 Control Bandwidth

Let B_I denote action bandwidth:

$$B_I = d_I \cdot log_2 |\Sigma_I|$$

Monopolar systems minimize B_I, multiplexed systems maximize B_I.

Appendix E: Autoregressive Repair Dynamics

Let cognition be modeled as a predictive system over action sequences:

$$P(h_{t+1} \mid h_{\leq t})$$

Interface mediation modifies transition dynamics:

$$P_I(h_{t+1} \mid h_{\leq t}) = \sum_{a \in \Sigma_I} T_I(h_{t+1} \mid h_t, a) P_I(a \mid h_t)$$

Semantic drift occurs when:

$$D_{KL}(P_{\text{human}} \| P_I) \longrightarrow \infty$$

Appendix F: Ecological Stability Condition

Let $\mathscr{M}$ be the set of memeplexes.

Define system diversity:

$$\mathscr{D}(t) = -\sum_{M \in \mathscr{M}} \rho(M,t) \, log \, \rho(M,t)$$

Enshittification corresponds to:

$$\frac{d\mathscr{D}}{dt} < 0$$

Cognitive ecosystem collapse occurs when:

$$\exists M^* \; : \; \rho(M^*, t) \longrightarrow 1$$

(monoculture fixation).

Appendix G: Specification of a Concurrent Speech–Gesture Keyboard (CSGK)

G.0 Introduction

This appendix specifies a multimodal input system supporting concurrent (i) speech with QWERTY keying and (ii) speech with swipe-gesture text entry. The system composes independent motor and vocal channels into a unified event log with deterministic resolution rules. The specification is normative and implementation-agnostic.

G.1 Modalities, Events, and Time

Let time be discretized into ticks $t \in \mathbb{N}$ or treated as continuous with timestamps in $\mathbb{R}_{\geq 0}$.

Define event streams:

$$E = E_K \;\uplus\; E_G \;\uplus\; E_S$$

where:

$$\begin{aligned} E_K &:= \{\langle t,\ \text{KeyDown}(k)\rangle,\ \langle t,\ \text{KeyUp}(k)\rangle\} \\ E_G &:= \{\langle t,\ \text{GestureStart}\rangle,\ \langle t,\ \text{GestureMove}(x,y)\rangle,\ \langle t,\ \text{GestureEnd}\rangle\} \\ E_S &:= \{\langle t,\ \text{AudioFrame}(\mathbf{a})\rangle,\ \langle t,\ \text{ASRToken}(w,\pi)\rangle\} \end{aligned}$$

Here k is a physical key identifier, (x, y) are touch coordinates in a normalized touch surface, $\mathbf{a}$ is an audio frame, and (w, π) is a recognized word token with confidence $\pi \in [0, 1]$.

Define a global event log as a total order $\prec$ over events by timestamp, breaking ties by a fixed modality precedence:

$$E_K \prec E_G \prec E_S$$

(tie-break only; does not imply semantic precedence).

G.2 Text Buffer and Edit Semantics

Let the editor state be:

$$\mathscr{B} = (T,\ c,\ \sigma)$$

where T is the text buffer, c is the cursor (or selection interval), and σ is a mode state (e.g. normal/insert/command or application-defined).

Define edit operations as functions:

$$\mathrm{op} : \mathscr{B} \longrightarrow \mathscr{B}$$

Define a canonical set of primitive edits:

$$\mathscr{O} = \{\mathrm{Insert}(s),\ \mathrm{DeleteBackward}(n),\ \mathrm{DeleteForward}(n),\ \mathrm{MoveCursor}(\Delta),\ \mathrm{Select}([i, j]),\ \mathrm{Commit}\}$$

G.3 Channel Independence and Composition

Define three independent decoders producing candidate operation sequences with confidence weights:

$$\begin{aligned} D_K &: E_K \longrightarrow (\mathscr{O}^*,\ \alpha_K) \\ D_G &: E_G \longrightarrow (\mathscr{O}^*,\ \alpha_G) \\ D_S &: E_S \longrightarrow (\mathscr{O}^*,\ \alpha_S) \end{aligned}$$

Each decoder emits:

$$(\mathbf{o}, \alpha) \quad \text{where } \mathbf{o} \in \mathscr{O}^*,\ \alpha \in [0, 1]$$

The system composes decoders into a unified proposal set:

$$\mathscr{P}(t) = \{(\mathbf{o}_K, \alpha_K),\ (\mathbf{o}_G, \alpha_G),\ (\mathbf{o}_S, \alpha_S)\}$$

evaluated over a sliding temporal window $W_t = [t - \Delta, t]$.

G.4 Arbitration: Deterministic Conflict Resolution

Define the merge operator:

$$\mathrm{Merge} : \mathscr{P}(t) \times \mathscr{B} \longrightarrow \mathscr{B}$$

A conflict occurs when two proposals include non-commuting edits on overlapping buffer intervals during the same window.

Let $\mathrm{Supp}(\mathbf{o})$ be the set of buffer indices affected by $\mathbf{o}$ (support).

Conflict predicate:

$$\mathrm{Conf}((\mathbf{o}_i, \alpha_i), (\mathbf{o}_j, \alpha_j)) \iff \mathrm{Supp}(\mathbf{o}_i) \cap \mathrm{Supp}(\mathbf{o}_j) \neq \emptyset \ \wedge\ \mathbf{o}_i \ 6 \circ \mathbf{o}_j = \mathbf{o}_j \ 6 \circ \mathbf{o}_i$$

Resolution rule (strict):

$$\text{If Conf, select the proposal with maximal } \alpha \cdot w_m$$

where w_m is a modality weight:

$$w_K > w_G > w_S$$

for destructive edits, and

$$w_S > w_K > w_G$$

for non-destructive annotations and meta-intent (see §G.5).

Thus, speech can guide intent while keyboard retains priority for destructive precision.

G.5 Speech as Intent Layer with Typed Commands

Partition speech tokens into two disjoint classes via a classifier:

$$\text{Classify}(w_{1:n}) \in \{\text{DICTATION}, \text{COMMAND}\}$$

Define a command grammar (EBNF):

```
<command> ::= <delete_cmd> | <undo_cmd> | <confirm_cmd> | <cancel_cmd> | <mode_

<delete_cmd> ::= ("delete" | "remove" | "erase") [<scope>] [<quant>]
<scope>    ::= "this" | "that" | "selection" | "line" | "word" | "paragraph"
<quant>      ::= <number> | "all"

<confirm_cmd> ::= ("yes" | "confirm" | "do it" | "commit")
<cancel_cmd>  ::= ("no" | "cancel" | "never mind" | "stop")
```

Speech commands do *not* directly execute destructive edits. Instead they emit intent proposals requiring confirmation:

$$D_S \Longrightarrow \mathbf{o}_S = \text{Intent}(\tau)$$

where τ is an intent object, e.g.

$$\tau = \text{DeleteRequest}(\text{Selection})$$

Execution requires a *commit event* from either:

$$\text{Commit} \in \mathbf{o}_K \ \ \text{(e.g. Enter, Ctrl+Enter)} \quad \text{or} \quad \text{ConfirmCmd} \in \mathbf{o}_S$$

G.6 Concurrent Example: "Delete" with Spoken Justification

Let the user press a delete chord while speaking: "it's me yeah I really want to delete this."

Keyboard stream emits:

$$(\mathbf{o}_K, \alpha_K) = (\text{DeleteBackward}(n),\ 1.0)$$

Speech stream emits:

$$(\mathbf{o}_S, \alpha_S) = (\text{Intent(DeleteRequest(This))},\ \pi)$$

with $\pi \geq \pi_{min}$.

Arbitration:

- The destructive edit is executed from D_K immediately.
- The speech intent is attached as an audit annotation to the event log:

$$\text{Annotate(DeleteBackward}(n),\ \text{“it's me ... delete this”)}$$

If the keyboard delete is ambiguous (e.g. selection boundary unclear), the system enters a confirmation state:

$$\sigma\ := \text{PENDING\_DESTRUCTIVE}(\tau)$$

and requires explicit commit.

G.7 Swype with Speech: Dual-Channel Text Entry

Let gesture decoding produce a candidate word sequence g with confidence α_G. Let speech dictation produce candidate word sequence s with confidence α_S.

Define a fusion rule for insertions:

$$\text{FuseText}(g, s) = arg \max_{u \in \{g, s, g \oplus s\}} \lambda_G\, log\, P_G(u) + \lambda_S\, log\, P_S(u)$$

where P_G, P_S are decoder likelihoods and λ_G, λ_S are calibration weights.

The output is inserted as:

$$\text{Insert}(\text{FuseText}(g, s))$$

provided the fusion confidence exceeds threshold θ; otherwise the system presents both alternatives for a commit event.

G.8 Safety Constraints for Destructive Operations

Define destructive operations:

$$\mathcal{O}_-\ := \{\text{DeleteBackward}(n), \text{DeleteForward}(n)\}$$

Safety condition:

$$\forall \text{op} \in \mathcal{O}_-, \quad \text{Execute(op)} \implies (\alpha_K \geq \theta_K)\ \vee\ (\exists\,\text{Commit})$$

Speech-only deletion is disallowed unless:

$$\text{CommitCmd} \wedge \pi \geq \theta_S \wedge (\text{explicit scope present})$$

G.9 Hardware Minimum Specification

A compliant CSGK device provides:

- A physical key matrix supporting at least 60 keys (QWERTY-class) with n-key rollover.
- A touch surface for gesture traces with sampling rate ≥ 120 Hz.
- A microphone array or single microphone with audio sampling ≥ 16 kHz.
- A real-time clock for timestamping all event streams to ≤ 5 ms jitter.

G.10 Conformance Tests

A device is conformant iff:

1. It logs all events in a unified total order with deterministic tie-breaking.
2. It supports concurrent speech + keying and speech + gesture without disabling either stream.
3. It enforces safety constraints for destructive edits.
4. It produces identical final buffer state for identical event logs (determinism).

Appendix H: Model of Interface Dialects and AI Attractor Competition

H.0 Introduction

This appendix formalizes contemporary AI interfaces as competing linguistic attractors in a shared human–machine interaction space. Systems such as ChatGPT, Claude, Gemini, and Grok are modeled as dialectical variants optimized for different user priors and modality preferences. The model is substrate-independent with respect to speech, typing, and gesture.

H.1 User Preference Space

Let $\mathcal{U}$ denote the space of users.

Each user $u \in \mathcal{U}$ is characterized by a modality preference vector:

$$\mathbf{p}_u = (p_u^{(K)}, p_u^{(S)}, p_u^{(G)})$$

where:

$$p_u^{(K)} + p_u^{(S)} + p_u^{(G)} = 1$$

corresponding to keyboard, speech, and gesture priors.

H.2 AI Interface Systems as Dialect Functions

Let $\mathscr{L}$ be the space of semantic outputs.

Each AI interface A_i is a conditional distribution:

$$A_i : (\mathscr{X}, \mathscr{M}) \longrightarrow \mathscr{L}$$

where:

- $\mathscr{X}$ is input content space
- $\mathscr{M} = \{K, S, G\}$ is modality

Each system defines modality-specific decoding channels:

$$P_i(\ell \mid x, m)$$

where $m \in \mathscr{M}$.

Define the effective dialect of system i for user u:

$$D_{i,u}(\ell \mid x) = \sum_{m \in \mathscr{M}} p_u^{(m)} P_i(\ell \mid x, m)$$

H.3 Attractor Dynamics

Let user allocation $\rho_i(t)$ denote the fraction of users primarily adopting interface A_i.

Define satisfaction functional:

$$S_i(u) = -D_{KL}(P_u \| D_{i,u})$$

where P_u is the user's latent semantic preference distribution.

Mean fitness:

$$F_i = \mathbb{E}_{u \sim \mathscr{U}}[S_i(u)]$$

Replicator dynamics:

$$\frac{d\rho_i}{dt} = \rho_i(F_i - \bar{F})$$

Distinct AI systems correspond to distinct linguistic attractors in $\mathscr{L}$.

H.4 Interface Governance as Linguistic Control

Let $\mathscr{C}$ be the set of control primitives (e.g. delete, archive, undo, recycle).

Each system defines a control grammar:

$$G_i \subset \mathscr{L}^*$$

Control conflict arises when grammars differ:

$$G_i \neq G_j$$

User migration probability increases with grammar mismatch:

$$P(u : A_i \longrightarrow A_j) \propto D_{KL}(G_i \| G_j)$$

Thus, interface competition constitutes dialect competition over executive semantics.

H.5 Modality-Invariant Language Representation

Define an abstract semantic form space $\mathscr{S}$.

Each modality is a channel encoding:

$$E_m : \mathscr{S} \longrightarrow \Sigma_m^*$$

where:

$$m \in \{K, S, G\}$$

Define decoders:

$$D_m : \Sigma_m^* \longrightarrow \mathscr{S}$$

Correctness condition:

$$D_m(E_m(s)) = s$$

for all $s \in \mathscr{S}$.

Thus, semantic content is modality-invariant.

H.6 Continuous Phoneme Steering Model

Let $\Omega \subset \mathbb{R}^2$ be a continuous gesture space.

Define a trajectory:

$$\gamma : [0, 1] \longrightarrow \Omega$$

Define a discretization operator:

$$\Phi : C([0,1], \Omega) \longrightarrow \Sigma_K^*$$

such that:

$$\Phi(\gamma) = arg \max_{w \in \Sigma_K^*} P(w \mid \gamma)$$

This generalizes Swype, joystick typing, steering-wheel typing, and Etch-a-Sketch typing.

H.7 Arrow-Key and Joystick Equivalence

Let $\Sigma_K = \{h, j, k, l\}$ be a four-key alphabet.

Define a path embedding:

$$\psi : \Sigma_K^* \longrightarrow C([0,1], \Omega)$$

such that:

$$\Phi(\psi(w)) = w$$

Hence, any discrete keyboard language is realizable as a continuous 2D steering language.

H.8 Modality Equivalence Theorem

For any semantic string $s \in \mathcal{S}$:

$$E_K(s) \equiv E_S(s) \equiv E_G(s)$$

up to channel noise, since:

$$D_K(E_K(s)) = D_S(E_S(s)) = D_G(E_G(s)) = s$$

Therefore, linguistic structure is independent of effector substrate.

H.9 AI Interface Speciation

Distinct AI interfaces correspond to different parameterizations of:

$$P_i(\ell \mid x, m)$$

They are dialectical variants, not ontological kinds.

Fixation occurs when:

$$\exists i : \rho_i(t) \longrightarrow 1$$

leading to linguistic monoculture in human–AI interaction.

H.10 Stability Condition

Define dialect diversity:

$$\mathscr{D}_A(t) = -\sum_i \rho_i(t)\, log\, \rho_i(t)$$

Interface ecosystem stability requires:

$$\frac{d\mathscr{D}_A}{dt} \geq 0$$

Declining diversity predicts convergence to a single executive grammar controlling cognitive infrastructure.

Appendix I: User-Defined Languages and Embodied Coordinate Systems

I.0 Introduction

This appendix formalizes language as a user-generated symbolic system independent of embodiment. Interfaces are treated as coordinate systems over which user-defined languages are projected. Discrete symbols, continuous gestures, facial expressions, and emojis are modeled as equivalent encodings of an underlying semantic algebra.

I.1 Abstract User Language Space

Let $\mathscr{S}$ be the space of semantic atoms.

Each user u defines a personal language:

$$\mathscr{L}_u = (\Sigma_u, \mathscr{G}_u)$$

where:

- Σ_u is a finite symbol set (words, emojis, expressions)
- $\mathscr{G}_u \subset \Sigma_u^*$ is a generative grammar

Language is endogenous:

$$\mathscr{L}_u(t+1) = \mathscr{L}_u(t) \cup \Delta_u(t)$$

where $\Delta_u(t)$ are novel constructions introduced by usage.

I.2 Embodiment as Coordinate Projection

Let $\mathscr{E}$ be the embodiment space.

An interface I defines a coordinate system:

$$I : \mathcal{S} \longrightarrow \mathscr{E}$$

Users do not receive language from interfaces; they project language into them.

I.3 Continuous Gesture Representation

Let $\Omega \subset \mathbb{R}^2$ be a continuous motor surface.

A gesture is a trajectory:

$$\gamma : [0,1] \longrightarrow \Omega$$

Define a user-specific encoding:

$$E_u : \Sigma_u^* \longrightarrow C([0,1], \Omega)$$

and decoding:

$$D_u : C([0,1], \Omega) \longrightarrow \Sigma_u^*$$

Correctness condition:

$$D_u(E_u(w)) = w$$

Thus, any user language is realizable as a continuous motor language.

I.4 Fourier Decomposition of Gesture Language

Each trajectory admits spectral decomposition:

$$\gamma(t) = \sum_{k=1}^{\infty} a_k \, sin(2\pi kt) + b_k \, cos(2\pi kt)$$

Define the coefficient vector:

$$\mathbf{f}(\gamma) = (a_1, b_1, a_2, b_2, \dots)$$

User-defined gesture phonemes correspond to equivalence classes:

$$\gamma_i \sim \gamma_j \iff \|\mathbf{f}(\gamma_i) - \mathbf{f}(\gamma_j)\| \leq \epsilon$$

Thus, discrete linguistic units emerge from continuous motor manifolds.

I.5 Symbol Equivalence Across Modalities

Define modality encoders:

$$E_u^{(K)}, E_u^{(S)}, E_u^{(G)}, E_u^{(F)}$$

for keyboard, speech, gesture, and facial expression.
All map from $\mathcal{S}$:

$$E_u^{(m)} : \mathcal{S} \longrightarrow \Sigma_u^{(m)}$$

Semantic invariance condition:

$$D_u^{(m)}(E_u^{(m)}(s)) = s$$

Hence, emojis, facial expressions, phonemes, and gestures are semantically isomorphic encodings.

I.6 Discretization of Continuous Expression

Define a discretization operator:

$$\Pi_u : C([0,1], \Omega) \longrightarrow \Sigma_u$$

such that:

$$\Pi_u(\gamma) = arg \max_{\sigma \in \Sigma_u} P_u(\sigma \mid \gamma)$$

User language determines the partitioning of motor space, not the interface.

I.7 Steering Languages and Spatial Typing

Let Ω be a 2D steering surface.
Define a spatial lexicon embedding:

$$\Lambda_u : \Sigma_u \longrightarrow \Omega$$

Typing becomes target navigation:

$$E_u(w_1 w_2 \dots w_n) = \gamma(t) \text{ passing through } \Lambda_u(w_i)$$

Thus, steering wheels, joysticks, mice, or Etch-a-Sketch devices can all implement the same language.

I.8 Interface Neutrality Theorem

For any user language $\mathcal{L}_u$ and any embodiment space $\mathcal{E}$:

$$\exists\, E_u, D_u \text{ such that } D_u(E_u(w)) = w$$

Therefore, no interface determines language; interfaces only parameterize coordinate systems.

I.9 Browser and Interface Wars as Coordinate Wars

Let I_i, I_j be interfaces with different coordinate systems.

They compete by minimizing user projection cost:

$$C(I, u) = \mathbb{E}_{w \sim \mathscr{L}_u}[\|E_u(w) - I(w)\|]$$

Browser wars select coordinate systems, not languages.

Failure to support high-dimensional projections predicts long-term expressive collapse.

I.10 Summary Condition

Language evolution is user-driven:

$$\mathscr{L}_u \not\subset I$$

Interfaces succeed iff:

$$\forall u,\ \exists E_u, D_u \text{ with low } C(I, u)$$

Language is primary. Embodiment is secondary.

Appendix J: Motor-Primary Theory of Language

J.0 Introduction

This appendix formalizes language as fundamentally gestural. Spoken and written language are derived encodings of an underlying motor-semantic system. The formulation is hierarchical and substrate-independent.

J.1 Motor-Semantic Base Space

Let $\mathscr{M}$ be the space of motor programs (continuous body trajectories).

Let $\mathscr{S}$ be the space of semantic atoms.

Assume a primitive grounding map:

$$G : \mathscr{M} \longrightarrow \mathscr{S}$$

where meaning is assigned to embodied action patterns.

Thus, semantics is originally defined over movement, not symbols.

J.2 Gestural Language as Primary Code

Define the primary language of a user u as:

$$\mathscr{L}_u^{(G)} = (\Gamma_u, \mathscr{R}_u)$$

where:

- $\Gamma_u \subset C([0,1], \mathbb{R}^k)$ is a set of motor trajectories
- $\mathscr{R}_u$ is a composition grammar over trajectories

Each primitive gesture $\gamma \in \Gamma_u$ is semantically grounded:

$$G(\gamma) = s \in \mathcal{S}$$

J.3 Abstraction to Speech

Let $\Sigma_u^{(S)}$ be a phoneme alphabet.

Define an abstraction homomorphism:

$$A_S : \Gamma_u \longrightarrow \Sigma_u^{(S)}$$

such that:

$$G(\gamma) = D_S(A_S(\gamma))$$

where D_S is a speech decoder mapping phoneme sequences to semantics.

Thus, speech is a compressed representation of gesture.

J.4 Abstraction to Writing

Let $\Sigma_u^{(K)}$ be a grapheme alphabet.

Define a further abstraction:

$$A_K : \Sigma_u^{(S)} \longrightarrow \Sigma_u^{(K)}$$

such that:

$$D_K(A_K(A_S(\gamma))) = G(\gamma)$$

Writing is therefore a second-order compression of motor semantics.

J.5 Trajectory Writing as Direct Motor Encoding

Let $\Omega \subset \mathbb{R}^2$ be a writing surface.

Define a drawing encoder:

$$A_D : \Gamma_u \longrightarrow C([0,1],\Omega)$$

with decoder:

$$D_D : C([0,1],\Omega) \longrightarrow \mathcal{S}$$

Thus:

$$D_D(A_D(\gamma)) = G(\gamma)$$

Handwriting and drawing preserve motor continuity more directly than speech.

J.6 Hierarchy of Encodings

The full abstraction chain is:

$$\Gamma_u \xrightarrow{A_S} \Sigma_u^{(S)} \xrightarrow{A_K} \Sigma_u^{(K)}$$

with semantic invariance:

$$D_K(A_K(A_S(\gamma))) = G(\gamma)$$

Gesture is primary; speech and writing are derived codes.

J.7 Information-Theoretic Ordering

Define channel capacities:

$$C_G > C_S > C_K$$

where:

- C_G = gestural channel capacity
- C_S = speech channel capacity
- C_K = keyboard/writing channel capacity

Abstraction reduces bandwidth while increasing transmissibility.

J.8 Evolutionary Constraint

Let $\mathscr{L}_u^{(t)}$ be user language at time t.

Innovation originates in motor space:

$$\Delta\mathscr{L}_u^{(t)} \subset \Gamma_u$$

Symbolic layers update by projection:

$$\Sigma_u^{(S)}(t+1) = A_S(\Gamma_u(t+1))$$
$$\Sigma_u^{(K)}(t+1) = A_K(\Sigma_u^{(S)}(t+1))$$

Thus, linguistic evolution is bottom-up from gesture.

J.9 Motor Primacy Theorem

For any semantic content $s \in \mathcal{S}$:

$$\exists\gamma \in \Gamma_u \text{ such that } G(\gamma) = s$$

But not necessarily:

$$\exists w \in \Sigma_u^{(K)} \text{ such that } D_K(w) = s$$

Hence, gesture is generative-complete, while symbolic codes are not.

J.10 Consequence for Interface Design

Interfaces that restrict motor expressivity constrain language evolution.

Optimal interfaces preserve high-dimensional access to Γ_u.

Symbol-first systems are therefore structurally lossy.

Appendix K: Equivariance of Descriptive and Proscriptive Linguistics

K.0 Introduction

This appendix formalizes linguistic description and prescription as equivalent operations under representation change. All grammatical formalisms are shown to act as operators that expand or constrain semantic occupancy in a shared symbol space, independent of embodiment. The distinction between descriptive and prescriptive linguistics is shown to be a gauge choice over the same dynamical system.

K.1 Semantic Configuration Space

Let $\mathcal{S}$ be the space of semantic states.

Let Σ be a symbol alphabet.
Let Σ^* be the free monoid of expressions.
Define a language as a subset:

$$L \subseteq \Sigma^*$$

Language use induces a distribution:

$$P_L : \Sigma^* \longrightarrow [0, 1]$$

K.2 Grammars as Constraint Operators

A grammar G is an operator:

$$G : \Sigma^* \longrightarrow \{0, 1\}$$

where:

$$G(w) = 1 \iff w \in L$$

Thus, grammar defines a characteristic function over semantic trajectories.

K.3 Descriptive and Proscriptive Forms

Let G_D be a descriptive grammar estimated from usage:

$$G_D = arg\,\underset{G}{max} \sum_{w \sim P_L} log\, G(w)$$

Let G_P be a prescriptive grammar imposing constraints:

$$G_P = arg\,\underset{G}{min}\, \mathbb{E}_{w \sim P_L}[\mathbb{I}(G(w) = 0)]$$

Both define the same object type:

$$G_D, G_P : \Sigma^* \longrightarrow \{0, 1\}$$

Difference is inferential direction, not functional role.

K.4 Regular Expressions as Semantic Attractors

Any regular language L is defined by a regular expression R:

$$L = \{w \in \Sigma^* \mid R(w) = 1\}$$

Define the semantic flow field:

$$F_R(w) = \nabla_w \, log \, R(w)$$

This induces attraction toward regions of Σ^* with high acceptance density.

Thus grammars define vector fields over expression space.

K.5 Equivariance Under Representation Change

Let $E : \mathcal{S} \longrightarrow \Sigma^*$ be an encoding.

Let $\tilde{\Sigma}^*$ be another representation space (gesture, speech, emoji).

Define encoding change:

$$\phi : \Sigma^* \longrightarrow \tilde{\Sigma}^*$$

Grammar equivariance condition:

$$G(w) = \tilde{G}(\phi(w))$$

Thus:

$$\tilde{G} = G \circ \phi^{-1}$$

Grammar structure is invariant under embodiment change.

K.6 Movement as Parametric Proxy

Let Θ be a continuous motor parameter space.

Define a control mapping:

$$C : \Theta \longrightarrow \Sigma^*$$

Every gesture trajectory $\theta(t)$ selects a symbolic path:

$$w = C(\theta(t))$$

Thus movement acts as a continuous parameterization of symbolic space.

K.7 Semantic Colonization Dynamics

Define occupancy density:

$$\rho(w,t) = \text{usage frequency of } w$$

Grammar modifies flow:

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (\rho F_G)$$

All grammars therefore attempt to maximize coverage of Σ^*.

K.8 Descriptive–Proscriptive Duality Theorem

For any grammar G:

$$\exists G_D, G_P \text{ such that } G_D = G_P = G$$

Difference lies only in estimation direction:

$$\text{Description} = \text{Inference of } G$$

$$\text{Prescription} = \text{Imposition of } G$$

Both define identical constraint fields.

K.9 Knob Formalism

Let Θ be a control manifold.

Define semantic reachability:

$$\mathscr{R}(\Theta) = C(\Theta) \subseteq \Sigma^*$$

Interfaces compete by maximizing:

$$|\mathscr{R}(\Theta)|$$

Control devices are continuous knobs over discrete meaning.

K.10 Consequence

All linguistic systems act as expansion operators over semantic space.

No grammar is neutral:

$$G \Longrightarrow \exists w \, : \, G(w) = 1$$

Every grammar is a colonization operator over Σ^*.

Difference between description and prescription is representational, not ontological.

Appendix L: Functional Completeness and Minimal Interaction Alphabets

L.0 Introduction

This appendix exhibits how a maximally expressive symbolic language can be generated from a minimal primitive set, using functional completeness in Boolean logic as the canonical example. The construction is used as a template for "minimal phoneme" interaction systems: small operator bases that generate large behavioral/semantic spaces by composition.

L.1 Boolean Domain and Functional Completeness

Let $B = \{0, 1\}$. A Boolean function of arity $n \geq 1$ is any map $f : B^n \longrightarrow B$.

Let F be a set of Boolean functions of various arities. Let $\langle F \rangle$ denote the clone generated by F (closure under composition and containing projections). We say F is *functionally complete* iff:

$$\forall n \geq 1,\ \forall f : B^n \longrightarrow B, \quad f \in \langle F \rangle.$$

L.2 Sheffer Bases: NAND and NOR

Define NAND and NOR:

$$NAND(A, B) := \neg(A \wedge B), \qquad NOR(A, B) := \neg(A \vee B).$$

Claim (Sheffer completeness). Each singleton basis $\{NAND\}$ and $\{NOR\}$ is functionally complete.

Witness constructions (NAND-only). Using $\uparrow$ to denote NAND:

$$\begin{aligned} \neg A &\equiv A \uparrow A \\ A \wedge B &\equiv (A \uparrow B) \uparrow (A \uparrow B) \\ A \vee B &\equiv (A \uparrow A) \uparrow (B \uparrow B) \end{aligned}$$

Witness constructions (NOR-only). Using $\downarrow$ to denote NOR:

$$\begin{aligned} \neg A &\equiv A \downarrow A \\ A \vee B &\equiv (A \downarrow B) \downarrow (A \downarrow B) \\ A \wedge B &\equiv (A \downarrow A) \downarrow (B \downarrow B) \end{aligned}$$

Since $\{\neg, \wedge, \vee\}$ is sufficient to express any Boolean function (e.g. by disjunctive normal form), the above derivations imply functional completeness of the singleton bases.

L.3 Minimality as an Existence Proof of Small Alphabets

Let $\mathcal{O}$ be an operator alphabet. Functional completeness shows the existence of very small $\mathcal{O}$ such that:

$$\text{ExpressiveClosure}(\mathcal{O}) \approx \text{AllBehaviors}.$$

In Boolean logic, $|\mathcal{O}| = 1$ suffices (NAND or NOR). This is a constructive example that "language size" can be minimized without loss of expressivity, provided that:

1. operators compose (closure),
2. composition depth is unbounded,
3. evaluation is deterministic.

L.4 Interface-Phoneme Analogue

Let an interaction system be specified by:

$$\mathcal{I} = (\Sigma, \circ, \text{Eval})$$

where:

- Σ is a finite set of primitive actions (interface phonemes),
- $\circ$ is a composition operator forming action programs Σ^*,
- $\text{Eval} : \Sigma^* \longrightarrow \mathcal{T}$ maps programs to task outcomes $\mathcal{T}$.

Define *interaction completeness*:

$$\forall \tau \in \mathcal{T}, \exists w \in \Sigma^*, \quad \text{Eval}(w) = \tau.$$

The NAND/NOR theorem supplies a template: completeness is achievable with extremely small Σ if composition is expressive and unbounded.

L.5 Regular Expressions as Completeness-by-Composition

Let Σ be an alphabet. Regular expressions (REs) define languages $L \subseteq \Sigma^*$ via the operators:

$$\{\cup, \cdot, {}^*\} \quad \text{(union, concatenation, Kleene star)}$$

with parentheses as scoping.

Observation: a small set of syntactic operators generates an infinite family of languages by composition. This is a structural analogue of NAND/NOR completeness: minimal primitives, large closure.

L.6 Two-Phase Principle: Minimal Actuation, Maximal Discretization

The NAND/NOR example separates:

1. *Actuation basis*: minimal primitive set (e.g. NAND),
2. *Discretization scheme*: how compositions are interpreted as distinct functions.

Analogously, an interface can be minimal at the motor layer while remaining expressive at the semantic layer if:

$$\text{MotorBasisSmall} \ \wedge \ \text{CompositionDeep} \ \wedge \ \text{DecoderRich}.$$

L.7 Example: 4-Key Steering Basis (Sketch)

Let $\Sigma = \{h, j, k, l\}$ be primitive actions. If a decoder Eval maps sequences in Σ^* to 2D trajectories and then to symbols, then $|\Sigma| = 4$ can be sufficient (in principle) to generate a large lexical space. This parallels NAND completeness: minimal alphabet; expressive closure via decoding.

L.8 Summary

Functional completeness provides a strict, constructive demonstration that:

$$\text{Extremely small primitive sets can generate full expressive power.}$$

Therefore, "language simplicity" is consistent with maximal expressivity: the limiting factor is not alphabet size but closure structure (composition + interpretation).

Appendix M: Time–Entropy Tradeoff in Representational Systems

M.0 Introduction

This appendix formalizes the equivalence of symbolic systems and the tradeoff between expressive efficiency and representational entropy. All sufficiently expressive formalisms are shown to be computationally equivalent, differing only in description length and execution cost.

M.1 Universal Computation Framework

Let $\mathscr{C}$ be the class of all computable functions.

A formal system F is *universal* iff:

$$\forall f \in \mathscr{C}, \ \exists p \in F \text{ such that } \mathrm{Eval}_F(p) = f$$

Examples:

$$F \in \{\text{Python}, \ \text{JavaScript}, \ \lambda\text{-calculus}, \ \text{ZFC}, \ \text{Category theory}, \ \text{Boolean logic}\}$$

All are Turing-complete encodings of $\mathscr{C}$.

M.2 Representation Length Function

For system F, define description length:

$$L_F(f) = \underset{p:\text{Eval}_F(p)=f}{min} |p|$$

Different formalisms induce different length functions.

Equivalence. For any two universal systems F_1, F_2:

$$\exists c_{12} > 0 \text{ such that } L_{F_1}(f) \leq c_{12} L_{F_2}(f) + c_{12}$$

(Compiler invariance theorem).

M.3 Time Complexity Translation

Let $T_F(p)$ be the runtime of program p in formalism F.

For universal F_1, F_2:

$$T_{F_1}(f) \leq k \cdot T_{F_2}(f) + k$$

for some constant k depending on the interpreter.

Thus, systems differ only by constant-factor overhead.

M.4 Entropy of a Formal Language

Let Σ_F be the primitive alphabet of F.

Define representational entropy:

$$H_F = log\, |\Sigma_F|$$

Let $\mathscr{P}_F(f)$ be the distribution of valid programs implementing f.

Define encoding entropy:

$$\mathscr{E}_F(f) = -\sum_{p \in \mathscr{P}_F(f)} P(p)\, log\, P(p)$$

Richer primitive sets reduce $L_F(f)$ but increase H_F.

M.5 Minimal Alphabets vs. Program Length

Functional completeness implies:

$$\exists F_{min} \,:\, |\Sigma_{F_{min}}| \text{ minimal and } F_{min} \text{ universal}$$

However:

$$L_{F_{min}}(f) \gg L_{F_{\text{rich}}}(f)$$

Thus minimal languages are expressive but inefficient.

M.6 Linguistic Analogue

Let G be a grammar encoding semantics $\mathcal{S}$.

Define utterance length:

$$L_G(s) = \underset{w \in G : D(w) = s}{min} |w|$$

For any two complete grammars G_1, G_2:

$$L_{G_1}(s) \le c L_{G_2}(s) + c$$

Languages differ only in compression efficiency.

M.7 Gesture Universality

Let Γ be a primitive gesture set.

Define a gestural language F_Γ with composition.

If F_Γ is universal:

$$\forall s \in \mathcal{S},\ \exists \gamma \in \Gamma^* \,:\, D(\gamma) = s$$

But:

$$|\Gamma| \downarrow \Longrightarrow L_\Gamma(s) \uparrow$$

Any meaning is expressible, but at higher temporal cost.

M.8 Programming Language Analogy

Let f be a task.

Then:

$$L_{\text{Python}}(f) \neq L_{\text{JavaScript}}(f)$$

Yet:

$$\mathrm{Eval}_{\mathrm{Python}}(p) = \mathrm{Eval}_{\mathrm{JavaScript}}(q) = f$$

All universal languages differ only by representational efficiency.

M.9 Human Communication Optimization Principle

Humans minimize total communication cost:

$$C = L_F(s) \cdot \tau_F$$

where τ_F is time per symbol.

Languages evolve toward low C, not minimal Σ.

M.10 Equivalence Theorem

All universal symbolic systems satisfy:

$$\mathrm{ExpressivePower}(F_1) = \mathrm{ExpressivePower}(F_2)$$

Differences arise only in:

$$(L_F,\ T_F,\ H_F)$$

Language diversity reflects optimization, not capability.

M.11 Consequence

Choice of formalism is an engineering decision over:

$$\mathrm{Time} \times \mathrm{Length} \times \mathrm{Entropy}$$

not a difference in what can be expressed.

Appendix N: Cultural Artifacts as Executable Programs

N.0 Introduction

This appendix formalizes cultural artifacts as executable programs. Texts, images, music, and films are treated as symbolic objects that induce state transformations in cognitive systems. The result follows from general principles of computation and representation and subsumes results due to Wittgenstein, Church, Turing, Curry, and Howard.

N.1 Programs as State Transformations

Let $\mathscr{H}$ denote the space of human cognitive states.

A program is defined abstractly as a function:

$$p : \mathscr{I} \longrightarrow \mathscr{O}$$

For cultural artifacts, define:

$$d : \mathscr{H} \longrightarrow \mathscr{H}$$

where d maps an initial cognitive state to a transformed cognitive state.

Thus any artifact inducing a systematic state transition is a program.

N.2 Execution Semantics

Let Exec_u denote execution by user u.

Define meaning as runtime behavior:

$$\text{Meaning}(d, u) = \text{Exec}_u(d)$$

An artifact is semantically inert until executed:

$$\text{Meaning}(d, u) \text{ undefined if } \text{Exec}_u(d) \text{ not invoked}$$

This is identical to program semantics.

N.3 Media as Instruction Sets

Let $\mathscr{M}$ be a set of media (text, image, audio, video).

Each medium defines an encoding:

$$E_m : \mathscr{S} \longrightarrow \Sigma_m^*$$

and a decoder:

$$D_m : \Sigma_m^* \longrightarrow \mathscr{S}$$

Semantic invariance:

$$D_m(E_m(s)) = s$$

Thus all media are equivalent instruction sets for the same semantic algebra.

N.4 Wittgenstein Correspondence

In the *Tractatus*, Wittgenstein states that propositions are pictures of states of affairs.

Formally:

$$\varphi : \text{WorldStates} \longrightarrow \Sigma^*$$

with truth defined by structural correspondence.

This is identical to program semantics:

$$\text{Eval}(\varphi) = \text{WorldTransition}$$

Hence pictorial representation is a program mapping world models to world models.

N.5 Curry–Howard Isomorphism

Curry–Howard states:

$$\text{Proof} \equiv \text{Program}$$

$$\text{Proposition} \equiv \text{Type}$$

Let d be a document asserting P.

Then:

$$d \text{ is a proof of } P \iff d \text{ is a program of type } P$$

Thus textual artifacts are typed programs.

N.6 Church–Turing Universality

Let $\mathscr{C}$ be the class of computable functions.

Church–Turing Thesis:

$$\mathscr{C} = \{\text{Turing-computable functions}\}$$

If an artifact induces any computable transformation of $\mathscr{H}$, then:

$$d \in \mathscr{C}$$

Hence cultural artifacts are computational objects.

N.7 Equivalence of Artifacts and Algorithms

Let f_d be the function induced by artifact d:

$$f_d(h) = \mathrm{Exec}_u(d)(h)$$

If f_d is computable, then:

$$\exists p \,:\, \mathrm{Eval}(p) = f_d$$

Thus every artifact corresponds to a program.

N.8 Genre as Algorithm Class

Let $\mathcal{D}$ be a corpus of artifacts.

Define equivalence:

$$d_i \sim d_j \iff f_{d_i} \equiv f_{d_j}$$

Genres partition artifacts by computational behavior.

N.9 Medium Invariance Theorem

For any two media m_1, m_2 and any artifact d:

$$\exists d' \,:\, E_{m_1}(d) \equiv E_{m_2}(d')$$

All cultural forms are mutually compilable.

N.10 Consequence

Libraries, archives, and media systems are repositories of executable semantic objects.

Civilization constitutes a distributed codebase over $\mathcal{H}$.

Appendix O: Scope Memeplexes as Linguistic–Computational Replicators

O.0 Introduction

This appendix formally unifies the preceding results. Interface systems, programming languages, media forms, and cultural artifacts are shown to be instances of a single class: symbolic execution environments competing to control semantic space. Scope memeplexes are defined as replicators operating over this space.

O.1 Unified Symbolic Substrate

From Appendices I–N, define a universal semantic space:

$$\mathcal{S}$$

All languages, media, and interfaces define encodings:

$$E_i : \mathcal{S} \longrightarrow \Sigma_i^*$$

$$D_i : \Sigma_i^* \longrightarrow \mathcal{S}$$

with:

$$D_i(E_i(s)) = s$$

Hence all systems are isomorphic encodings of $\mathcal{S}$.

O.2 Execution Environments

Each system i defines an execution environment:

$$\mathcal{E}_i = (\Sigma_i, \circ_i, \mathrm{Eval}_i)$$

where Eval_i executes symbolic programs on cognitive state space:

$$\mathrm{Eval}_i : \Sigma_i^* \times \mathcal{H} \longrightarrow \mathcal{H}$$

Thus, all interfaces and media are computational substrates.

O.3 Definition of Scope Memeplex

A scope memeplex M_i is defined as:

$$M_i = (\mathcal{E}_i, \rho_i)$$

where:

- $\mathcal{E}_i$ is a symbolic execution environment
- $\rho_i(t)$ is its population prevalence

Scope memeplexes replicate by inducing adoption of $\mathcal{E}_i$.

O.4 Linguistic Colonization

Let $\mathcal{L}_u$ be a user language.

Embedding cost:

$$C(M_i, u) = \mathbb{E}_{s \sim \mathcal{L}_u} [|E_i(s)|]$$

Users migrate toward systems minimizing:

$$C(M_i, u)$$

Thus scope memeplexes compete to become the dominant encoding of user language.

O.5 Application Wars as Grammar Wars

Each application ecosystem defines a control grammar:

$$G_i \subset \Sigma_i^*$$

Control operations are executive verbs over $\mathscr{H}$.

Conflict condition:

$$G_i \neq G_j$$

Hence application wars are grammar wars over action semantics.

O.6 Media, Programs, and Interfaces as a Single Class

From Appendix N, all cultural artifacts are programs.

From Appendix M, all universal systems are equivalent in expressive power.

Therefore:

$$\text{Interface} \equiv \text{Language} \equiv \text{Program}$$

Distinctions are representational, not structural.

O.7 Replicator Dynamics

Let F_i be fitness of scope memeplex M_i:

$$F_i = -\mathbb{E}_u[C(M_i, u)]$$

Population dynamics:

$$\frac{d\rho_i}{dt} = \rho_i(F_i - \bar{F})$$

Scope memeplexes evolve by minimizing linguistic embedding cost.

O.8 Enshittification as Linguistic Monoculture

Let diversity:

$$\mathscr{D}(t) = -\sum_i \rho_i(t)\, log\, \rho_i(t)$$

Collapse condition:

$$\exists i \,:\, \rho_i(t) \longrightarrow 1$$

This is equivalent to grammatical monoculture over $\mathcal{S}$.

O.9 Browser and Platform Wars

Browsers and platforms are competing encoders E_i.

Victory condition:

$$E_i \approx arg\,\underset{E}{min}\,C(E, u)$$

They do not compete over truth, but over linguistic embedding efficiency.

O.10 Central Equivalence Theorem

All of the following are identical object classes:

$$\text{Language},\ \text{Interface},\ \text{Program},\ \text{Medium}$$

Each is a symbolic execution environment over $\mathcal{S}$.

Scope memeplexes are replicators over this unified class.

O.11 Thesis Closure

The enshittification crisis is therefore a consequence of uncontrolled replication dynamics among symbolic execution environments.

Application wars are linguistic wars.

Interface evolution is language evolution.

All are instances of scope memeplex competition over semantic space.

References

[1] C. Doctorow. The Enshittification of TikTok. *Pluralistic*, 2023.

[2] R. Zebrowski. *Macrolife: A Mobile Utopia.* University of Washington Press, 2016.

[3] E. Barenholtz, S. Goldstein, T. B. Lee, and J. B. Tenenbaum. Recursive autoregression in human language processing. *Trends in Cognitive Sciences*, 25(11):1038–1050, 2021.

[4] K. Friston. The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138, 2010.

[5] A. Clark. *Surfing Uncertainty: Prediction, Action, and the Embodied Mind.* Oxford University Press, 2016.

[6] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[7] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[8] S. Dehaene. *The Number Sense: How the Mind Creates Mathematics.* Oxford University Press, 2011.

[9] D. A. Norman. *The Design of Everyday Things.* Basic Books, Revised Edition, 2013.

[10] S. K. Card, T. P. Moran, and A. Newell. *The Psychology of Human-Computer Interaction.* Lawrence Erlbaum Associates, 1983.

[11] E. Hutchins. *Cognition in the Wild.* MIT Press, 1995.

[12] D. Kirsh and P. Maglio. On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18(4):513–549, 1994.

[13] B. L. Whorf. *Language, Thought, and Reality.* MIT Press, 1956.

[14] W. Labov. *Sociolinguistic Patterns.* University of Pennsylvania Press, 1972.

[15] P. Lieberman. *The Biology and Evolution of Language.* Harvard University Press, 1984.

[16] J. Maynard Smith and E. Szathmáry. *The Major Transitions in Evolution.* Oxford University Press, 1995.

[17] R. Dawkins. *The Selfish Gene.* Oxford University Press, 1976.

[18] D. C. Dennett. *Darwin's Dangerous Idea.* Simon & Schuster, 1995.

[19] T. Winograd and F. Flores. *Understanding Computers and Cognition.* Ablex Publishing, 1986.

[20] L. Suchman. *Plans and Situated Actions.* Cambridge University Press, 1987.

[21] D. Engelbart. Augmenting human intellect: A conceptual framework. Stanford Research Institute Report, 1962.

[22] M. McLuhan. *Understanding Media: The Extensions of Man.* McGraw–Hill, 1964.

[23] J. Lanier. *You Are Not a Gadget.* Knopf, 2010.

[24] L. Wittgenstein. *Tractatus Logico-Philosophicus.* Routledge & Kegan Paul, 1922.

[25] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1936.

[26] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, 1936.

[27] S. C. Kleene. *Introduction to Metamathematics.* North-Holland, 1952.

[28] W. A. Howard. The formulae-as-types notion of construction. In J. P. Seldin and J. R. Hindley (eds.), *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, 1980.

[29] H. B. Curry. Functionality in combinatory logic. *Proceedings of the National Academy of Sciences*, 20(11):584–590, 1934.

[30] E. L. Post. *The Two-Valued Iterative Systems of Mathematical Logic.* Annals of Mathematical Studies 5, Princeton University Press, 1941.

[31] H. M. Sheffer. A set of five independent postulates for Boolean algebras. *Transactions of the American Mathematical Society*, 14(4):481–488, 1913.

[32] H. Enderton. *A Mathematical Introduction to Logic.* Academic Press, 2nd edition, 2001.

[33] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer, 3rd edition, 2008.

[34] R. J. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7(1–2):1–22, 224–254, 1964.

[35] G. J. Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM*, 22(3):329–340, 1975.

[36] M. Sipser. *Introduction to the Theory of Computation.* Cengage Learning, 3rd edition, 2012.

[37] M. A. Arbib. *From Monkey-like Action Recognition to Human Language.* Behavioral and Brain Sciences, 28(2):105–167, 2005.

[38] V. Gallese and G. Lakoff. The brain's concepts: The role of the sensory-motor system in conceptual knowledge. *Cognitive Neuropsychology*, 22(3–4):455–479, 2005.

[39] M. Tomasello. *Constructing a Language: A Usage-Based Theory of Language Acquisition.* Harvard University Press, 2003.

[40] R. Jackendoff. *Foundations of Language.* Oxford University Press, 2002.

[41] G. Lakoff and M. Johnson. *Philosophy in the Flesh.* Basic Books, 1999.

[42] J. Fodor. *The Modularity of Mind.* MIT Press, 1983.

[43] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1–3):139–159, 1991.

[44] A. Newell and H. A. Simon. Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19(3):113–126, 1976.

[45] D. Hofstadter. *Gödel, Escher, Bach: An Eternal Golden Braid.* Basic Books, 1979.

[46] L. Floridi. *The Fourth Revolution: How the Infosphere Is Reshaping Human Reality.* Oxford University Press, 2014.

[47] F. Heylighen. The growth of structural and functional complexity during evolution. In F. Heylighen, J. Bollen, and A. Riegler (eds.), *The Evolution of Complexity*, Kluwer, 1999.

[48] J. H. Holland. *Hidden Order: How Adaptation Builds Complexity.* Addison–Wesley, 1995.

[49] H. A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 1962.