Issue VIII: Quantum Interpreter

Максим Сохацький [1]

[1] Національний технічний університет України
«Київський Політехнічний Інститут» ім. Ігора Сікорського
29 жовтня 2018

Анотація

Ця робота є спробою огляду існуючих мов програмування для квантових обчислень, їх теоретичних основ та особливостей. Як приклад розглядається дискретне перетворення Фур'є, яке в якості вправи імплементується на двох мовах програмування: практичній, імперативній QCL від Бернхарда Омера[12] та теоретичній формальній, функціональній мові програмування, лямбда-численні з лінійними функціями від Андре ван Тондера[16]. Як висновок пропонуємо нарис своєї мови PLQ для квантових обчислень з залежними, залежними-лінійними та квантовими типами.
Ключові слова: Теорія типів, квантові комп'ютери

Зміст

Присвячується

Феліксу Блоху

1 The Felix Language

1.1 Лінійна алгебра

Нотація Дірака це компактний формалізм лінійної алгебри який будемо застосовувати для визначень квантової механіки[1].

Табл. 1: Нотація Дірака

Нотація	Визначення
$\|\psi\rangle$	загальний кет-вектор, наприклад $(c_0, ..., c_n)^T$
$\langle\psi\|$	дуальний бра-вектор, наприклад $(c_0^*, ..., c_n^*)$
$\|n\rangle$	n-й базис вектор стандартного базису $N = (\|0\rangle, ..., \|n\rangle)$
$\|\tilde{n}\rangle$	n-й базис вектор альтернативного базису $\tilde{N} = (\|\tilde{0}\rangle, ..., \|\tilde{n}\rangle)$
$\langle\phi\|\psi\rangle$	скалярний добуток
$\|i, j\rangle$	тензорний добуток базисних векторів $\|i\rangle$ та $\|j\rangle$
$\|\phi\rangle \otimes \|\psi\rangle$	тензорний добуток

Definition 1. (Векторний простір). Множина V називається векторни мпростором над скалярним полем F, тоді і тільки тоді, коли визначені операції $+ : V \times V \to V$ (сума векторів) та $\cdot : F \times V \to V$ (добуток скаляра та вектора) з наступними властивостями: i) $(V, +)$ утворюють комутативну групу; ii) $\lambda|\psi\rangle = |\psi\rangle\lambda$; iii) $\lambda(\mu|\psi\rangle) = (\lambda\mu)|\psi\rangle$; iv) $(\lambda + \mu)|\psi\rangle = \lambda|\psi\rangle + \mu|\psi\rangle$; v) $\lambda(|\psi\rangle + |\varphi\rangle) = \lambda|\psi\rangle + \lambda|\varphi\rangle$. Далі будемо розглядати скалярне поле комплексних чисел $F = C$.

Definition 2. (Скалярний добуток). Функція $\langle\cdot|\cdot\rangle : V \times V \to C$ називається скалярним добутком, тоді і тільки тоді, коли: i) $\langle\psi|(\lambda\varphi\rangle + \mu|\chi\rangle) = \lambda\langle\psi|\varphi\rangle + \mu\langle\psi|\varphi\rangle$; ii) $\langle\psi|\varphi\rangle = \overline{\langle\varphi|\psi\rangle}^*$; iii) $0 < \langle\psi|\psi\rangle \in \mathbb{R}$. Скалярний добуток визначає норму $\| \, |\psi\rangle \, \| = \sqrt{\langle\psi|\psi\rangle} = \| \, \psi \, \|$.

Definition 3. (Повний векторний простір). Нехай V векторний простір з нормою $\| \cdot \|$ та $|\psi_n\rangle \in V$ послідовність векторів. i) $|\psi\rangle$ є послідовністю Коші ттт. $\forall\epsilon > 0 \exists N > 0 : \forall n, m > N, \| \, |\psi_n\rangle - |\psi_{n+1}\rangle \, \| < \epsilon$. ii) $|\psi\rangle$ сходиться ттт. $\forall\epsilon > 0 \exists N > 0 : \forall n > N, \| \, |\psi_n\rangle - |\psi\rangle \, \| < \epsilon$. Простір V повний ттт. кожна послідовність Коші сходиться.

Definition 4. (Гільбертів простір). Повний векторний простір H зі скалярний добутком $\langle\cdot|\cdot\rangle$ та відповідною нормою $\| \, \psi \, \| = \sqrt{\langle\psi|\psi\rangle}$ називається Гільбертовим.

[1] І.О. Вакарчук. Квантова Механіка. 2012

Definition 5. (Лінійний оператор). Нехай V – векторний простір, а A – функція $A : V \to V$. Тоді A називається лінійним оператором ттт.

$$A(\lambda|\psi\rangle + \mu|\varphi\rangle) = \lambda A|\psi\rangle + \mu A|\varphi\rangle$$

В C^n лінійний оператор є матрицею $m \times n$ з елементами $a_{i,j} = \langle i|A|j\rangle$, де $A = \Sigma_{i,j} a_{i,j}|i\rangle\langle j|$. За визначенням лінійності оператор A можно записати через лінійну суму векторів базису B:

$$A : |n\rangle \to \Sigma_k a_{kn}|k\rangle, \text{ де } |k\rangle \in B.$$

Definition 6. (Тензорний добуток гільбертових просторів). Нехай H_1 та H_2 — Гільбертові простори з базисами B_1 та B_2. Тоді тензорний добуток

$$H = H_1 \otimes H_2 = \{\Sigma_{|i\rangle \in B_1} \Sigma|j\rangle \in B_2 c_{ij}|i,j\rangle \| c_{ij} \in \mathbb{C}\}.$$

також Гільбертів простір з базисом $B = B_1 \times B_2$ та скалярним добутком:

$$\langle i,j|i',j'\rangle = \langle i|i'\rangle\langle j|j'\rangle = \delta_{ii'}\delta jj', \text{ де } |i\rangle, |i'\rangle \in B_1, |j\rangle, |j'\rangle \in B_2.$$

Definition 7. (Тензорний добуток лінійних операторів). Нехай A та B лінійні оператори на Гільбертових протосторах H_1 та H_2, тоді тензорний добуток

$$A \otimes B = \Sigma_{i,j}\Sigma_{i',j'}|i,j\rangle\langle i|A|i'\rangle\langle j|B|j'\rangle\langle i',j'|$$

лінійний оператор на на гільбертовому просторі $H_1 \otimes H_2$.

Definition 8. (Комутатор та антикомутатор). Нехай A та B лінійні оператори на гільбертовому просторі H. Оператор $[A,B] = AB - BA$ називається комутатором, а $A,B = AB + BA$ називається антикомутатором.

1.2 Інтерпретація квантової механіки

В залежності від того як саме моделюються та конструюються гільбертові простори та гамільтоніани, виникають різні теорії, від нерятивістської квантової електродинаміки до квантової хронодинаміки яка вводить поняття кварків та глюонів.

Теорія квантових обчислень — це ще одна теорія поверх абстрактного квантового формалізму та є інтерпретацією квантової механіки. Однак це не фізична теорія в тому сенсі, що вона не описує природній процес, а є ближчою до схемотехніки, з квабітами та квантовими вентилями, без визначення як саме моделюється квантова система, вона може бути або фізичним об'єктом або симулятором.

Точно так як для апаратного забезпечення будуються мови порграмування та вищі мови програмування, так само для квантових обчислень, квантових станів та квантових логічних елементів (вентилів), існують свої мови програмування. У наступній секції дамо огляд існуючих мов та підходів до їх побудови, а тут дамо основні принципи та компоненти архітекти квантових обчислень, аби пояснити основні мовні елементи, та їх перетворення.

1.3 Пам'ять квантового комп'ютера

Definition 9. (Квантовий біт). Квантовий біт або квабіт визначається як квантова система, стан якої може бути повністю виражений як суперпозиція (лінійна комбінація) двох ортонормованих власних базових станів позначених $|0\rangle$ та $|1\rangle$. Загальний стан $|\psi\rangle$ квабіта тоді визначається як $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, |\alpha|^2 + |\beta|^2 = 1$. Значення квабіта описується спостереженням $N = |1\rangle\langle 1|$. $\langle N\rangle$ дає вірогідність знайти систему в стані $|1\rangle$, якщо над квабітом були проведені виміри. Простір станів квабіта є гільбертовим простором $H = \mathbb{C}^2$. Ортонормована система $|0\rangle, |1\rangle$ називається обчислювальним базисом.

Definition 10. (Сфера Блоха). Загальний стан квабіта може бути виражений в полярних координатах θ та ϕ:

$$|\psi\rangle = cos\frac{\theta}{2}|0\rangle + e^{i\phi}sin\frac{\theta}{2}|1\rangle.$$

Одиничний вектор стану $|\psi\rangle$ називається вектором Блоха $\tilde{r}_\psi$, та має наступну властивість $\tilde{r}_\phi = -\tilde{r}_\xi \leftrightarrow \langle\phi|\xi\rangle = 0$. Оберти навколо осей X, Y та Z

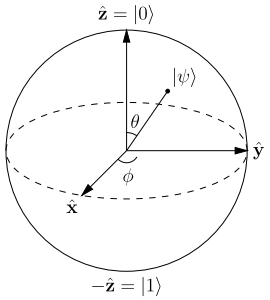

Рис. 1: Сфера Блоха як представлення квабіта $|\psi\rangle$

вектора Блоха тоді визначаються як оператори:

$$X(\theta) = \begin{bmatrix} cos\frac{\theta}{2} & -isin\frac{\theta}{2} \\ -isin\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix}, Y(\theta) = \begin{bmatrix} cos\frac{\theta}{2} & -sin\frac{\theta}{2} \\ sin\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix}, Z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

Definition 11. (Квантова система).

Definition 12. (Еволюція системи). Темпоральна еволюція стану системи описується рівнянням Шрьодінгера:

$$i\hbar\frac{\delta}{\delta t}|\psi\rangle = H|\psi\rangle.$$

де $\hbar \equiv 1.05457 \cdot 10^{-43}$ експериментальна константа Планка, а H — фіксований самоспряжений оператор на гільбертовому просторі, знаний як Гамільтоніан квантової системи. В квантовій фізиці нормують відносно $\hbar$ тоді

рівняння можна записани у безвимірній формі $i|\psi\rangle = H|\psi\rangle$. Гамільтоніан H повністю визначає квантову систему. Другий спосіб визначення, через унітарний оператор $U = e^{-iH}$. Темпоральна еволюція замкненої квантової системи зі страну $|\psi\rangle$ та часу t_1 в стан $|\psi'\rangle$ та часу t_2 може бути описана унітарним оператором $U = U(t_2 - t_1)$, таким, що $|\psi'\rangle = U|\psi\rangle$.

Definition 13. (Вимірювання). Проективне вимірювання визначається як самоспряжений оператор M який називається спостереженням зі спектральною композицією $M = \Sigma_m m P_m$, де P_m проекція на власний простір власного значення m. Власні значення m оператора M відповідаються усім можливим результатам вимірювання. Вимірювання $|\psi\rangle$ дасть результат m з вірогідністю $p(m) = \langle\psi|P_m|\psi\rangle$, таким чином через скорочення $|\psi\rangle$ отримаємо новий стан системи $|\psi'\rangle = \frac{1}{\sqrt{p(m)}} P_m|\psi\rangle$. Для стану квабіта, самоспряжений оператор N знаний як стандартне вимірювання.

$$N = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = 0 \cdot |0\rangle\langle 0| + 1 \cdot |1\rangle\langle 1|.$$

Біль загально для простору стану $H = C^n$ стандартне вимірювання визначається як $N = \Sigma_i i|i\rangle\langle i|$.

Definition 14. (Виважене середнє). Виважене середнє $\langle M\rangle$ усіх можливих результатів вимірювання M називється очікуваним значенням та визначається як

$$\langle M\rangle = \Sigma_n p(m)m = \Sigma_m \langle\psi|mP_m|\psi\rangle = \langle\psi|M|\psi\rangle.$$

1.4 Оператори обчислювального ядра

Definition 15. (Оператори Паулі). Оператори Паулі пов'язані з довільним обертом вектора Блоха формулою:

$$R_{\tilde{n}}(\theta) = e^{-\frac{i}{2}\theta\tilde{n}\cdot(x,y,z)} = cos\frac{\theta}{2}I - isin\frac{\theta}{2}(n_x X + n_y Y + n_z Z)$$

де X,Y,Z — оператори:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Найпростіший випадок унітарного перетворення це оператор який діє на одиничний квабіт. Загальна форма 2-вимірної комплексної унітарної матриці $U \in SU(2)$ має вигляд:

$$U = e^{i\varphi} \begin{pmatrix} e^{\frac{i}{2}(-\alpha-\beta)}cos\frac{\theta}{2} & -e^{\frac{i}{2}(-\alpha+\beta)}sin\frac{\theta}{2} \\ e^{\frac{i}{2}(\alpha-\beta)}cos\frac{\theta}{2} & e^{\frac{i}{2}(\alpha+\beta)}sin\frac{\theta}{2} \end{pmatrix}$$

Нехай $U \in SU(2)$ має базисні вектори $|u\rangle$ та $|v\rangle$ та власні значення u та v. Тоді U можна виразити:

$$U = u|u\rangle\langle u| + v|v\rangle\langle v| = e^{i\varphi}R_{\tilde{n}}(\delta),$$

де δ — фазова різниця між u та v — пов'язана як $v = e^{i\delta}v$.

Definition 16. (Оператор Адамара).

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Definition 17. (Фазовий та $\pi/8$ оператори).

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, R_3 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

Definition 18. (Контрольований вентиль). Нехай U унітарний q-квабітний вентиль, контрольований U-вениил з n-контрольними бітами тоді визначається як

$$C^m[U] = \begin{bmatrix} I & \dots & 0 & 0 \\ \vdots & \ddots & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & U \end{bmatrix}$$

в просторі станів $B^{\otimes n+m}$.

Definition 19. (Контрольований НЕ вентиль).

$$CNot : |x, y\rangle \to |x \otimes y, y\rangle$$

$$CNot = C[X] = \begin{bmatrix} I & 0 \\ 0 & X \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Definition 20. (Вентиль перестановки).

$$Swap : |x, y\rangle \to |y, x\rangle$$

Definition 21. (Контрольований фазовий вентиль).

$$CPhase : |x, y\rangle \to i^{xy}|x, y\rangle$$

Definition 22. (Вентиль Тофолі).

$$CCNot : |x, y, z\rangle \to i^{xy}|x, y\rangle$$

1.5 Огляд існуючих мов

З огляду на новизну предмету розроблено не так багато мов, усі що є можна розділіти на імперативні (по духу своєї імплемантації) та функціональні (або побудовані на базі певного виду лямбда числення). Ми наведемо приклади програм для обох підходів. У якості порівняльної характеристики візьмемо алгоритм дискретного перетворення Фур'є.

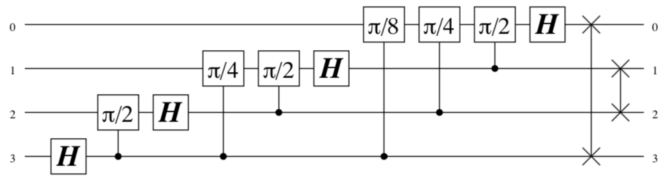

Рис. 2: Графічне представлення квантової схеми вентилів дискретного перетворення Фур'є для 4-квабітного регістру

1.5.1 Імперативні мови програмування

Станом на 2018 рік найбільш розробленою мовою, яка компілюється під Linux та Mac є QLC від Бернхарда Омера [2]. До QLC можна відзначти наступні мови: 1) Q-gol від Грега Бейкера (1996) [3]; 2) qGCL від Паоло Зуліані (2000)[14]; 3) Quantum C Language від Стефана Блаха (2002); 4) QPAlg від Марі Лолір та Філіпа Жорранда (2004)[11], синтаксис та семантика цієї системи базується на численні процесів Мілнера CCS та формальній мові LOTOS; 5) CQP (Communication Quantum Processes) від Саймона Гея та Раджагопала Нагараджана (2004)[8]; 6) Q (DSL мова для C++) від Беттеллі, Каларко та Серафіні[**?**]. 7) LanQ від Гинека Млнаріка[13].

Definition 23. (Перестановка).

```
cond qufunct Swap(qureg a,qureg b) {
  int i; if #a!=#b { exit "arguments must equal"; }
  for i=0 to #a-1 {
    CNot(b[i],a[i]);
    CNot(a[i],b[i]);
    CNot(b[i],a[i]); } }
```

Definition 24. (Зміна порядку квабітів).

```
cond qufunct flip(qureg q) {
  int i; for i=0 to #q/2-1 { Swap(q[i],q[#q-i-1]); } }
```

Definition 25. (Дискретне перетворення Фур'є, Коперсміт).

```
operator dft(qureg q) {
  const n=#q; int i; int j;
  for i=1 to n {
      for j=1 to i-1 { V(pi/2^(i-j),q[n-i] & q[n-j]); }
      H(q[n-i]); }
  flip(q); }
```

[2] Bernhard Ömer. Structured Quantum Programming. PhD. TU Vienna. 2003. `http://tph.tuwien.ac.at/~oemer/doc/structquprog.pdf`

[3] Gregory David Baker. Qgol. A system for simulating quantum computations: Theory, Implementation and Insights. 1996. PhD. Macquarie University. `http://www.ifost.org.au/~gregb/q-gol/QgolThesis.pdf`

1.5.2 Функціональні мови програмування

Лямбда числення лежить в основі сучасних алгебраїчних систем та основ математики[6], воно може бути розглянуте не лише як мова програмування загального використання, але і фреймворк для судження про обчислювальні властивості програм.

Таксономію лямбда числень, яку виконав Хенк Барендрегт та подав у вигляді лямбда кубу[7], можна розширити квантовими мовними примітивами, вводити в вищі лямбда числення з гомотопічними типами, або в системи доведення теорем з екстрактом або лише верифікацією.

З огляду на природу квантових обчислень необхідним компонентом квантового лямбда числення є система лінійних типів, де за час існування змінної в області видимості під час виконання дозволяється звертання до неї тільки один раз. Така система типів уже використовується не тільки в есперементальних верифікаторах але і в сучасних системних мовах програмування (Rust), де завдяки верифікатору лініних типів вдається обійтися без алгоритмів автоматичного вивільнення памяті під час виконання програми (схема памяті повністю моделюється під час компіляції програми).

Серед робіт присвячених мовам на базі лямбда числень можна відзначити наступні: 1) Найбільш класичне нетипизоване лямбда числення Андре ван Тондера[16]. Тут подається доведення ізоморфізму машині Тюрінга, повнота та звучання системи, що є формальною перевагою перед імперативними мовами, такими як QCL; 2) Quipper від Гріна, Люмсдейна, Росса, Селінжера та Валірона [10][9]. Це спроба вбудувати DSL для квантових обчислень в мову Haskell, симулятор мови побудований на генераторах групи Кліффорда, квантових операторах X,Y,Z,H,S. 3) Робота по лямбда численню від Аррігі та Довека[5]. 4) QML для Haskell від Торстена Альтенкірха та Джонатана Граттажа[1][2][3].

Definition 26. (Синтаксичне дерево O_H). Синтаксичне дерево O_H визначає лямбда числення з лінійними змінними поєднане з класичним нетипизованим лямбда численням. Тобто O_H є найпростішим операційним квантовим лямбда численням для середовищ виконання.

$$
\begin{array}{l}
\mathrm{c} = \mathbf{0} \mid \mathbf{1} \mid \mathbf{H} \mid \mathrm{S} \mid R_3 \mid \mathbf{CNot} \mid \mathrm{X} \mid \mathrm{Y} \mid \mathrm{Z} \mid \ldots \\
\mathrm{t} = \mathrm{x} \mid \lambda\ \mathrm{x}\ .\ \mathrm{t} \mid \mathbf{let}\ \mathrm{x} = \mathrm{t}\ \mathbf{in}\ \mathrm{t} \mid \mathrm{t}\ \mathrm{t} \\
\qquad \mid (\mathrm{t}, \mathrm{t}) \mid \mathrm{t} \mid \mathrm{c} \mid\ !\ \mathrm{t} \mid \lambda\ !\ \mathrm{x}\ .\ \mathrm{t}
\end{array}
$$

Тут даються примітиви лінійних типів, де доступ до змінної можливий лише раз в обсласті визначення змінної. Для нелінійних або звичайних лямбда функції даються примітиви позначені !. В переліку квантових примітивів с даються: i) ортонормований базис $|0\rangle$ та $|1\rangle$; ii) H — оператор Адамара; iii) S — фазовий вентиль; iv) R_3 — $\pi/8$ вентиль; v) контрольований не вентиль CNot; vi) вентилі Паулі X, Y та Z.

Definition 27. (Пара Айнштайна-Подольського-Розена).

$$\mathbf{EPR} = \mathbf{CNot}\ ((\mathbf{H}\ \mathbf{0}), \mathbf{0}) \tag{1}$$

Definition 28. (Квантова телепортація).

$$
\begin{array}{l}
\mathbf{teleport}\ x = \mathbf{let}\ (e_1, e_2)\ =\ \mathbf{EPR}\ \mathbf{in} \\
\qquad\qquad \mathbf{let}\ (x', y') = \mathbf{alice}\ (x, e_1)\ \mathbf{in}\ \mathbf{bob}\ (x', y', e_2)
\end{array} \tag{2}
$$

$$
\begin{array}{l}
\mathbf{alice}\ (x, e_1) = \mathbf{let}\ (x', y') = \\
\qquad\qquad \mathbf{CNot}\ (x, e_1)\ \mathbf{in}\ ((\mathbf{H}\ x'), y') \\
\mathbf{bob}\ (x', y', e_2) = \mathbf{let}\ (y'', e_2') = cX\ (y', e_2)\ \mathbf{in} \\
\qquad\qquad \mathbf{let}\ (x'', e_2'') = cZ\ (x', e_2')\ \mathbf{in}\ (x'', y'', e_2'')
\end{array} \tag{3}
$$

Definition 29. (Дискретне перетворення Фур'є).

$$\mathbf{fourier}\ list\ =\ \mathbf{reverse}\ \mathbf{fourier}'\ list \tag{4}$$

$$
\mathbf{fourier}'\ list\ =\ \mathbf{case}\ list\ \mathbf{of}
\begin{cases}
() \to () \\
h : t \to \mathbf{let}\ h'\ :\ t' = \mathbf{phases}\ (\mathbf{H}\ h)\ t\ !2 \\
\mathbf{in}\ h'\ :\ \mathbf{fourier}'\ t'
\end{cases} \tag{5}
$$

$$
\begin{array}{l}
\mathbf{phases}\ target\ controls\ !n\ = \\
\qquad \mathbf{case}\ control\ \mathbf{of}
\begin{cases}
() \to target \\
control : t \to \mathbf{let}\ (control', target') = \\
(cR\ !n)\ (control, target)\ \mathbf{in} \\
\mathbf{let}\ target'' : t' = \mathbf{phases}\ target'\ t\ !(\mathbf{succ}\ n)\ \mathbf{in} \\
target'' : control' : t'
\end{cases}
\end{array} \tag{6}
$$

1.6 Висновки

Як видно для реалізації семантики мови програмування для квантових комп'ютерів достатньо поєднаяти тензорне числення разом з π-численням процесів, або лінійними типами. На сьогоднішній день (2018) серед імперативних мов програмування найбільш завершена, повна та практична на думку автора є QLC від Бернхарда Омера, серед мов для лямбда числень немає жодної достатньо зрілої імплементації.

1.7 Мова PLQ

Відкрите питання в теорії типів, імплементація та дизайн мови з лінійними та залежними типами та квантовими примітивами. Тому ми пропонуємо як висновок після огляду мов та їх синтаксисів свій синтаксис такої мови, та показуємо з яких інгрідієнтів її можна побудувати.

Мову квантових обчислень O_H можно розкласти до комбінації (прямої суми) трьох синтаксичних дерев: i) дерева O_λ — базового чистого лямбда числення з залежними типами (pure, імплементовану автором[15]); ii) дерева O_π — числення з лінійними типами, або просто інший вид стрілок (linear); iii) операторний базис квантового числення O_Q з конструкторами X,Y,Z,S,H,R_3 та конструктором визначення квабіта (quantum). Тоді формальний синтаксис мови для квантових комп'ютерів, записаний на **cubicaltt**[4] у вигляді індуктивного типу синтаксичного дерева, буде виглядати так:

```
data pure (lang: U)
   = star (n: nat) | var (l: nat) | pi (l: nat) (f: lang)
   | lambda  (x: nat) (f: lang)    | app (f a: lang)

data linear (lang: U)
   = star (n: nat) | var (l: nat) | pi (l: nat) (f: lang)
   | lambda  (x: nat) (f: lang)    | app (f a: lang)

data quantum (lang: U)
   = register (n:nat) | i (n:nat) | 0 | 1 | H (l:lang) | S (:lang)
   | X (f:lang) | Y (f:lang) | Z (f:lang) | CNot (f a: lang)
```

Результуюча авторська мова (назвемо її PLQ) виражається як взаморекурсивна сума елментарних мовних синтаксисів.

```
data PLQ =Pure      (_: pure    PLQ)
         | Linear   (_: linear   PLQ)
         | Quantum  (_: quantum PLQ)
```

З огляду на розмів реферату, семантику цієї мови наводити не будемо, однак, очевидно, що мову Андре ван Тондера можна продовжиди до PTS з лінійними типами в категоріях з сімействами зі значеннями в симетричних моноїдальних категоріях, як було показано Матейсом Вакаром[**?**].

Література

[1] Thorsten Altenkirch and Jonathan Grattage. *A functional quantum programming language.* Logic in Computer Science, Proceedings. 20th Annual IEEE Symposium LICS '05, IEEE, 2005, pp. 249–258.

[2] Thorsten Altenkirch, Jonathan Grattage, Juliana K. Vizzotto, and Amr Sabry. *An algebra of pure quantum programming.* Electronic Notes in Theoretical Computer Science, 170:23–47, 2007.

[3] Jonathan Grattage. *An overview of QML with a concrete implementation in Haskell.* Electronic Notes in Theoretical Computer Science, Proceedings of the Joint 5th International Workshop on Quantum Physics and Logic and 4th Workshop on Developments in Computational Models, 270(1):165–174, QPL/DCM, 2011.

[4] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. *Cubical Type Theory: a constructive interpretation of the univalence axiom.* 2017.

[5] Pablo Arrighi and Gilles Dowek. *Operational semantics for formal tensorial calculus.* 2004.

[6] Andrej Bauer, Steve Awodey, Matthieu Sozeau, Michael Shulman, Dan Licata, Yves Bertot, Peter Dybjer, and Nicola Gambino. *Homotopy Type Theory: Univalent Foundations of Mathematics.* 2013.

[7] H. P. Barendregt. *Lambda calculi with types.* Handbook of Logic in Computer Science (Vol. 2), Oxford University Press, New York, NY, USA, 1992, pp. 117–309.

[8] Simon J. Gay and Rajagopal Nagarajan. *Communicating quantum processes.* Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '05, ACM, New York, NY, USA, 2005, pp. 145–157.

[9] Alexander S. Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. *Quipper: a scalable quantum programming language.* ACM SIGPLAN Notices, vol. 48, ACM, 2013, pp. 333–342.

[10] Alexander S. Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. *An introduction to quantum programming in Quipper.* International Conference on Reversible Computation, Springer, 2013, pp. 110–124.

[11] Marie Lalire and Philippe Jorrand. *A process algebraic approach to concurrent and distributed quantum computation: Operational semantics.* 2004.

[12] Bernhard Ömer. *Structured quantum programming.* 2003.

[13] Hynek Mlnarik. *Operational semantics and type soundness of quantum programming language LanQ.* 2007.

[14] J. W. Sanders and P. Zuliani. *Quantum programming.* In Proceedings of the 5th International Conference on Mathematics of Program Construction, MPC '00, Springer-Verlag, 2000.

[15] Namdak Tonpa. *The systems engineering of consistent pure language with effect type system for certified applications and higher languages.* AIP Conference Proceedings, vol. 1982, 2018.

[16] Andre van Tonder. *A lambda calculus for quantum computation.* SIAM J. Comput., 33(5):1109–1135, 2004.

[17] Matthijs Vakar. *Syntax and semantics of linear dependent types.* arXiv:1405.0033, 2014.