Proposed Thesis Research

Xiaoning Bian

Dalhousie University

Contents

Background

Quantum state, gate and measurement

Definition. An *n-qubit state* ψ is a unit vectors in the 2^n dimensional complex Hilbert space $\mathbf{H}_n = (\mathbb{C}^{2^n}, \langle \_|\_ \rangle)$.

Definition. An *n-qubit gate* is a unitary operator U on $\mathbf{H}_n$.

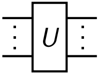
Definition. An *n*-qubit measurement M is given by a finite family $\{S_i\}_i$ of operators satisfying $\Sigma_{i \in I} S_i^\dagger S_i = I$. When performing the measurement M on a quantum state $\varphi \in \mathbf{H}_n$, one of the *measurement outcomes* $i \in I$ will be observed with probability $P(i) = ||S_i \varphi||^2$, and the state will be changed to $\frac{S_i \psi}{\sqrt{P(i)}}$.

Quantum circuit

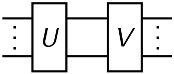
Definition. Quantum circuits are string diagrams for symmetric monoidal groupoid [Joyal and Street, 1991].

$\otimes^n A \quad \longleftrightarrow$

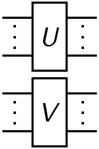
$U : \otimes^n A \to \otimes^n A \quad \longleftrightarrow$



$V \circ U \quad \longleftrightarrow$



$U \otimes V \quad \longleftrightarrow$



QRAM model

A useful abstract model of a quantum computer is the so-called QRAM model [Knill, 1996]. In this model, we assume to have access to N numbered qubits, and be able to perform operations:

- prepare qubit i in state $|0\rangle$ or $|1\rangle$.
- apply n-qubit gates to n distinct qubits $i_1, i_2, \ldots, i_n$.
- measure qubit i.

Untyped lambda calculus

Definition. Lambda terms

$$M, N \quad ::= \quad x \quad | \quad MN \quad | \quad \lambda x.M$$

where x ranges over an infinite set of symbols called variables.

Example. *$\lambda x.x$ is a lambda term. It stands for the identity function. $(\lambda x.xy)(\lambda y.yz)$ is a lambda term. x is bound, z is free, and the variable y has both a free and a bound occurrence.*

- Alpha equivalence — terms differ only in the choice of bound variables are considered the same.
- Beta reduction — $(\lambda x.M)N \to M[N/x]$.

Simply Typed lambda calculus

Definition. Types $S, T \quad ::= \quad B \quad | \quad S \to T$, where B ranges over a set of symbols called basic types.

Example. *B and $B \to B$ are types. $B \to B$ is a function type.*

Definition. A type assignment Γ is a function from some finite set of variables to types. Γ is written as $\{x : A, y : B, ...\}$.

Definition. A typing judgement is a triple (Γ, M, T) of a type assignment, a term, and a type, written in the form $\Gamma \vdash M : T$.

Typing rules

Definition. A typing judgement is valid if it follows from typing rules:

$$\frac{x : A \in \Gamma}{\Gamma \vdash x : A}$$

$$\frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

Propositional logic

Definition. Formulas (only the implication fragment)

$$S, T \quad ::= \quad B \quad | \quad S \to T\,,$$

where B ranges over a set of atomic propositions.

Definition. A sequent is a pair (Γ, A), written as $\Gamma \vdash A$, where Γ is a finite multiset of formulas, called a context, and A is a formula. We have three natural ways (derivations) to prove a formula.

$$\frac{A \in \Gamma}{\Gamma \vdash A} \qquad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}.$$

Curry-Howard correspondence

Curry-Howard correspondence [Curry, 1934, Howard, 1995].

- formulas $\leftrightarrow$ types (if atomic propositions $\leftrightarrow$ basic types).
- well-typed lambda terms $\leftrightarrow$ derivations.

Recall definitions: Types $S, T \quad ::= \quad B \quad | \quad S \rightarrow T$, where B ranges over basic types. Formulas $S, T \quad ::= \quad B \quad | \quad S \rightarrow T$, where B ranges over atomic propositions.

Proofs-as-terms

Example. *Let* $\Gamma = \{A, B\}$, *we have two derivations of* B*:*

$$\frac{B \in \Gamma}{\Gamma \vdash B} \qquad\qquad \frac{\dfrac{A \in \Gamma}{\Gamma \vdash A} \qquad \dfrac{\dfrac{B \in \Gamma}{\Gamma, A \vdash B}}{\Gamma \vdash A \to B}}{\Gamma \vdash B}$$

Label assumptions $\Gamma = \{x : A, y : B\}$, *and put lambda terms in:*

$$\frac{y : B \in \Gamma}{\Gamma \vdash y : B} \qquad\qquad \frac{\dfrac{x : A \in \Gamma}{\Gamma \vdash x : A} \qquad \dfrac{\dfrac{y : B \in \Gamma}{\Gamma, x : A \vdash y : B}}{\Gamma \vdash \lambda x.y : A \to B}}{\Gamma \vdash (\lambda x.y)x : B}$$

Quantum lambda calculus

Definition. The types of Quantum Lambda Calculus (QLC) [Valiron, 2004] are defined by

$$A, B \quad ::= \quad \top \quad | \quad \mathit{bit} \quad | \quad \mathit{qubit} \quad | \quad A \multimap B \quad | \quad A \otimes B \quad | \quad !A.$$

Definition. The terms of QLC are defined by

$$\begin{array}{rl} M, N, P \quad ::= & x \quad | \quad \lambda x.M \quad | \quad MN \\ | & * \quad | \; \langle M, N \rangle \; | \quad \mathit{let}\ \langle x, y \rangle = M\ \mathit{in}\ N \quad | \quad \mathit{let}\ * \ = M\ \mathit{in}\ N \\ | & 0 \quad | \quad 1 \quad | \quad \mathit{if}\ P\ \mathit{then}\ M\ \mathit{else}\ N \\ | & \mathit{new} \; | \; \mathit{meas} \; | \quad U \quad | \quad q \end{array}$$

U ranges over a given set of symbols called <u>circuit constants</u>, and q ranges over a given infinite set of symbols called <u>quantum names</u>.

Type system

Definition. We define a relations $<:$ on types, called subtyping relations satisfying ...

$$\frac{}{!\,A \;<:\; A} \qquad \frac{!\,A <: B}{!A <: !\,B} \qquad \frac{A_1 <: B_1 \quad A_2 <: B_2}{(A_1 \otimes A_2) \;<:\; (B_1 \otimes B_2)}\otimes$$

Definition. A typing judgement is a quadruple $\Gamma; Q \vdash M : T$, where Γ is a type assignment, Q is a finite set of quantum names, called quantum context, M is a term, and T is a type. A typing judgement is valid if it follows from ...

$$\frac{\Gamma, x : A; Q \vdash M : B}{\Gamma; Q \vdash \lambda x.M : A \multimap B}\lambda_1 \qquad \frac{!\Delta, x : A; \emptyset \vdash M : B}{!\Delta; \emptyset \vdash \lambda x.M : !(A \multimap B)}\lambda_2$$

Operational semantics

Definition. A *quantum closure* is a triple $[Q, L, M]$ where

- Q is an n-qubit state.
- L is a list of n distinct quantum names, written as $|q_1 q_2 ... q_n\rangle$.
- M is a QLC term.

Definition. $[Q, L, M] \rightarrow_p [Q', L', M']$ is a single-step reduction of quantum closures that takes place with probability p. Some reduction rules:

$$[Q, |q_1, ...q_n\rangle, U \langle q_{j_1}, ..., q_{j_n}\rangle] \rightarrow_1 [Q', |q_1, ...q_n\rangle, \langle q_{j_1}, ..., q_{j_n}\rangle]$$

$$[\alpha |Q_0\rangle + \beta |Q_1\rangle, |q_1, ...q_n\rangle, \ \mathit{meas}\ q_i] \rightarrow_{|\alpha|^2} [Q_0, |q_1, ...q_n\rangle, 0]$$

$$[Q, |q_1, ...q_n\rangle, \ \mathit{new}\ 0] \rightarrow_1 [Q \otimes |0\rangle, |q_1, ...q_n, q_{n+1}\rangle, q_{n+1}]$$

Proto-Quipper

Definition. The types of Proto-Quipper (PQ) [Ross, 2015] are defined by

$$A, B \quad ::= \quad \dots \quad | \quad Circ(T, U)$$

where $T, U \quad ::= \quad \top \quad | \quad qubit \quad | \quad T \otimes U$.

Definition. The terms of PQ are defined by $M, N, P \quad ::= \quad \dots$

$$| \quad (t, C, M) \quad | \quad rev \quad | \quad unbox \quad | \quad box^T \quad | \quad q$$

where $t, u \quad ::= \quad * \quad | \quad q \quad | \quad \langle t, u \rangle$. Here, C ranges over a set $\mathbb{C}$ of circuit constants.

Type system

Type system of PQ is similar to the one of QLC.

- Almost the same subtyping rules as in QLC.
- One more typing rule

$$\frac{!\Delta; Q_1 \vdash t : T \quad !\Delta; Q_2 \vdash M : U \quad In(C) = Q_1 \quad Out(C) = Q_2}{!\Delta, \emptyset \vdash (t, C, M) : Circ(T, U)}\,circ$$

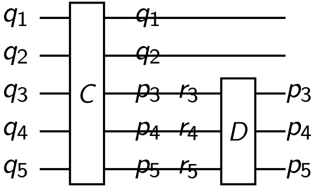
Labelled quantum circuit

Definition. Consider the category $\mathbb{LC}$ with objects finite sequences of distinct quantum names, and morphisms between $\langle q_1, q_2, ..., q_n \rangle$ and $\langle p_1, p_2, ..., p_n \rangle$ are quantum circuits. We call the morphisms labelled quantum circuits.

- Partial tensor. If two objects s_1 and s_2 are disjoint (seen as sets), define $s_1 \otimes s_2 = s_1, s_2$. The tensor product of morphisms is just the tensor product of labelled quantum circuits.
- Partial composition. The purpose of partial composition $D \circ' C$ is to append circuit D to C, when $dom(D) \neq cod(C)$.

Partial compostition

Example.



Operational semantics

Definition. A circuit closure is a pair $[C, M]$ where

- C is a labelled quantum circuit.
- M is a lambda term.

Definition. The one-step reduction relation, written as $\to$, is defined on circuit closures by rules such as

$$[C, rev\ (i, D, o)] \to [C, (o, D^{-1}, i)]$$

$$\frac{s \in Obj(\mathbb{L}\mathbb{C}) \quad len(s) = len(T) \quad t = Term(s, T)}{[C, box^T(M)] \to [C, (t, Id_s, Mt)]}\, box$$

$$\frac{b = bind(V, u) \quad b' = bind(dom(D), cod(D))}{[C, (unbox(u, D, u'))V] \to [D \circ' (b \circ' C), b'(u')]}\, unbox$$

Problems

Extension of PQ

- PQ extends a minimal version of QLC. How to incorporate additional features of the QLC in PQ, such as coproducts, recursion, and measurement.
- We will extend the notion of quantum circuit to also include classical wires, which hold a classical bit at circuit execution time.
- It is no longer the case that all circuits are reversible. It will be necessary to extend the type system to be aware of this fact.

Type inference

- Valiron described [Valiron, 2004] a type inference algorithm for QLC.
- Type inference is useful because it is tedious for programmers to write type annotations.
- But it is not known whether it can be done <u>efficiently</u>.
- An interesting open problem is to find efficient type inference algorithm for QLC and/or PQ.

Imperative style

A typical quantum program reads

$$\begin{aligned}&let\ (x, y) = Cnot(x, y)\ in\\&let\ y = Hy\ in\\&let\ (y, x) = subroutine(y, x)\ in\\&(x, y)\end{aligned}$$

In Quipper, we can use a simpler ‘imperative style’

$$\begin{aligned}&Cnot(x, y);\\&Hy;\\&subroutine(y, x);\end{aligned}$$

No theoretical foundation for such a syntax yet.

Imperative style

It gets complicated when

- Functions produce 'garbage' (ancilla qubits to hold intermediate results of the computation).
- Function have some imperative and some non-imperative arguments. For example,

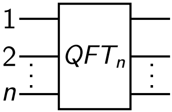
$$let\ x' = Cnot(x, y).$$

- Functions are in the body of loops.

$$loop\ 10\ (\ \lambda\,\langle x, y\rangle\ .let\ x = H\,x\ in\,\langle x, y\rangle\ ).$$

Parameter-state distinction

Consider a circuit defined on n qubits, such as the QFT_n



It would be natural to define a circuit family as a lambda term

$$QFT : (n : \ Nat) \multimap \otimes^n qubit \multimap \otimes^n qubit.$$

But this requires dependent types.

- In QFT_n, n is a parameter that is known when generating circuit.
- In Hq, q refers to a quantum state that is known when executing the circuit H.

References

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

Howard, W. A. (1995).
The formulae-as-types notion of construction.

Joyal, A. and Street, R. (1991).
The geometry of tensor calculus, i.
Advances in Mathematics, 88(1):55–112.

Knill, E. (1996).
Conventions for quantum pseudocode.
Technical report, Citeseer.

Ross, N. J. (2015).
Algebraic and logical methods in quantum computation.
arXiv preprint arXiv:1510.02198.

Valiron, B. (2004).

Questions?