Recursive types for linear/non-linear quantum programming

Vladimir Zamdzhiiev

Université de Lorraine, CNRS, Inria, LORIA, F 54000 Nancy, France

Joint work with Michael Mislove and Bert Lindenhovius

Dagstuhl Seminar on Quantum Programming Languages
17 September 2018
Proto-Quipper-M

- We consider adding recursive types to *Proto-Quipper-M*.
- Original language developed by Francisco Rios and Peter Selinger.
- Language is equipped with formal denotational and operational semantics.
- Primary application is in quantum computing, but the language can describe arbitrary string diagrams.
- In prior work, we described an abstract model of the language and added recursion.
Circuit Model

The language is used to describe *families* of morphisms of an arbitrary small symmetric monoidal category, which we denote $\mathcal{M}$.

**Remark**

$\mathcal{M}$ could also be a category of string diagrams which is freely generated.
Example

Shor’s algorithm for integer factorization may be seen as an infinite family of quantum circuits – each circuit is a procedure for factorizing an \( n \)-bit integer, for a fixed \( n \).

\[
\begin{array}{c}
|x_n\rangle \\
|x_{n-1}\rangle \\
\vdots \\
|x_2\rangle \\
|x_1\rangle \\
\end{array}
\begin{array}{c}
\text{QFT}_{n-1} \\
\end{array}
\begin{array}{c}
R_{\pi/2}^{n-1} \\
R_{\pi/2}^{n-2} \\
\vdots \\
R_{\pi/2}^{n/2-1} \\
H \\
\end{array}
\begin{array}{c}
|y_1\rangle \\
|y_2\rangle \\
\vdots \\
|y_{n/2}\rangle \\
|y_{n}\rangle \\
\end{array}
\]

**Figure:** Quantum Fourier Transform on \( n \) qubits (subroutine in Shor’s algorithm).\(^1\)

\(^1\)Figure source: https://commons.wikimedia.org/w/index.php?curid=14545612
Main difficulty is on the denotational side.

How can we copy/discard intuitionistic recursive types?
- A list of qubits should be *linear* – cannot copy/discard.
- A list of natural numbers should be *intuitionistic* – can implicitly copy/discard.

For the rest of the talk we focus on the linear/non-linear type structure.

How do we design a linear/non-linear FPC?
The Basic (non-recursive) Types

Types

\[ A, B ::= \alpha | 0 | A + B | I | A \otimes B | A \rightarrow B | !A | \text{Circ}(T, U) \]

Intuitionistic types

\[ P, R ::= 0 | P + R | I | P \otimes R | !A | \text{Circ}(T, U) \]

M-types

\[ T, U ::= \alpha | I | T \otimes U \]

Remark

\[ \text{Circ}(T, U) \cong ! (T \rightarrow U). \]
Denotational Model

A model of PQM is induced by a Linear/Non-Linear (LNL) model\(^2\):

- A cartesian closed category \( \mathbf{V} \).
- A symmetric monoidal closed category \( \mathbf{C} \).
- A symmetric monoidal adjunction:

\[
\begin{array}{c}
\mathbf{V} \\
\downarrow \\
\mathbf{C}
\end{array}
\]

\[
\begin{array}{c}
\downarrow \quad F \\
\mathbf{V} \\
\downarrow \\
G \\
\mathbf{C}
\end{array}
\]

together with some additional data which is irrelevant for this talk.

Remark

An LNL model is a model of Intuitionistic Linear Logic.

\(^2\)Nick Benton. *A mixed linear and non-linear logic: Proofs, terms and models.* CSL'94
Copying and discarding of intuitionistic types

In PQM, any type $A$ is interpreted as an object $[A] \in C$.

**Theorem**

For any intuitionistic type $P$, there exists a canonical isomorphism $\alpha_P : [P] \to F(P)$.

Next, define copy and discard morphisms for each intuitionistic type $P$:

$$
\Diamond_P := [P] \xrightarrow{\alpha_P} F(P) \xrightarrow{F \langle id, id \rangle} F(P \times P) \xrightarrow{\cong} F(P) \otimes F(P) \xrightarrow{\alpha_P^{-1} \otimes \alpha_P^{-1}} [P] \otimes [P]
$$

$$
\Delta_P := [P] \xrightarrow{\alpha_P} F(P) \xrightarrow{F(id, id)} F(P \times P) \xrightarrow{\cong} F(P) \otimes F(P) \xrightarrow{\alpha_P^{-1} \otimes \alpha_P^{-1}} [P] \otimes [P]
$$
Adding Recursive Types

Type Variables $X, Y$

Types $A, B ::= X | \alpha | A + B | I | A \otimes B | A \rightarrow B | !A | \text{Circ}(T, U) | \mu X. A$

Intuitionistic types $P, R ::= X | P + R | I | P \otimes R | !A | \text{Circ}(T, U) | \mu X. P$

M-types $T, U ::= \alpha | I | T \otimes U$

Remark

These types are accompanied by some formation rules, which we omit.

Design Choice: Two kinds of type variables – intuitionistic and linear? Or just one kind (like above)?
Some useful recursive types

Example
Nat ≜ \mu X. I + X \quad \text{(intuitionistic)}

Example
List Nat ≜ \mu X. I + X \otimes \text{Nat} \quad \text{(intuitionistic)}

Example
List Qubit ≜ \mu X. I + X \otimes \text{Qubit} \quad \text{(linear)}
Term level recursion

In FPC, a term-level recursion operator may be defined using fold/unfold maps. The same is true for our language.

**Theorem**

The term-level recursion operator for PQM\(^3\) is now a derived rule. For a given term \(\Phi, z : !A \vdash m : A\), define:

\[
\alpha^z_m \equiv \text{lift fold } \lambda x^1 \mu X.((!X \to !A)) . (\lambda z^1 A . m)(\text{lift (unfold force } x) x)
\]

\[
\text{rec } z^1 A . m \equiv (\text{unfold force } \alpha^z_m) \alpha^z_m
\]

\(^3\)Bert Lindenhovius, Michael Mislove, Vladimir Zamzhiiev: Enriching a Linear/Non-linear Lambda Calculus: A Programming Language for String Diagrams. LICS 2018
A **CPO**-enriched model

1. A **CPO**-symmetric monoidal closed category $\mathbf{C}$ such that $\mathbf{C}$ has finite **CPO**-coproducts.

2. A **CPO**-symmetric monoidal adjunction:

   $\mathbf{CPO} \xrightarrow{F} \mathbf{C}$

3. The category $\mathbf{C}$ is **CPO**$_\perp$-enriched and has $\omega$-colimits.

   together with some additional data which is irrelevant for this talk.

**Remark**

1. and 3. imply $\mathbf{C}$ has a zero object and we can solve recursive domain equations.
Interpretation of recursive types

Interpreting recursive types amounts to finding initial (final) (co)algebras of certain endofunctors.

Lemma (Adámek)

Let \( C \) be a category with an initial object \( \emptyset \) and let \( T : C \to C \) be an endofunctor. Assume further that the following \( \omega \)-diagram

\[
\emptyset \xrightarrow{\iota} T\emptyset \xrightarrow{T\iota} T^2\emptyset \xrightarrow{T^2\iota} \ldots
\]

has a colimit and \( T \) preserves it. Then, the induced isomorphism is the initial \( T \)-algebra.

Corollary

In a symmetric monoidal closed category with finite coproducts and \( \omega \)-colimits, any endofunctor composed from constants, \( \otimes \) and \( + \) has an initial algebra.
Embedding-projection pairs

**Problem:** How do we interpret recursive types which also contain ! and →?

**Textbook Solution:** CPO-enrichment and embedding-projection pairs.

**Definition**
Given a CPO-enriched category $\mathbf{C}$, an embedding-projection pair is a pair of morphisms $e : A \to B$ and $p : B \to A$, such that $p \circ e = \text{id}$ and $e \circ p \leq \text{id}$.

**Theorem**
If $e$ is an embedding, then it has a unique projection, which we denote $e^*$.

**Definition**
The subcategory of $\mathbf{C}$ with the same objects, but whose morphisms are embeddings is denoted $\mathbf{C}_e$. 
Interpretation of recursive types (contd.)

Theorem (Smyth and Plotkin)

If $T : C \to D$ is a CPO-enriched functor and $C$ has $\omega$-colimits, then $T$ preserves $\omega$-colimits of embeddings. In other words, the restriction $T_e : C_e \to D_e$ is $\omega$-continuous.

Theorem

In our categorical model, any CPO-endofunctor $T : C \to C$ has an initial $T$-algebra, whose inverse is a final $T$-coalgebra.

Remark

The above theorem follows directly from results in Fiore’s PhD thesis.
Interpretation of types in FPC

Definition
Let $\mathcal{C} := \mathcal{C}^\text{op} \times \mathcal{C}$. An object of $\mathcal{C}$ is called *symmetric* if it is of the form $(A, A)$. The subcategory of symmetric objects and morphisms of the form $(e^\text{op}, e)$ is denoted $\mathcal{C}_{se}$. In FPC, a type with a free type variable $X \vdash A$ is interpreted as a *symmetric* CPO-enriched endofunctor

$$\llbracket X \vdash A \rrbracket : \mathcal{C} \to \mathcal{C}$$

which therefore restricts to an endofunctor

$$\llbracket X \vdash A \rrbracket_{se} : \mathcal{C}_{se} \to \mathcal{C}_{se}.$$

Remark
$\mathcal{C}_{se} \cong \mathcal{C}_e$. Thus, $\llbracket X \vdash A \rrbracket_{se}$ may be seen as an endofunctor on $\mathcal{C}_e$. 
A term $X; \Gamma \vdash m : A$ is interpreted as a *family* of morphisms in $\mathbf{C}$ parameterised by the objects $B$ of $\mathbf{C}$.

$$\llbracket X; \Gamma \vdash m : A \rrbracket = \{ \Pi_2 \circ \llbracket X \vdash \Gamma \rrbracket (B, B) \xrightarrow{[X; \Gamma \vdash m : A]_B} \Pi_2 \circ \llbracket X \vdash A \rrbracket (B, B) \mid B \in \text{Ob}(\mathbf{C}) \}$$
Recursive types for PQM

Using the data from our categorical model:

\[
\vdash
\]

we may solve all required recursive domain equations and interpret all required type expressions \( \Theta \vdash A \) as functors \([\Theta \vdash A] : \check{C} \rightarrow \check{C}\).

Remark

This follows easily using well-known results from the literature.

Problem: How do we copy/discard the (recursive) intuitionistic types?
Final notations

Definition
Given two CPO-enriched categories $\mathbf{C}$ and $\mathbf{D}$ and a CPO-functor $T : \mathbf{C} \to \mathbf{D}$, a pre-embedding in $\mathbf{C}$ w.r.t $T$ is a morphism $f \in \mathbf{C}$, s.t. $Tf$ is an embedding in $\mathbf{D}$.

Definition
Let $\mathbf{PE}$ be the subcategory of $\mathbf{CPO}$ with the same objects, but whose morphisms are pre-embeddings w.r.t $F$ in our model.

Example
Every embedding in $\mathbf{CPO}$ is a pre-embedding, but not vice versa. The empty map $\iota : \emptyset \to X$ is a pre-embedding (w.r.t to $F$ in our model), but not an embedding.

Remark
$\Pi_2 : \mathcal{C}_{se} \to \mathcal{C}_e$ is an isomorphism with inverse $D : \mathcal{C}_e \to \mathcal{C}_{se}$ given by

\[
D(A) = (A, A) \\
D(e^\text{op}) = (e^\text{op}, e)
\]
Recall that in PQM with basic types, the basis for copying and discarding is given by the canonical iso (for $P$ intuitionistic):

$$\alpha_P : [P] \overset{\simeq}{\rightarrow} F(P)$$

**Problem:** How do we generalise this to work with recursive types, where the interpretation of a type is now a functor?
Main conjecture

Conjecture

For any intuitionistic type $\Theta \vdash P$, there exists a natural isomorphism

$$\alpha_{\Theta \vdash P} : \Pi_2 \circ [\Theta \vdash P]_{se} \circ D^{\times n} \circ F^{\times n} \Rightarrow F \circ (\Theta \vdash P)$$

diagrammatically:

\[\begin{array}{c}
\text{PE}^n \xrightarrow{[\Theta \vdash P]} \tilde{C}_{se}^n \xrightarrow{[\Theta \vdash P]_{se}} \tilde{C}_{se}^n \xleftarrow{F^{\times n}} \tilde{C}_e^n \xrightarrow{D^{\times n}} \tilde{C}_{se}^n \\
\text{PE}^n \xrightarrow{(\Theta \vdash P)} \tilde{C}_{se}^n \xleftarrow{F^{\times n}} \tilde{C}_e^n \xrightarrow{D^{\times n}} \tilde{C}_{se}^n \\
\end{array}\]

or equivalently

\[\begin{array}{c}
\text{PE}^n \xrightarrow{[\Theta \vdash P]} \tilde{C}_{se}^n \xrightarrow{[\Theta \vdash P]_{se}} \tilde{C}_{se}^n \xleftarrow{F^{\times n}} \tilde{C}_e^n \xrightarrow{D^{\times n}} \tilde{C}_{se}^n \\
\text{PE}^n \xrightarrow{(\Theta \vdash P)} \tilde{C}_{se}^n \xleftarrow{F^{\times n}} \tilde{C}_e^n \xrightarrow{D^{\times n}} \tilde{C}_{se}^n \\
\end{array}\]
Interpretation of terms in PQM with recursive types

Recall, that in FPC the interpretation of the term $X; \Gamma \vdash m : A$ is parameterised by the objects $B$ of $\mathbf{C}$ (where $\mathbf{C}$ is some Kleisli category or category of partial maps).

$$\llbracket X; \Gamma \vdash m : A \rrbracket = \{ \Pi_2 \circ \llbracket X \vdash \Gamma \rrbracket (B, B) \xrightarrow{\llbracket X; \Gamma \vdash m : A \rrbracket_B} \Pi_2 \circ \llbracket X \vdash A \rrbracket (B, B) \mid B \in \text{Ob}(\mathbf{C}) \}$$

For PQM with recursive types, define

$$\llbracket X; \Gamma \vdash m : A \rrbracket = \{ \Pi_2 \circ \llbracket X \vdash \Gamma \rrbracket (FY, FY) \xrightarrow{\llbracket X; \Gamma \vdash m : A \rrbracket_Y} \Pi_2 \circ \llbracket X \vdash A \rrbracket (FY, FY) \mid Y \in \text{Ob}(\text{CPO}) \}$$

so the interpretation is parameterised by the objects of $\text{CPO}$ (which is intuitionistic).
Given a term $X; \Gamma \vdash m : P$ where $P$ is intuitionistic, then:

$$\llbracket X; \Gamma \vdash m : P \rrbracket = \{ \Pi_2 \circ \llbracket X \vdash \Gamma \rrbracket (FY, FY) \xrightarrow{\llbracket X; \Gamma \vdash m : A \rrbracket_Y} \Pi_2 \circ \llbracket X \vdash P \rrbracket (FY, FY) \mid Y \in \text{Ob}(\text{CPO}) \}$$

Observe that

$$\Pi_2 \circ \llbracket X \vdash P \rrbracket (FY, FY) = \Pi_2 \circ \llbracket X \vdash P \rrbracket_{se} \circ D \circ F(Y) \cong F \circ (\Theta \vdash P)(Y)$$

due to the main conjecture. Hence, we may copy or discard the required types / objects.
Let $M_*$ be the free $\text{CPO}_{\bot!}$-enrichment of $M$ and $\overline{M}_* = [M_*^\text{op}, \text{CPO}_{\bot!}]$ be the associated enriched functor category.

**Concrete model**

![Diagram illustrating the model](image)

**Remark**

If $M = 1$, then the above model degenerates to the left vertical adjunction, which is a model of FPC.
Future work

- Finish the proofs (conjecture + soundness).
- Investigate computational adequacy.
Future work

- Finish the proofs (conjecture + soundness).
- Investigate computational adequacy.
- Abstract model (i.e. do not assume CPO-enrichment)?
Thank you for your attention!