# Recursive types for linear/non-linear quantum programming

Vladimir Zamdzhiev

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 **M**.

### Remark

**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*.

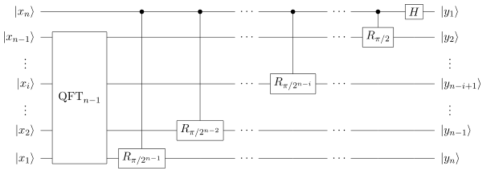



Figure: Quantum Fourier Transform on *n* qubits (subroutine in Shor's algorithm).[1]

[1] Figure source: `https://commons.wikimedia.org/w/index.php?curid=14545612`

# Long story short

- 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 | $A, B$ | $::=$ | $\alpha \mid 0 \mid A + B \mid \mathrm{I} \mid A \otimes B \mid A \multimap B \mid !A \mid \mathrm{Circ}(T, U)$ |
| Intuitionistic types | $P, R$ | $::=$ | $0 \mid P + R \mid \mathrm{I} \mid P \otimes R \mid !A \mid \mathrm{Circ}(T, U)$ |
| M-types | $T, U$ | $::=$ | $\alpha \mid \mathrm{I} \mid T \otimes U$ |

## Remark

$Circ(T, U) \cong !(T \multimap U)$.

# 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:

$$\mathbf{V} \underset{G}{\overset{F}{\rightleftarrows}} \mathbf{C} \quad (\bot)$$

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

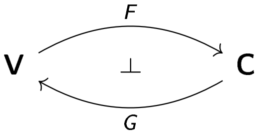
**Remark**

*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 interepreted as an object $[\![A]\!] \in \mathbf{C}$.

**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 $P$:

$$\diamond_P := [\![P]\!] \xrightarrow{\alpha_P} F(\!|P|\!) \xrightarrow{F1} F1 \xrightarrow{\cong} I$$

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

# Adding Recursive Types

| | | | |
|---|---|---|---|
| Type Variables | $X, Y$ | | |
| Types | $A, B$ | $::=$ | $X \mid \alpha \mid A + B \mid I \mid A \otimes B \mid A \multimap B \mid !A \mid \mathrm{Circ}(T, U) \mid \mu X.A$ |
| Intuitionistic types | $P, R$ | $::=$ | $X \mid P + R \mid I \mid P \otimes R \mid !A \mid \mathrm{Circ}(T, U) \mid \mu X.P$ |
| M-types | $T, U$ | $::=$ | $\alpha \mid I \mid T \otimes U$ |

## 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**

$\mathsf{Nat} \equiv \mu X.I + X$ (intuitionistic)

**Example**

$\mathsf{List\ Nat} \equiv \mu X.I + X \otimes \mathsf{Nat}$ (intuitionistic)

**Example**

$\mathsf{List\ Qubit} \equiv \mu X.I + X \otimes \mathsf{Qubit}$ (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:*

$$\alpha_m^z \equiv \mathit{lift}\ \mathit{fold}\ \lambda x^{!\mu X.(!X \multimap A)}.(\lambda z^{!A}.m)(\mathit{lift}\ (\mathit{unfold}\ \mathit{force}\ x)x)$$

$$\mathit{rec}\ z^{!A}.m \equiv (\mathit{unfold}\ \mathit{force}\ \alpha_m^z)\alpha_m^z$$

[3] Bert Lindenhovius, Michael Mislove, Vladimir Zamdzhiev: 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 **C** such that **C** has finite **CPO**-coproducts.
2. A **CPO**-symmetric monoidal adjunction:

$$\mathbf{CPO} \underset{\mathbf{C}(I,-)}{\overset{F}{\rightleftarrows}} \mathbf{C}, \qquad F \dashv \mathbf{C}(I,-)$$

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

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

### Remark

*1. and 3. imply* **C** *has a zero object and we can solve recursive domain equations.*

# Interpretation of recursive types

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

**Lemma (Adámek)**

*Let $\mathbf{C}$ be a category with an initial object $\emptyset$ and let $T : \mathbf{C} \to \mathbf{C}$ be an endofunctor. Assume further that the following $\omega$-diagram*

$$\emptyset \xrightarrow{\iota} T\emptyset \xrightarrow{T\iota} T^2\emptyset \xrightarrow{T^2\iota} \cdots$$

*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 $\multimap$?
**Textbook Solution: CPO**-enrichment and embedding-projection pairs.

## Definition

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

## Theorem

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

## Definition

The subcategory of **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 $T : \mathbf{C} \to \mathbf{D}$ is a* **CPO***-enriched functor and* $\mathbf{C}$ *has $\omega$-colimits, then $T$ preserves $\omega$-colimits of embeddings. In other words, the restriction $T_e : \mathbf{C}_e \to \mathbf{D}_e$ is $\omega$-continuous.*

**Theorem**

*In our categorical model, any* **CPO***-endofunctor $T : \mathbf{C} \to \mathbf{C}$ 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 $\breve{\mathbf{C}} := \mathbf{C}^{\mathrm{op}} \times \mathbf{C}$. An object of $\breve{\mathbf{C}}$ is called *symmetric* if it is of the form $(A, A)$. The subcategory of symmetric objects and morphisms of the form $(e^{*^{\mathrm{op}}}, e)$ is denoted $\breve{\mathbf{C}}_{se}$.

In FPC, a type with a free type variable $X \vdash A$ is interpreted as a *symmetric* **CPO**-enriched endofunctor

$$[\![X \vdash A]\!] : \breve{\mathbf{C}} \to \breve{\mathbf{C}}$$

which therefore restricts to an endofunctor

$$[\![X \vdash A]\!]_{se} : \breve{\mathbf{C}}_{se} \to \breve{\mathbf{C}}_{se} \ .$$

## Remark

$\breve{\mathbf{C}}_{se} \cong \mathbf{C}_e$. *Thus,* $[\![X \vdash A]\!]_{se}$ *may be seen as an endofunctor on* $\mathbf{C}_e$.

# Interpretation of terms in FPC

A term $X; \Gamma \vdash m : A$ is interpreted as a *family* of morphisms in $\mathbf{C}$ parameterised by the objects $B$ of $\mathbf{C}$.

$$[\![X; \Gamma \vdash m : A]\!] = \{\Pi_2 \circ [\![X \vdash \Gamma]\!](B, B) \xrightarrow{[\![X;\Gamma \vdash m:A]\!]_B} \Pi_2 \circ [\![X \vdash A]\!](B, B) \mid B \in \mathrm{Ob}(\mathbf{C})\}$$

Using the data from our categorical model:

$$\mathbf{V} \underset{G}{\overset{F}{\underset{\perp}{\rightleftarrows}}} \mathbf{C}$$

we may solve all required recursive domain equations and interpret all required type expressions $\Theta \vdash A$ as functors $[\![\Theta \vdash A]\!] : \breve{\mathbf{C}}^n \to \breve{\mathbf{C}}$.

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

### Definition

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

### Example

Every embedding in **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

$\Pi_2 : \breve{\mathbf{C}}_{se} \to \mathbf{C}_e$ *is an isomoprhism with inverse* $D : \mathbf{C}_e \to \breve{\mathbf{C}}_{se}$ *given by*

$$D(A) = (A, A)$$

$$D(e) = (e^{*^{op}}, e)$$

# Copying and discarding?

Recall that in PQM with basic types, the basis for copying and discarding is given by the canonical iso (for $P$ intuitionistic):

$$\alpha_P : [\![P]\!] \xrightarrow{\cong} F(\!|P|\!)$$

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

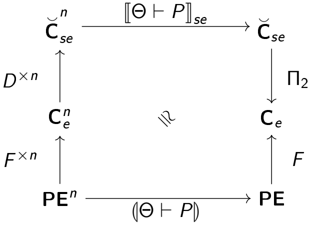
# Main conjecture

## Conjecture

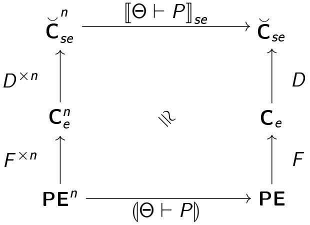
*For any intuitionistic type $\Theta \vdash P$, there exists a natural isomorphism*

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

*diagrammatically:*




*or equivalently*



# Intherpretation 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).

$$[\![X; \Gamma \vdash m : A]\!] = \{\Pi_2 \circ [\![X \vdash \Gamma]\!](B, B) \xrightarrow{[\![X;\Gamma \vdash m:A]\!]_B} \Pi_2 \circ [\![X \vdash A]\!](B, B) \mid B \in \mathrm{Ob}(\mathbf{C})\}$$

For PQM with recursive types, define

$$[\![X; \Gamma \vdash m : A]\!] = \{\Pi_2 \circ [\![X \vdash \Gamma]\!](FY, FY) \xrightarrow{[\![X;\Gamma \vdash m:A]\!]_Y} \Pi_2 \circ [\![X \vdash A]\!](FY, FY) \mid Y \in \mathrm{Ob}(\mathbf{CPO})\}$$

so the interpretation is parameterised by the objects of **CPO** (which is intuitionistic).

# Copying and discarding!

Given a term $X;\Gamma \vdash m : P$ where $P$ is intuitionistic, then:

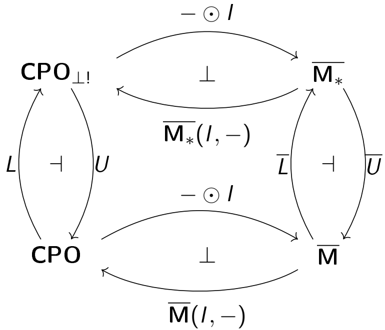
$$[\![X;\Gamma \vdash m : P]\!] = \{\Pi_2 \circ [\![X \vdash \Gamma]\!](FY, FY) \xrightarrow{[\![X;\Gamma \vdash m:A]\!]_Y} \Pi_2 \circ [\![X \vdash P]\!](FY, FY) \mid Y \in \mathrm{Ob}(\mathbf{CPO})\}$$

Observe that

$$\Pi_2 \circ [\![X \vdash P]\!](FY, FY) = \Pi_2 \circ [\![X \vdash P]\!]_{se} \circ D \circ F(Y) \cong F \circ (\!|\Theta \vdash P|\!)(Y)$$

due to the main conjecture. Hence, we may copy or discard the required types / objects.

Let $\mathbf{M}_*$ be the free $\mathbf{CPO}_{\perp!}$-enrichment of $\mathbf{M}$ and $\overline{\mathbf{M}_*} = [\mathbf{M}_*^{op}, \mathbf{CPO}_{\perp!}]$ be the associated enriched functor category.




## Remark

*If* $\mathbf{M} = \mathbf{1}$*, 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!