A categorical model for a quantum programming language with recursive types

Bert Lindenhovius     Michael Mislove     Vladimir Zamdzhiyev

Department of Computer Science
Tulane University

Foundational Methods in Computer Science
University of Ottawa
10 June 2017
We will consider a functional programming language called *Proto-Quipper-M*. Language and model developed by Francisco Rios and Peter Selinger. Language is equipped with formal denotational and operational semantics. Type-safe. Model is computationally sound and adequate. Language can describe families of morphisms from any symmetric monoidal category. Primary application is in quantum computing. Model does not support general recursion. Our contribution is to extend the categorical model so that it supports recursive types and general recursion on the term level.
Proto-Quipper-M is used to describe families of morphisms of an arbitrary, but fixed, symmetric monoidal category, which we denote $Q$.

Example

If $Q = \text{FdCStar}$, the category of finite-dimensional $C^*$-algebras and completely positive maps, then a program in our language is a family of quantum circuits.

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).](https://commons.wikimedia.org/w/index.php?curid=14545612)
Syntax of Proto-Quipper-M

The type system is given by:

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

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

Simple Q-types
\[ T, U ::= \alpha | I | T \otimes U \]

The term language is given by:

Terms
\[ M, N ::= x | I | c | \text{let } x = M \text{ in } N \]
\[ | \square_A M | \text{left}_{A,B} M | \text{right}_{A,B} M | \text{case } M \text{ of } \{ \text{left } x \rightarrow N | \text{right } y \rightarrow P \} \]
\[ | * | M; N | \langle M, N \rangle | \text{let } \langle x, y \rangle = M \text{ in } N | \lambda x^A.M | MN \]
\[ | \text{lift } M | \text{force } M | \text{box}_T M | \text{apply}(M, N) | \langle \bar{l}, C, \bar{l}' \rangle \]

Label tuples
\[ \bar{l}, \bar{k} ::= I | * | \langle \bar{l}, \bar{k} \rangle \]

Values
\[ V, W ::= x | I | c | \text{left}_{A,B} V | \text{right}_{A,B} V | * | \langle V, W \rangle | \lambda x^A.M | \text{lift } M | \langle \bar{l}, C, \bar{l}' \rangle \]
Families Construction

The following construction is well-known.

**Definition**

Given a category $\mathsf{C}$, we define a new category $\mathsf{Fam}(\mathsf{C})$:

- Objects are pairs $(X, A)$ where $X$ is a discrete category and $A : X \to \mathsf{C}$ is a functor.
- A morphism $(X, A) \to (Y, B)$ is a pair $(f, \phi)$ where $f : X \to Y$ is a functor and $\phi : A \to B \circ f$ is a natural transformation.
- Composition of morphisms is given by: $(g, \psi) \circ (f, \phi) = (g \circ f, \psi f \circ \phi)$.

**Remark**

$\mathsf{Fam}(\mathsf{C})$ is the free coproduct completion of $\mathsf{C}$ and as a result has all small coproducts.

**Proposition (Rios & Selinger 2017)**

*If $\mathsf{C}$ is a symmetric monoidal closed and product-complete category, then $\mathsf{Fam}(\mathsf{C})$ is a symmetric monoidal closed category.*
Categorical Model

Definition
A categorical model of Proto-Quipper-M, for a fixed symmetric monoidal category $Q$, is given by the following data:

- A symmetric monoidal closed and product-complete category $M$.
- A full embedding $Q \hookrightarrow M$.
- A symmetric monoidal closed category $Fam(M)$ which we will refer to as $Fam$.
- A symmetric monoidal adjunction:

$$\begin{array}{ccc}
Set & \cong & Fam \\
\downarrow & & \downarrow \\
Fam(I, -) & \cong & Fam
\end{array}$$

where

$$F(X) = (X, l_X), \quad \text{where } l_X(x) = l$$
$$F(f) = (f, \iota), \quad \text{where } \iota_X = \text{id}_I.$$

Remark
For any symmetric monoidal category $Q$, we can set $M := [Q^{op}, Set]$ and then the Yoneda embedding, together with the Day tensor product, satisfy the first two requirements.
Categorical Model

Theorem (Rios & Selinger 2017)

Every categorical model of Proto-Quipper-M is computationally sound and adequate with respect to its operational semantics.

Open Problem

Find a categorical model of Proto-Quipper-M which supports recursion.

Our approach:

- Instead of considering set-indexed families of objects of $M$, we consider dcpo-indexed families.
- Establish a new symmetric monoidal adjunction which is in addition, DCPO-enriched.
- Discover a suitably large class of algebraically compact endofunctors in the new model.
Directed-complete Partial Orders

Definition
A directed set in a partial order \((P, \leq)\) is a nonempty subset \(D \subseteq P\), such that any two elements of \(D\) have an upper bound in \(D\).

Definition
A directed-complete partial order (dcpo) is a partial order \((P, \leq)\) such that every directed set in \(P\) has a supremum. A dcpo is called pointed if it has a least element, which we shall denote with \(\bot\).

Example
The poset \((\mathbb{N}_\top, \sqsubseteq)\), where
\[
\mathbb{N}_\top = \mathbb{N} \cup \{ \top \} \quad \text{and} \\
n \sqsubseteq m \quad \text{iff} \quad n \leq m \text{ or } m = \top
\]
is a (pointed) dcpo.

Definition
Given two dcpo's \(P\) and \(Q\), a function \(f : P \rightarrow Q\) is called *Scott-continuous* if it is monotone and preserves directed suprema: \(f(\sup D) = \sup f(D)\) for every directed \(D \subseteq P\). If \(P\) and \(Q\) are pointed, then \(f\) is called *strict* if it preserves the least element of \(P\): \(f(\bot_P) = \bot_Q\).
Categories of dcpo’s

Theorem
\( \text{DCPO} \), the category whose objects are dcpo’s and morphisms are Scott-continuous functions, is:
- Cartesian closed and enriched over itself
- Complete
- Cocomplete

Theorem
\( \text{DCPO}_\perp \), the category whose objects are pointed dcpo’s and morphisms are strict Scott-continuous functions, is:
- Symmetric monoidal closed and enriched over itself
- Complete
- Cocomplete
- \( \text{DCPO} \)-enriched.

Remark
A standard technique for modelling recursion is developing \( \text{DCPO} \)-enriched categorical models.

Directed Families Construction

Definition
For a given category $\mathbf{C}$, we define $\text{DFam}(\mathbf{C})$ to be the category such that:

- Objects are pairs $(X, A)$ where $X$ is a poset category whose induced poset is a dcpo and $A : X \to \mathbf{C}$ is a functor.
- A morphism $(X, A) \to (Y, B)$ is a pair $(f, \phi)$ where $f : X \to Y$ is a functor whose induced order-preserving function is Scott-continuous and $\phi : A \to B \circ f$ is a natural transformation.
- Composition of morphisms is given by: $(g, \psi) \circ (f, \phi) = (g \circ f, \psi f \circ \phi)$.

Proposition
$\text{DFam}(\mathbf{C})$ has all small coproducts.

Theorem
If $\mathbf{M}$ is a symmetric monoidal closed and product-complete category, then $\text{DFam}(\mathbf{M})$ is symmetric monoidal closed with the following structure:

$\text{id} = (1, I)$

$(X, A) \otimes (Y, B) = (X \times Y, A \otimes B)$, \hspace{1cm} \text{where} \hspace{1cm} (A \otimes B)(x, y) = A(x) \otimes B(y)$

$(X, A) \rightarrow (Y, B) = ([X \to Y], A \rightarrow B)$, \hspace{1cm} \text{where} \hspace{1cm} (A \rightarrow B)(f) = \prod_{x \in X} (A(x) \rightarrow B \circ f(x))$.

If, in addition, $\mathbf{M}$ is complete and cocomplete, then $\text{DFam}(\mathbf{M})$ is also complete and cocomplete.
From now on, we regard $M$ as fixed and we will refer to $\text{Fam}(M)$ and $\text{DFam}(M)$ simply as $\text{Fam}$ and $\text{DFam}$ respectively.

So far, $\text{DFam}$ shares all of the crucial properties of $\text{Fam}$, as required. But, we get in addition:

**Theorem**

$\text{DFam}$ is DCPO-enriched. More specifically, given two morphisms

\[
(f, \phi) \quad \leftrightarrow \quad (g, \psi)
\]

we define an order $\sqsubseteq$ on $\text{DFam}((X, A), (Y, B))$ by

\[
(f, \phi) \sqsubseteq (g, \psi) \iff f \leq g \text{ and } B(f(x) \leq g(x)) \circ \phi_x = \psi_x \text{ for each } x \in X.
\]
A \textbf{DCPO-enriched adjunction}

The essential structure of the categorical model of our language is the symmetric monoidal adjunction:

\[
\begin{array}{c}
\text{Set} \\ \bot \\ \text{Fam} \\
\end{array} \quad \underset{F'}{\xRightarrow{\text{}}}
\begin{array}{c}
\text{Fam} \\ \text{Fam}(l, -) \\
\end{array}
\]

where

\[
F'(X) = (X, l_X), \quad \text{where } l_X(x) = l \\
F'(f) = (f, \iota), \quad \text{where } \iota_x = \text{id}_l.
\]

We can now extend this.

\textbf{Theorem}

\textit{The following diagram}

\[
\begin{array}{c}
\text{DCPO} \\ \bot \\ \text{DFam} \\
\end{array} \quad \underset{F}{\xRightarrow{\text{}}}
\begin{array}{c}
\text{DFam} \\ \text{DFam}(l, -) \\
\end{array}
\]

is a \textbf{DCPO-enriched symmetric monoidal adjunction}, where \( F \) is given by:

\[
F(X) = (X, l_X), \quad \text{where } l_X(x) = l \\
F(f) = (f, \iota), \quad \text{where } \iota_x = \text{id}_l.
\]
Relationship between the models

The two models are related by the following diagram:

\[
\begin{tikzcd}
\text{DCPO} & \\
\downarrow & \downarrow \\
\text{DFam} & \text{Fam}
\end{tikzcd}
\]

\[
\begin{tikzcd}
\text{Set} & \text{DCPO} & \text{DFam} & \text{Fam} \\
\downarrow & \downarrow & \downarrow & \downarrow \\
\text{Set} & \text{Fam} & \text{DFam} & \text{Set}
\end{tikzcd}
\]

where:

- Both \text{Fam} and \text{DFam} are defined over the same category \( M \) which is product-complete and SMC.
- All the adjunctions are symmetric monoidal.
- The top adjunction is also DCPO-enriched.
- The adjunctions commute.
- The functors \( S, S' \) are the obvious inclusion functors.
- The functors \( U, U' \) are the obvious forgetful functors.
- The adjoint functors \( F' \) and \( \text{Fam}(I, -) \) can be equivalently defined by:
  \[
  F' = U' \circ F \circ S \quad \text{and} \quad \text{Fam}(I, -) = U \circ \text{DFam}(I, -) \circ S'
  \]
- If \( M \) is complete and cocomplete, then \text{Fam} and \text{DFam} are also complete and cocomplete.
- Both the bottom and the top adjunctions are a linear/non-linear model as described by Benton.
Remainder of the talk

- We have verified all of the presented results so far.
- What follows is work in progress:
  - We need to verify some remaining details.
  - But we are confident that the constructions are correct.
New categorical model

Our new model of Proto-Quipper-M is given by the DCPO-enriched symmetric monoidal adjunction:

\[
\begin{array}{c}
\text{DCPO} \\
\downarrow
\end{array}
\xrightarrow{F}
\begin{array}{c}
\text{DFam} \\
\text{DFam}(l, -)
\end{array}
\]

**Problem:** Neither the old model, nor the new one can interpret recursive types.

**Solution:** Consider *algebraic compactness*, which is a well-known approach to modelling recursive types.
Definition
An endofunctor $F : \mathbf{C} \to \mathbf{C}$ is called algebraically compact, if $F$ has an initial $F$-algebra $(A, \alpha)$ and a final $F$-coalgebra $(B, \beta)$ and they are isomorphic (as $F$-algebras):

![Diagram](attachment:diagram.png)

**Definition**
A \( \mathbf{V} \)-enriched category \( \mathbf{C} \) is \( \mathbf{V} \)-algebraically compact if every \( \mathbf{V} \)-endofunctor \( F : \mathbf{C} \to \mathbf{C} \) is algebraically compact.

**Example**
The category \( \mathbf{DCPO} \) is not \( \mathbf{DCPO} \)-algebraically compact. However, the category \( \mathbf{DCPO}_{\perp!} \) is \( \mathbf{DCPO} \)-algebraically compact.

The two categories are related via the following \( \mathbf{DCPO} \)-enriched symmetric monoidal adjunction:

\[
\begin{array}{ccc}
\mathbf{DCPO}_{\perp!} & \xleftarrow{L} & \mathbf{DCPO} \\
\xrightarrow{U} & & \\
\end{array}
\]

where \( U \) is the forgetful functor and \( L \) is the lift functor:

\[
L(X) = X_{\perp} = X \sqcup \{ \bot \}
\]

\[
L(f) = f_{\perp} = x \mapsto \begin{cases} f(x), & \text{if } x \neq \bot \\ \bot, & \text{if } x = \bot \end{cases}
\]
Lifting the model

We take inspiration from the classical case and we will *lift* our entire model to get algebraic compactness.
Zero objects

In order to lift $\text{DFam}(M)$, we will need additional structure on $M$.

**Definition**

A *zero object* in a category $\mathbf{C}$ is an object $0$ which is both initial and terminal.

If $\mathbf{C}$ has a zero object, then each homset $\mathbf{C}(A, B)$ has a canonical morphism $0_{A,B}$ given by:

$$
\begin{array}{ccc}
A & \xrightarrow{0_{A,B}} & B \\
\downarrow & & \downarrow \\
0 & \downarrow & 0
\end{array}
$$

**Example**

- The zero object in $\text{DCPO}_!$ is the singleton $\{\bot\}$. The zero morphism $0_{X,Y}$ is given by the constant function $0_{X,Y}(x) = \bot_Y$.
- Similarly, the zero object in $\text{Set}_! \cong \{\bot\}/\text{Set}$, the category of pointed sets, is again the singleton $\{\bot\}$.
- Neither $\text{Set}$, nor $\text{DCPO}$ have a zero object.

**Remark**

- *In $\text{DCPO}_!$, $0_{X,Y}$ is the least element in $\text{DCPO}_!(X, Y)$.*
- *Existence of a zero object in $\text{DFam}_!$ is necessary for $\text{DCPO}$-algebraic compactness.*
Updated assumptions on $M$

In order to define $\text{DFam}_{\bot!}(M)$, we will need an additional assumption on $M$.

Given a symmetric monoidal category $Q$, we assume that:

- $M$ is symmetric monoidal closed.
- $M$ is product-complete.
- There exists a full embedding $Q \rightarrow M$.
- $M$ has a zero object.

Example

- If $Q$ is enriched over $\text{Set}_{\bot!}$, then the category $M = [Q^{op}, \text{Set}_{\bot!}]$ satisfies these requirements and is in addition complete and cocomplete.
The category $\text{DFam}_{\bot!}$

**Definition**
For a given category $\mathbf{C}$ with zero object $0$, we define $\text{DFam}_{\bot!}(\mathbf{C})$ to be the category such that:

- Objects are pairs $(X, A)$ where $X$ is a poset category whose induced poset is a pointed dcpo and $A : X \to \mathbf{C}$ is a functor, such that $A(\bot) = 0$.
- A morphism $(X, A) \to (Y, B)$ is a pair $(f, \phi)$ where $f : X \to Y$ is a functor whose induced order-preserving function is strict Scott-continuous and $\phi : A \to B \circ f$ is a natural transformation.
- Composition of morphisms is given by: $(g, \psi) \circ (f, \phi) = (g \circ f, \psi f \circ \phi)$.

**Proposition**
$\text{DFam}_{\bot!}(\mathbf{C})$ has all small coproducts.

**Theorem**
If $\mathbf{M}$ is a symmetric monoidal closed and product-complete category with a zero object, then $\text{DFam}_{\bot!}(\mathbf{M})$ is symmetric monoidal closed with the following structure:

- $I = (I, I)$
- $(X, A) \otimes (Y, B) = (X \otimes Y, A \otimes B)$, where $(A \otimes B)(z) = \begin{cases} A(x) \otimes B(y), & \text{if } z = (x, y) \\ 0, & \text{if } z = \bot \end{cases}$
- $(X, A) \multimap (Y, B) = (X \multimap Y, A \multimap B)$, where $(A \multimap B)(f) = \prod_{x \in X} (A(x) \multimap B \circ f(x))$.

If, in addition, $\mathbf{M}$ is complete and cocomplete, then $\text{DFam}_{\bot!}(\mathbf{M})$ is also complete and cocomplete.
Zero objects and morphisms in \( \text{DFam}_{\bot!} \)

**Proposition**

\( \text{DFam}_{\bot!} \) has a zero object given by \((0,0)\). Thus, for any two \( \text{DFam}_{\bot!} \)-objects \((X,A)\) and \((Y,B)\), there’s a canonical morphism \((0_X,Y,0_{A,B})\), where:

\[
0_{A,B} : A \to B \circ 0_X, Y \\
(0_{A,B})_x = 0_{A(x),0}
\]
So far, \( \text{DFam}_{\bot !} \) shares all of the crucial properties of \( \text{Fam} \), as required. But, we get in addition:

**Theorem**

\( \text{DFam}_{\bot !} \) is \( \text{DCPO}_{\bot !} \)-enriched. More specifically, given two morphisms

\[
(f, \phi) : (X, A) \rightarrow (Y, B), \quad (g, \psi) : (X, A) \rightarrow (Y, B)
\]

we define an order \( \sqsubseteq \) on \( \text{DFam}_{\bot !}((X, A), (Y, B)) \) by

\[
(f, \phi) \sqsubseteq (g, \psi) \iff f \leq g \text{ and } B(f(x) \leq g(x)) \circ \phi_x = \psi_x \text{ for each } x \in X
\]

or

\[
(f, \phi) = (0_X, Y, 0_{A, B}).
\]
A DCPO\(_\bot\!\!1\)-enriched adjunction

The essential structure of our categorical model is the DCPO-enriched symmetric monoidal adjunction:

\[
\begin{array}{ccc}
\text{DCPO} & \downarrow & \text{DFam} \\
\text{DFam}(I, -) & F & \text{DFam}(I, -)
\end{array}
\]

where \( F \) is given by:

\[
F(X) = (X, I_X), \quad \text{where } I_X(x) = I
\]

\[
F(f) = (f, \iota), \quad \text{where } \iota_x = \text{id}_I.
\]

We can now extend this.

**Theorem**

The following diagram

\[
\begin{array}{ccc}
\text{DCPO} & \downarrow & \text{DFam} \\
\text{DFam}(I, -) & \overline{F} & \text{DFam}(I, -)
\end{array}
\]

is a DCPO\(_\bot\!\!1\)-enriched symmetric monoidal adjunction, where \( \overline{F} \) is given by:

\[
\overline{F}(X) = (X, J_X), \quad \text{where } J_X(x) = \begin{cases} I, & \text{if } x \neq \bot \\ 0, & \text{if } x = \bot \end{cases}
\]

\[
\overline{F}(f) = (f, \gamma), \quad \text{where } \gamma_x = \begin{cases} \text{id}_I, & \text{if } x \neq \bot \\ \text{id}_0, & \text{if } x = \bot \end{cases}
\]
The two order-enriched models are related by the following diagram:

\[
\begin{array}{c}
\text{DCPO}_{\perp!} \quad \perp \quad \text{DFam}_{\perp!} \\
\text{DFam}_{\perp!}(I, -) \quad \perp \quad \text{DFam}_{\perp!}(I, -) \\
L \quad \perp \quad U \quad \perp \quad L' \quad \perp \quad U'
\end{array}
\]

where:

- Both \text{DFam} and \text{DFam}_{\perp!} are defined over the same category \( M \) which is product-complete, SMC and has a zero object.
- All the adjunctions are \text{DCPO}-enriched symmetric monoidal.
- The top adjunction is also \text{DCPO}_{\perp!}-enriched.
- The adjunctions commute.
- The functors \( L, L' \) are the lift functors.
- The functors \( U, U' \) are the obvious forgetful functors.
- If \( M \) is complete and cocomplete, then \text{DFam} and \text{DFam}_{\perp!} are also complete and cocomplete.
- Composing any two adjunctions gives a linear/non-linear model as described by Benton.
Recursive types

We can now state our main result.

**Theorem**

*If the category \( M \) is cocomplete, then \( \text{DFam}_{\perp!}(M) \) is DCPO-algebraically compact.*

- This means we can find canonical solutions to recursive domain equations, e.g. \( F(D) \cong D \) in \( \text{DFam}_{\perp!} \) for any DCPO-functor \( F \).
- Work-in-progress: classify the DCPO-endofunctors.
  - Restricting to parameter types OK – so we may model classical recursion.
  - Functors corresponding to \( \alpha, 0, I, !A \) also OK.
- (Ideally?) all type expressions of our type system induce DCPO-functors only.

Then, we may extend the type system with a recursive type constructor:

\[
\text{Types} \quad A, B \ ::= \quad \alpha \mid 0 \mid A + B \mid I \mid A \otimes B \mid A \rightarrow B \mid !A \mid \text{Circ}(T, U) \mid \mu X.A
\]

and we may extend the term language with terms:

\[
\text{Terms} \quad M, N \ ::= \quad \cdots \mid \text{fold}_{\mu X.A}(M) \mid \text{unfold}(M)
\]

which can be used for the implementation of recursion on the term level.
One diagram summary

- Let $Q$ be a symmetric monoidal category enriched over $\text{Set}_{\bot!}$ and let $M := [Q^{\text{op}}, \text{Set}_{\bot!}]$.
- $\text{Fam}$, $\text{DFam}$ and $\text{DFam}_{\bot!}$ are defined over $M$.
- $\text{Fam}$, $\text{DFam}$ and $\text{DFam}_{\bot!}$ are SMC, complete and cocomplete.
- All adjunctions are symmetric monoidal.
- Adjunctions in top square are $\text{DCPO}$-enriched.
- Top adjunction is $\text{DCPO}_{\bot!}$-enriched.
- The adjunctions commute.
- Any adjunction between $\text{Set}$ or $\text{DCPO}$ and a right-column category forms a linear/non-linear model as described by Benton.
- $\text{DCPO}_{\bot!}$ and $\text{DFam}_{\bot!}$ are algebraically compact for $\text{DCPO}$-endofunctors.
Conclusion and Future Work

- Proto-Quipper-M is a circuit description language proposed by Rios and Selinger.
- It allows us to describe families of morphisms (circuits) from any symmetric monoidal category $Q$ enriched over $\mathbf{Set}_{\perp}$.
- If $Q = \mathbf{FdCStar}$, then we get a quantum programming language.
- We extended their categorical model with support for recursive types and terms.
- We need to update the operational and categorical semantics and prove computational soundness and adequacy.
- We are planning on investigating dependent types as future work.
- We would also like to investigate applications outside of quantum programming.
- Some of the mentioned results are work-in-progress and we have to verify some remaining details.
  - But we are confident we are on the right track.
Thank you for your attention!