Categorical models of circuit description languages

Bert Lindenhovius    Michael Mislove    Vladimir Zamdzhiev

Department of Computer Science
Tulane University

29 October 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. Primary application is in quantum computing, but the language can describe arbitrary string diagrams. Their model supports primitive recursion, but does not support general recursion.
Proto-Quipper-M is used to describe families of morphisms of an arbitrary, but fixed, symmetric monoidal category, which we denote $\mathbf{M}$.

Example

If $\mathbf{M} = \text{FdCStar}$, 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).\footnote{Figure source: 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**

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

**M-types**

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

The term language is given by:

**Terms**

\[ m, n ::= x | \ell | c | \text{let } x = m \text{ in } n \]

\[ \square_A m | \text{left}_{A,B} m | \text{right}_{A,B} m | \text{case } m \text{ of } \{ \text{left } x \rightarrow n | \text{right } y \rightarrow p \} \]

\[ \ast | m ; n | \langle m, n \rangle | \text{let } \langle x, y \rangle = m \text{ in } n | \lambda x^A.m | mn \]

\[ \text{lift } m | \text{force } n | \text{box}_T m | \text{apply}(m, n) | (\tilde{\ell}, C, \tilde{\ell}') \]
Our approach

- Consider an *abstract* categorical model for the same language.
- Describe a *candidate* categorical model for each of the following language variants:
  - The original Proto-Quipper-M language (base).
  - Proto-Quipper-M extended with general recursion (base+rec).
  - Proto-Quipper-M extended with dependent types (base+dep).
  - Proto-Quipper-M extended with dependent types and recursion (base+dep+rec).
An abstract model of the base language

Conjecture

A model of the base language is given by the following data:

1. A cartesian closed category $\mathbf{V}$ (the category of parameter values) enriched over itself such that:
   - $\mathbf{V}$ has finite coproducts.
   - $\mathbf{V}$ has colimits of $\omega$-sequences.

2. A $\mathbf{V}$-enriched symmetric monoidal category $\mathbf{M}$ representing the circuits.

3. A $\mathbf{V}$-enriched symmetric monoidal closed category $\mathbf{L}$ (the category of (linear) higher-order circuits) such that:
   - $\mathbf{L}$ has $\mathbf{V}$-copowers.
   - $\mathbf{L}_0$ has finite coproducts.
   - $\mathbf{L}_0$ has colimits of $\omega$-sequences.

4. A $\mathbf{V}$-enriched fully faithful strong symmetric monoidal embedding $E : \mathbf{M} \to \mathbf{L}$.

5. A $\mathbf{V}$-enriched symmetric monoidal adjunction:

\[
\begin{array}{ccc}
\mathbf{V} & \xrightarrow{\bot} & \mathbf{L} \\
L(I,-) & \rotatebox{90}{\reflectbox{$\sim$}} & \rotatebox{90}{\reflectbox{$\sim$}} & \rotatebox{90}{\reflectbox{$\sim$}}
\end{array}
\]

Less formally, a model of Proto-Quipper-M is given by a model of ILL, where one has to exploit the enrichment.
Concrete models of the base language

Fix an arbitrary symmetric monoidal category $\mathcal{M}$, and embed it via the Yoneda embedding into $\overline{\mathcal{M}} = [\mathcal{M}^{\text{op}}, \text{Set}]$. 

Concrete models of the base language

Fix an arbitrary symmetric monoidal category $\mathcal{M}$, and embed it via the Yoneda embedding into $\overline{\mathcal{M}} = [\mathcal{M}^{\text{op}}, \text{Set}]$.

The original Proto-Quipper-M model is given by the model of ILL

$$\begin{array}{ccc}
\text{Set} & \xleftarrow{\perp} & \text{Fam}[\overline{\mathcal{M}}] \\
\downarrow & & \downarrow \\
\text{Fam}[\overline{\mathcal{M}}](I, -) & \xleftarrow{\perp} & \text{Fam}[\mathcal{M}]
\end{array}$$
Concrete models of the base language

Fix an arbitrary symmetric monoidal category $\mathcal{M}$, and embed it via the Yoneda embedding into $\overline{\mathcal{M}} = [\mathcal{M}^{\text{op}}, \text{Set}]$.
The original Proto-Quipper-M model is given by the model of ILL

\[
\begin{array}{ccc}
\text{Set} & \dashv & \text{Fam}[\overline{\mathcal{M}}] \\
\downarrow & & \downarrow \\
\text{Fam}[\mathcal{M}](I, -) & \dashv & \text{Fam}[\mathcal{M}]
\end{array}
\]

**Definition**
Given a locally small category $\mathcal{C}$, the category $\text{Fam}[\mathcal{C}]$ consists of the following data:

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

Fix an arbitrary symmetric monoidal category $\mathcal{M}$, and embed it via the Yoneda embedding into $\widehat{\mathcal{M}} = [\mathcal{M}^{\text{op}}, \text{Set}]$.

The original Proto-Quipper-M model is given by the model of ILL

$$
\begin{array}{c}
\text{Set} \\
\downarrow \\
\text{Fam}[\mathcal{M}]
\end{array} \\
\begin{array}{c}
\text{Fam}[\mathcal{M}](I, -)
\end{array}
$$

Definition

Given a locally small category $\mathcal{C}$, the category $\text{Fam}[\mathcal{C}]$ consists of the following data:

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

Theorem (Rios & Selinger 2017)

This categorical model of Proto-Quipper-M is computationally sound and adequate with respect to its operational semantics.
Concrete models of the base language (contd.)

Fix an arbitrary symmetric monoidal category $\mathbf{M}$. A simpler model for the same language is given by the model of ILL:

$$\begin{array}{ccc}
\text{Set} & \xleftarrow{\bot} & \overline{\mathbf{M}} \\
\xrightarrow{\bot} & \overline{\mathbf{M}(I,-)} & \xrightarrow{- \odot I}
\end{array}$$

where $\overline{\mathbf{M}} = [\mathbf{M}^{\text{op}}, \text{Set}]$. 
Concrete models of the base language (contd.)

Fix an arbitrary symmetric monoidal category $\mathcal{M}$. A simpler model for the same language is given by the model of ILL:

$$\begin{array}{c}
\text{Set} \\
\downarrow \bigcirc I \\
\mathcal{L}(I, -)
\end{array} \cong \begin{array}{c}
\mathcal{M}(I, -)
\end{array}$$

where $\mathcal{M} = [\mathcal{M}^{\text{op}}, \text{Set}]$.

Remark

When $\mathcal{M} = 1$, the latter model degenerates to $\text{Set}$ which is a model of a simply-typed (non-linear) lambda calculus.
Concrete models of the base language (contd.)

Fix an arbitrary symmetric monoidal category $\mathbf{M}$. A simpler model for the same language is given by the model of ILL:

\[
\begin{array}{ccc}
\text{Set} & \xleftarrow{\circ} & \mathbf{M} \\
\downarrow & & \downarrow \\
\mathbf{M}(I, -) & = & \mathbf{M}
\end{array}
\]

where $\mathbf{M} = [\mathbf{M}^{\text{op}}, \text{Set}]$.

**Remark**

When $\mathbf{M} = 1$, the latter model degenerates to $\text{Set}$ which is a model of a simply-typed (non-linear) lambda calculus.

Equipping $\mathbf{M}$ with the free $\text{DCPO}$-enrichment, we can embed it into a $\text{DCPO}$-enriched category $\mathbf{M} = [\mathbf{M}^{\text{op}}, \text{DCPO}]$ of higher order circuits, which yields another concrete (order-enriched) Proto-Quipper-$\mathbf{M}$ model:

\[
\begin{array}{ccc}
\text{DCPO} & \xleftarrow{\circ} & \mathbf{M} \\
\downarrow & & \downarrow \\
\mathbf{M}(I, -) & = & \mathbf{M}
\end{array}
\]
Fix an arbitrary symmetric monoidal category $\mathcal{M}$. Original Proto-Quipper-M model:

![Diagram]

Simpler model:

![Diagram]

**Question:** What does the extra layer of abstraction provide?

**Conjecture:** A model of the language extended with dependent types, since

$$\text{Fam}[\mathcal{C}] \to \text{Set}, \quad (X, A) \mapsto A$$

is a fibration.
Dependent types

- Types that depend on terms, i.e., the type of lists of natural numbers of length \( n \)

\[ n : \mathbb{N} \vdash \text{NatList}(n) : \text{Type}. \]

- Can be regarded as a family of types indexed by term variables \( n : \mathbb{N} \):

\[ \text{NatList} = (\text{NatList}(n))_{n : \mathbb{N}}. \]

- This is like sets depending on sets, i.e., \( S = (S_x)_{x \in X} \) with \( X \in \text{Set} \), or equivalently, a pair \((X, S)\) with \( S : X \to \text{Set} \) a functor,

- Hence fibrations as \( \text{Fam}[\text{Set}] \to \text{Set} \) can be used as models for dependent type theory.
Linear dependent types

Theorem
The category $\text{Fam}[\mathcal{M}]$ is a model of dependently typed intuitionistic linear logic (type dependence is allowed only on intuitionistic terms) $^2$.

Conjecture
The symmetric monoidal adjunction:

\[
\begin{array}{ccc}
\text{Set} & \xrightarrow{- \odot I} & \text{Fam}[\mathcal{M}] \\
\downarrow & & \downarrow \\
\text{Fam}[\mathcal{M}](I, -)
\end{array}
\]

is a model of Proto-Quipper-$M$ extended with dependent types.

Remark
If $\mathcal{M} = 1$, the above model degenerates to

$\text{Fam}[\mathcal{M}] = \text{Fam}[\mathcal{M}^{op}, \text{Set}] \cong \text{Fam}[\text{Set}] \sim [2^{op}, \text{Set}]$, which is a closed comprehension category and thus a model of intuitionistic dependent type theory $^3$.


$^3$ Bart Jacobs. Categorical Logic and Type Theory. 1999.
Abstract model with dependent types?

Theorem

A model of dependently typed intuitionistic linear logic is given by a monoidal fibration with some additional structure, i.e., comprehension\(^4\).


Abstract model with dependent types?

Theorem

A model of dependently typed intuitionistic linear logic is given by a monoidal fibration with some additional structure, i.e., comprehension\(^4\).

Conjecture

An abstract model of Proto-Quipper-M extended with dependent types is given by an *enriched* monoidal fibration\(^5\) with some additional structure, i.e., comprehension.

---


What about recursion?

- Forget about dependent types for now.
- Consider the base Proto-Quipper-M language.
- How can we model general recursion?
What about recursion?

- Forget about dependent types for now.
- Consider the base Proto-Quipper-M language.
- How can we model general recursion?
  - We already have a concrete order-enriched model:

\[
\begin{array}{c}
\text{DCPO} \\
\downarrow \\
\overline{M}
\end{array}
\quad \xrightarrow{\circ} 
\begin{array}{c}
\downarrow \\
\overline{M(I, -)}
\end{array}
\]

where \(\overline{M} = [M^{\text{op}}, \text{DCPO}]\), and where the underlying induced (co)monad endofunctors are algebraically compact.

- Thus, we add partiality to the above model:

\[
\begin{array}{c}
\text{DCPO}_{\perp!} \\
\downarrow \\
\overline{M^*}
\end{array}
\quad \xrightarrow{\circ} 
\begin{array}{c}
\downarrow \\
\overline{M^*_*(I, -)}
\end{array}
\]

where \(M^*_*\) is the \(\text{DCPO}_{\perp!}\)-category obtained by freely adding a zero object to \(M\) and \(\overline{M^*_*} = [M^{\text{op}}^*, \text{DCPO}_{\perp!}]\) is the associated enriched functor category.
Proposed concrete model of Proto-Quipper-M extended with recursion

\[
\text{DCPO} \quad \bot \quad \land \quad M^* \quad (I, -) \quad \land \quad M^*(I, -) \quad \land \quad \text{DCPO}
\]

Remark

If \( M = 1 \), then the above model degenerates to the left vertical adjunction, which is a model of a simply-typed lambda calculus with term-level recursion.
Abstract model with recursion?

Theorem

A categorical model of a linear/non-linear lambda calculus extended with recursion is given by a model of ILL:

\[
V \vdash L \\
\begin{array}{c}
\downarrow \\
\end{array}
\]

where \( F \) (or equivalently \( G \)) is algebraically compact \(^6\).

\(^6\) Benton & Wadler. *Linear logic, monads and the lambda calculus*. LiCS’96.
Abstract model with recursion?

**Theorem**

A categorical model of a linear/non-linear lambda calculus extended with recursion is given by a model of ILL:

$$V \vdash L$$

where $FG$ (or equivalently $GF$) is algebraically compact 6.

**Conjecture**

An abstract model of Proto-Quipper-M extended with recursion is given by a model of Proto-Quipper-M:

$$V \vdash L$$

where the underlying induced (co)monad endofunctors are algebraically compact.

**Remark**

*The above definition is not the whole picture, but it describes the essential idea.*

---

Benton & Wadler. *Linear logic, monads and the lambda calculus*. LiCS’96.
What about recursion and dependent types simultaneously?

- Idea: \textbf{CFam}[C], a version of the families construction where objects of a category \( C \) are indexed by dcpo’s.

- Must have a linear/non-linear adjunction between \textbf{CFam}[C] and \textbf{DCPO}.

- The induced monad and comonad must be algebraically compact.

- The right adjoint of the adjunction must be a representable functor.

- For this reason \textbf{CFam}[C] must be \textbf{DCPO}-enriched.

- Must have a enriched monoidal fibration \( \textbf{CFam}[C] \to \textbf{DCPO} \) with some extra structure, i.e., comprehension.
Definition \textit{CFam}: 

Construction: a generalization of the \textit{CFam}[DCPO]-construction\textsuperscript{78} with \textit{DCPO} replaced by a \textit{DCPO}-enriched category \textit{C}.


\textsuperscript{8}Bart Jacobs. \textit{Categorical Logic and Type Theory}. 1999.
Definition CFam:

Construction: a generalization of the CFam[DCPO]-construction\(^7\) with DCPO replaced by a DCPO-enriched category \(\mathcal{C}\).

- Objects are pairs \((X, A)\) with \(X \in \text{DCPO}\) and \(A : X \to \mathcal{C}\) is a functor such that:


\(^8\)Bart Jacobs. *Categorical Logic and Type Theory*. 1999.
Definition \textbf{CFam}: 

Construction: a generalization of the \textbf{CFam}[DCPO]-construction\textsuperscript{78} with \textbf{DCPO} replaced by a \textbf{DCPO}-enriched category \textbf{C}.

- Objects are pairs \((X, A)\) with \(X \in \textbf{DCPO}\) and \(A : X \to \textbf{C}\) is a functor such that:
  - \(A(x \leq y)\) is an embedding for each \(x \leq y\) in \(X\); the corresponding projection is denoted by \(A(x \leq y)^p\);
  - \(A(\sup D) = \lim_{d \in D} Ad\) for each directed \(D \subseteq X\);


\textsuperscript{8}Bart Jacobs. \textit{Categorical Logic and Type Theory}. 1999.
Definition **CFam**: 

Construction: a generalization of the **CFam[DCPO]**-construction\(^7\) with **DCPO** replaced by a **DCPO**-enriched category **C**. 

- Objects are pairs \((X, A)\) with \(X \in \text{DCPO}\) and \(A : X \to C\) is a functor such that:
  - \(A(x \leq y)\) is an embedding for each \(x \leq y\) in \(X\); the corresponding projection is denoted by \(A(x \leq y)^p\); 
  - \(A(\sup D) = \lim_{d \in D} A(d)\) for each directed \(D \subseteq X\); 
- A morphism \((X, A) \to (Y, B)\) is a pair \((f, \varphi)\) where \(f : X \to Y\) is a Scott continuous and \(\varphi : A \to B \circ f\) consists of morphisms \(\varphi_x : Ax \to B \circ f(x)\) satisfying:
  - \(B(f(x) \leq f(y)) \circ \varphi_x \leq \varphi_y \circ A(x \leq y)\) for each \(x \leq y\) in \(X\) (i.e., \(\varphi\) is lax natural); 
  - \(\varphi_y = \sup_{x \in D} B(f(x) \leq f(y)) \circ \varphi_x \circ A(x \leq y)^p\) for each directed \(D \subseteq X\) with supremum \(y\). 

---


\(^8\)Bart Jacobs. *Categorical Logic and Type Theory*. 1999.
We define $(f, \varphi) \leq (g, \psi)$ in $\mathbf{CFam}[\mathcal{C}]((X, A), (Y, B))$ if $f \leq g$ in $[X \to Y]$ and

$$B(f(x) \leq g(x)) \varphi_x \leq \psi_x$$

in $\mathcal{C}(Ax, Bf(x))$ for each $x \in X$. 
DCPO-enrichment of CFam[C]

We define \((f, \varphi) \leq (g, \psi)\) in \(\text{CFam}[C]((X, A), (Y, B))\) if \(f \leq g\) in \([X \to Y]\) and

\[
B(f(x) \leq g(x)) \varphi_x \leq \psi_x
\]

in \(C(Ax, Bf(x))\) for each \(x \in X\).

If \(\{(f_i, \varphi_i) : i \in I\}\) is a directed set in \(\text{CFam}[C]((X, A), (Y, B))\), then its supremum \((f, \varphi)\) is determined by

\[
f = \sup_{i \in I} f_i
\]

calculated in the dcpo \([X \to Y]\), and

\[
\varphi_x = \sup_{i \in I} B(f_i(x) \leq f(x))(\varphi_i)_x
\]

calculated in the dcpo \(C(Ax, Bf(x))\) for each \(x \in X\);
Monoidal structure

Let \((X, A)\) and \((Y, B)\) be objects in \(\text{CFam}[C]\). Then:

\[
(X, A) \otimes (Y, B) = (X \times Y, A \otimes B),
\]

where \((A \otimes B)(x, y) = (Ax) \otimes (By)\).

Question: do we need monoidal closure of the total category? If so it is probably of the form:

\[
(X, A) \Rightarrow (Y, B) = ([X \to Y], A \Rightarrow B),
\]

with \((A \Rightarrow B) f = \int_{x \in X} A x \Rightarrow B f(x)\), where \(\int\) denotes some kind of `lax end' satisfying

\[
\int_{x \in X} C(Fx, Gx) = \{\text{lax natural transformations } F \to G\}
\]

for functors \(F, G : X \to C\).

Question: what are the requirements on \(C\) to assure the existence of this `lax end'.
Monoidal structure

Let \((X, A)\) and \((Y, B)\) be objects in \(\text{CFam}[C]\). Then:

\[
(X, A) \otimes (Y, B) = (X \times Y, A \otimes B),
\]

where

\[
(A \otimes B)(x, y) = (Ax) \otimes (By).
\]
Monoidal structure

Let \((X, A)\) and \((Y, B)\) be objects in \(\text{CFam}[C]\). Then:
\[
(X, A) \otimes (Y, B) = (X \times Y, A \otimes B),
\]
where
\[
(A \otimes B)(x, y) = (Ax) \otimes (By).
\]

Question: do we need monoidal closure of the total category? If so it is probably of the form:
\[
(X, A) \rightharpoonup (Y, B) = ([X \to Y], A \rightharpoonup B),
\]
with
\[
(A \rightharpoonup B)f = \int_{x \in X} Ax \rightharpoonup Bf(x),
\]
where \(\int\) denotes some kind of ‘lax end’ satisfying
\[
\int_{x \in X} \mathcal{C}(Fx, Gx) = \{\text{lax natural transformations } F \to G\}
\]
for functors \(F, G : X \to \mathcal{C}\).
Monoidal structure

Let \((X, A)\) and \((Y, B)\) be objects in \(\text{CFam}[C]\). Then:

\[(X, A) \otimes (Y, B) = (X \times Y, A \otimes B),\]

where

\[(A \otimes B)(x, y) = (Ax) \otimes (By).\]

Question: do we need monoidal closure of the total category? If so it is probably of the form:

\[(X, A) \rightarrow (Y, B) = ([X \rightarrow Y], A \rightarrow B),\]

with

\[(A \rightarrow B)f = \int_{x \in X} Ax \rightarrow Bf(x),\]

where \(\int\) denotes some kind of ‘lax end’ satisfying

\[\int_{x \in X} \mathcal{C}(Fx, Gx) = \{\text{lax natural transformations } F \rightarrow G\}\]

for functors \(F, G : X \rightarrow \mathcal{C}\).

Question: what are the requirements on \(\mathcal{C}\) to assure the existence of this ‘lax end’.
Abstract model with recursion and dependent types?

- This is the most complicated case by far.

\[ \text{DCPO} \perp \left( \begin{array}{c} L \\ \downarrow \end{array} \right) \quad \left( \begin{array}{c} \downarrow \\ - \circ I \end{array} \right) \quad \left( \begin{array}{c} - \circ I \\ \downarrow \end{array} \right) \quad \left( \begin{array}{c} \downarrow \\ \text{CFam}_\perp[\mathcal{M}_\ast] \end{array} \right) \quad \left( \begin{array}{c} \downarrow \\ U \end{array} \right) \quad \left( \begin{array}{c} \downarrow \end{array} \right) \quad \left( \begin{array}{c} \text{CFam}_\perp[\mathcal{M}_\ast](I, -) \end{array} \right) \quad \left( \begin{array}{c} \downarrow \end{array} \right) \quad \left( \begin{array}{c} \text{UD} \end{array} \right) \quad \left( \begin{array}{c} \downarrow \end{array} \right) \quad \left( \begin{array}{c} \text{CFam}[\mathcal{M}] \end{array} \right) \quad \left( \begin{array}{c} \downarrow \end{array} \right) \quad \left( \begin{array}{c} \text{CFam}[\mathcal{M}](I, -) \end{array} \right) \quad \left( \begin{array}{c} \downarrow \end{array} \right) \quad \left( \begin{array}{c} \text{DCPO} \end{array} \right) \]

Remark

*If \( \mathcal{M} = 1 \), then the model collapses to a model which is very similar to Palmgren and Stoltenberg-Hansen’s model of partial intuitionistic dependent type theory* \( ^9 \).

---

Conclusion

- By taking the enrichment of certain denotational models into account, one can obtain models of circuit description languages.

- Systematic construction for concrete models that works for any circuit (string diagram) model described by a symmetric monoidal category.

- We have identified different candidate models for Proto-Quipper-M depending on the feature set.

- Plenty of work (and verification) remains to be done...
Thank you for your attention.