Categorical models of circuit description languages

Bert Lindenhovius  Michael Mislove  Vladimir Zamdzhiev

Department of Computer Science
Tulane University

12 October 2017
Proto-Quipper-M

- We will consider several variants of 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 $M$.

**Example**

If $M = \text{FdCStar}$, 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 $n$.

![Quantum Fourier Transform](https://commons.wikimedia.org/w/index.php?curid=14545612)

*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](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**

\[ M, N ::= x | I | c | \text{let } x = M \text{ in } N \]

\[ \square_A M | \text{left}_{A,B} M | \text{right}_{A,B} M | \text{case } M \text{ of } \{ \text{left } x \rightarrow N | \text{right } y \rightarrow P \} \]

\[ \star | M; N | \langle M, N \rangle | \text{let } \langle x, y \rangle = M \text{ in } N | \lambda x^A. M | MN \]

\[ \text{lift } M | \text{force } M | \text{box}_T M | \text{apply}(M, N) | \langle \tilde{I}, C, \tilde{I}' \rangle \]
Families Construction

The following construction is well-known.

**Definition**
Given a category $\mathcal{C}$, we define a new category $\text{Fam}[\mathcal{C}]$:

- 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 $(f, \phi)$ 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**
$\text{Fam}[\mathcal{C}]$ is the free coproduct completion of $\mathcal{C}$ and as a result has all small coproducts.

**Proposition**
If $\mathcal{C}$ is a symmetric monoidal closed and product-complete category, then $\text{Fam}[\mathcal{C}]$ is a symmetric monoidal closed category.
Categorical Model

Definition

- A symmetric monoidal closed and product-complete category $\overline{M}$.
- A fully faithful strong monoidal embedding $M \rightarrow \overline{M}$.
- A symmetric monoidal closed category $\text{Fam}[\overline{M}]$ which we will refer to as $\text{Fam}$.
- A symmetric monoidal adjunction:

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

where

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

Remark

For any symmetric monoidal category $M$, we can set $\overline{M} := [M^{op}, \text{Set}]$ 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.*

Question

*Sam Staton: Why do you need the Fam construction for this?*

Open Problem

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

**Related work:** Rennela and Staton describe a different circuit description language where they also use enriched category theory.
• Everybody is advertising books, so I have to do it as well.
A model of Intuitionistic Linear Logic (ILL) as described by Benton is given by the following data:

- A cartesian closed category $\mathcal{V}$.
- A symmetric monoidal closed category $\mathcal{L}$.
- A symmetric monoidal adjunction:

$$
\begin{array}{ccc}
\mathcal{V} & \xrightarrow{\perp} & \mathcal{L} \\
G & & F
\end{array}
$$

---

Nick Benton. *A mixed linear and non-linear logic: Proofs, terms and models.* CSL’94

Bert Lindenhovius, Michael Mislove, Vladimir Zamdzhev

Categorical models of circuit description languages
A model of the Enriched Effect Calculus (EEC) is given by the following data:

- A cartesian closed category $\mathbf{V}$, enriched over itself.
- A $\mathbf{V}$-enriched category $\mathbf{L}$ with powers, copowers, finite products and finite coproducts.
- A $\mathbf{V}$-enriched adjunction:

$$\begin{array}{c}
\mathbf{V} \\
\downarrow \downarrow \\
\mathbf{L} \\
\end{array} \\
\begin{array}{c}
F \\
\downarrow \\
G \\
\end{array}$$

**Theorem**

*Every model of $\text{ILL}$ with additives determines an $\text{EEC}$ model.*
An abstract model of the base language

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:
   - $\mathbf{V}_0$ has finite coproducts.
   - $\mathbf{V}_0$ has colimits of initial sequences.

2. A $\mathbf{V}$-enriched symmetric monoidal category $\mathbf{M}$ which describes the circuit model.

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 initial sequences.

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

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

\[
\begin{array}{ccc}
\text{V} & \overset{- \odot I}{\longrightarrow} & \text{L} \\
\downarrow & & \downarrow \circlearrowright \\
\mathbf{L}(I,-) & & \\
\end{array}
\]

Less formally, a model of Proto-Quipper-M is given by an enriched model of ILL.
Concrete models of the base language

Fix an arbitrary symmetric monoidal category $M$.
The original Proto-Quipper-M model is given by the model of ILL

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

Fix an arbitrary symmetric monoidal category $\mathcal{M}$.

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

$$
\begin{array}{ccc}
\mathcal{M} & \dashv & \mathcal{P} \\
\mathcal{M}(I, -) & \downarrow & \mathcal{P}[\mathcal{M}](I, -)
\end{array}
$$

A simpler model for the same language is given by the model of ILL:

$$
\begin{array}{ccc}
\mathcal{M} & \dashv & \mathcal{P} \\
\mathcal{M}(I, -) & \downarrow & \mathcal{M}(l, -)
\end{array}
$$

where in both cases $\mathcal{M} = [\mathcal{M}^{\text{op}}, \mathcal{P}]$.

**Remark**

*When $\mathcal{M} = 1$, the latter model degenerates to $\mathcal{P}$ which is a model of a simply-typed (non-linear) lambda calculus.*
Fix an arbitrary symmetric monoidal category $\mathbf{M}$. Equipping $\mathbf{M}$ with the free DCPO-enrichment yields another concrete (order-enriched) Proto-Quipper-$\mathbf{M}$ model:

$$\mathbf{DCPO} \xrightarrow{\perp} \mathbf{M}$$

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

**Remark**

*The three concrete models of Proto-Quipper-$\mathbf{M}$ are EEC models whose underlying (unenriched) structure is a model of ILL.*
Fix an arbitrary symmetric monoidal category $\mathcal{M}$.

Original Proto-Quipper-M model:

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

Simpler model:

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

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

**Answer:** A model of the language extended with dependent types.
Linear dependent types

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

Conjecture
The symmetric monoidal adjunction:

$$
\begin{array}{ccc}
\text{Set} & \circlearrowright & \text{Fam}[M] \\
\downarrow & & \downarrow \\
\text{Fam}[M](I, -) & \circlearrowleft & \text{Fam}[M]
\end{array}
$$

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

Remark
If $M = 1$, the above model degenerates to $\text{Fam}[M] = \text{Fam}[M^{op}, \text{Set}] \cong \text{Fam}[\text{Set}] \simeq [2^{op}, \text{Set}]$, 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 an indexed monoidal category with some additional structure (comprehension, strictness, ...) \(^4\).

**Conjecture**

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

---


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:
    \[
    \begin{array}{c}
    \text{DCPO} \\
    \downarrow \\
    \bar{M}
    \end{array}
    \quad (\shortcdot, -)
    \]
    where \( \bar{M} = [M^\text{op}, \text{DCPO}] \).
  - Thus, we add partiality to the above model:
    \[
    \begin{array}{c}
    \text{DCPO} \\
    \downarrow \\
    \bar{M}_*(I, -)
    \end{array}
    \]
    where \( \bar{M}_* \) is the \( \text{DCPO}_{\perp!} \)-category obtained by freely adding a zero object to \( M \) and 
    \( \bar{M}_* = [M_*^\text{op}, \text{DCPO}_{\perp!}] \) is the associated enriched functor category.
Concrete model of Proto-Quipper-M extended with recursion

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?

Intuitionistic linear logics correspond to linear/non-linear lambda calculi under the Curry-Howard isomorphism.

Theorem

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

\[ \mathbf{V} \vdash \mathbf{L} \]

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

---

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

Intuitionistic linear logics correspond to linear/non-linear lambda calculi under the Curry-Howard isomorphism.

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

\[ \text{V} \vdash \text{L} \]

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

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

\[ \text{V} \vdash \text{L} \]

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.

---

\(^6\)Benton & Wadler. Linear logic, monads and the lambda calculus. LiCS'96.
What about recursion and dependent types simultaneously?

- This is the most complicated case by far.

![Diagram]

Remark

If $M = 1$, then the model collapses to a model which is very similar to Palmgren and Stoltenberg-Hansen's model of partial intuitionistic dependent type theory\(^7\).

---

Conjecture

An abstract model of Proto-Quipper-M extended with recursion and dependent types is given by an enriched indexed monoidal category with some additional structure (comprehension, strictness, ...) and suitable algebraic compactness conditions on the underlying adjoint functors.
Conclusion

- You can cheese yourself into a model of circuit description languages by categorically enriching certain denotational models.

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

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

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