An abstract model for Proto-Quipper-M extended with general recursion

Bert Lindenhovius Michael Mislove Vladimir Zamdzhiev

Department of Computer Science
Tulane University

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.

Circuit Model

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 $\mathbf{M} = \mathbf{FdCStar}$, 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.

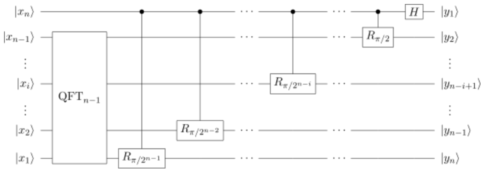

Figure: Quantum Fourier Transform on n qubits (subroutine in Shor's algorithm).[1]

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

Syntax of Proto-Quipper-M

The type system is given by:

$$
\begin{array}{llcl}
\text{Types} & A, B & ::= & \alpha \mid 0 \mid A + B \mid \mathrm{I} \mid A \otimes B \mid A \multimap B \mid !A \mid \mathbf{Circ(T, U)} \\
\text{Parameter types} & P, R & ::= & \alpha \mid 0 \mid P + R \mid \mathrm{I} \mid P \otimes R \mid !A \mid \mathbf{Circ(T, U)} \\
\text{M-types} & T, U & ::= & \alpha \mid \mathrm{I} \mid T \otimes U
\end{array}
$$

The term language is given by:

$$
\begin{array}{llcl}
\text{Terms} & M, N & ::= & x \mid l \mid c \mid \mathrm{let}\ x = M \ \mathrm{in}\ N \\
& & & \mid \Box_A M \mid \mathrm{left}_{A,B} M \mid \mathrm{right}_{A,B} M \mid \mathrm{case}\ M\ \mathrm{of}\ \{\mathrm{left}\ x \to N \mid \mathrm{right}\ y \to P\} \\
& & & \mid * \mid M; N \mid \langle M, N \rangle \mid \mathrm{let}\ \langle x, y \rangle = M\ \mathrm{in}\ N \mid \lambda x^A.M \mid MN \\
& & & \mid \mathrm{lift}\ M \mid \mathrm{force}\ M \mid \mathbf{box_T M} \mid \mathbf{apply(M, N)} \mid \mathbf{(\tilde{l}, C, \tilde{l}')}
\end{array}
$$

Families Construction

The following construction is well-known.

Definition

Given a category $\mathbf{C}$, we define a new category $\mathbf{Fam}[\mathbf{C}]$:

- Objects are pairs (X, A) where X is a discrete category and $A : X \to \mathbf{C}$ is a functor.
- A morphism $(X, A) \to (Y, B)$ is a pair (f, ϕ) 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

$\mathbf{Fam}[\mathbf{C}]$ *is the free coproduct completion of* $\mathbf{C}$ *and as a result has all small coproducts.*

Proposition

If $\mathbf{C}$ *is a symmetric monoidal closed and product-complete category, then* $\mathbf{Fam}[\mathbf{C}]$ *is a symmetric monoidal closed category.*

Categorical Model

Definition

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

$$\begin{tikzcd} \mathbf{Set} \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] & & \mathbf{Fam} \arrow[ll, bend left, "\mathbf{Fam}(I, -)"] \end{tikzcd}$$

Remark

Setting $\overline{\mathbf{M}} := [\mathbf{M}^{op}, \mathbf{Set}]$ *satisfies the first two requirements and can be done for any* $\mathbf{M}$.

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

- Describe an *abstract* categorical model for the same language.
- Describe an abstract categorical model for the language extended with recursion.

Related work: Rennela and Staton describe a different circuit description language where they also use enriched category theory.

A model of Intuitionistic Linear Logic (ILL) as described by Benton is given by the following data:

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

$$\begin{tikzcd} \mathbf{V} \arrow[rr, bend left, "F"] \arrow[rr, phantom, "\perp"] & & \mathbf{L} \arrow[ll, bend left, "G"] \end{tikzcd}$$

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

Models of the Enriched Effect Calculus

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:

$$\begin{tikzcd} \mathbf{V} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{L} \arrow[l, bend left, "G"] \end{tikzcd}$$

Theorem

Every model of ILL with additives determines an EEC model.

Egger, Møgelberg, Simpson. *The enriched effect calculus: syntax and semantics*. Journal of Logic and Computation 2012

An abstract model for Proto-Quipper-M

A model of Proto-Quipper-M is given by the following data:

1. A cartesian closed category $\mathbf{P}$ (the category of parameters) together with its self-enrichment $\mathcal{P}$, such that $\mathcal{P}$ has finite $\mathbf{P}$-coproducts.
2. A $\mathbf{P}$-symmetric monoidal category $\mathcal{M}$ with underlying category $\mathbf{M}$.
3. A $\mathbf{P}$-symmetric monoidal closed category $\mathcal{C}$ with underlying category $\mathbf{C}$ such that $\mathcal{C}$ has finite $\mathbf{P}$-coproducts.
4. A $\mathbf{P}$-strong symmetric monoidal functor $E : \mathcal{M} \to \mathcal{C}$.
5. A $\mathbf{P}$-symmetric monoidal adjunction: $\begin{tikzcd} \mathcal{P} \arrow[r, bend left, "- \odot I"] \arrow[r, phantom, "\perp"] & \mathcal{C} \arrow[l, bend left, "\mathcal{C}(I,-)"] \end{tikzcd}$,

   where $(- \odot I)$ denotes the $\mathbf{P}$-copower of the tensor unit in $\mathcal{C}$.

Remark: A model of PQM is essentially given by an **enriched** model of ILL.

Soundness

Theorem (Soundness)

Every abstract model of Proto-Quipper-M is computationally sound.

Concrete models of PQM

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

$$\begin{tikzcd} \mathbf{Set} \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] & & \mathbf{Fam}[\overline{\mathbf{M}}] \arrow[ll, bend left, "\mathbf{Fam}[\overline{\mathbf{M}}](I, -)"] \end{tikzcd}$$

Concrete models of PQM

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

$$
\begin{tikzcd}
\mathbf{Set} \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] & & \mathbf{Fam}[\overline{\mathbf{M}}] \arrow[ll, bend left, "\mathbf{Fam}[\overline{\mathbf{M}}](I, -)"]
\end{tikzcd}
$$

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

$$
\begin{tikzcd}
\mathbf{Set} \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] & & \overline{\mathbf{M}} \arrow[ll, bend left, "\overline{\mathbf{M}}(I, -)"]
\end{tikzcd}
$$

where in both cases $\overline{\mathbf{M}} = [\mathbf{M}^{\mathrm{op}}, \mathbf{Set}]$.

Remark

When $\mathbf{M} = \mathbf{1}$, *the latter model degenerates to* $\mathbf{Set}$ *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 $\mathbf{M}$.
Equipping $\mathbf{M}$ with the free $\mathbf{DCPO}$-enrichment yields another concrete (order-enriched) Proto-Quipper-M model:

$$\begin{tikzcd} \mathbf{DCPO} \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] & & \overline{\mathbf{M}} \arrow[ll, bend left, "\overline{\mathbf{M}}(I, -)"] \end{tikzcd}$$

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

Remark

The three concrete models of Proto-Quipper-M are EEC models whose underlying (unenriched) structure is a model of ILL.

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:

$$\begin{tikzcd} \mathbf{V} \arrow[rr, bend left, "F"] \arrow[rr, phantom, "\perp"] & & \mathbf{L} \arrow[ll, bend left, "G"] \end{tikzcd}$$

where FG (or equivalently GF) is parametrically algebraically compact [2]*.*

[2] Benton & Wadler. *Linear logic, monads and the lambda calculus.* LiCS'96.

Proto-Quipper-M extended with general recursion

Definition

A categorical model of PQM extended with general recursion is given by a model of PQM, where in addition:

6. The comonad endofunctor:

$$\begin{tikzcd} \mathcal{P} \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] & & \mathcal{C}, \arrow[ll, bend left, "\mathcal{C}(I,-)"] \end{tikzcd}$$

   is parametrically algebraically compact.

Moreover, if:

7. $\mathcal{P} = \mathbf{DCPO}$ and $0_{T,U} \notin \mathrm{Im}(E)$.

then we call this a *computationally adequate* categorical model of PQM extended with general recursion.

Recursion

Extend the syntax:

$$\frac{\Phi, x :!A; \emptyset \vdash m : A}{\Phi; \emptyset \vdash \mathsf{rec}\ x^{!A}\ m : A}\ (\mathsf{rec})$$

Extend the operational semantics:

$$\frac{(C, m[\mathsf{lift}\ \mathsf{rec}\ x^{!A} m/x]) \Downarrow (C', v)}{(C, \mathsf{rec}\ x^{!A} m) \Downarrow (C', v)}$$

Recursion (contd.)

Extend the denotational semantics: $\llbracket \Phi ; \emptyset \vdash \mathrm{rec}\ x^{!A}\ m : A \rrbracket := \sigma_{\llbracket m \rrbracket} \circ \gamma_{\llbracket \Phi \rrbracket}$.

$$
\begin{tikzcd}
\llbracket \Phi \rrbracket \otimes ! \llbracket \Phi \rrbracket \arrow[d, dashed, "\mathrm{id} \otimes ! \gamma_{\llbracket \Phi \rrbracket}"'] & \llbracket \Phi \rrbracket \otimes \llbracket \Phi \rrbracket \arrow[l, "\mathrm{id} \otimes F\eta"'] & \llbracket \Phi \rrbracket \arrow[l, "\Delta"'] \arrow[d, dashed, "\gamma_{\llbracket \Phi \rrbracket}"] \\
\llbracket \Phi \rrbracket \otimes ! \Omega_{\llbracket \Phi \rrbracket} \arrow[d, "\mathrm{id}"'] & & \Omega_{\llbracket \Phi \rrbracket} \arrow[ll, "\omega^{-1}_{\llbracket \Phi \rrbracket}"'] \arrow[d, "\mathrm{id}"] \\
\llbracket \Phi \rrbracket \otimes ! \Omega_{\llbracket \Phi \rrbracket} \arrow[rr, "\omega_{\llbracket \Phi \rrbracket}"] \arrow[d, dashed, "\mathrm{id} \otimes ! \sigma_{\llbracket m \rrbracket}"'] & & \Omega_{\llbracket \Phi \rrbracket} \arrow[d, dashed, "\sigma_{\llbracket m \rrbracket}"] \\
\llbracket \Phi \rrbracket \otimes ! \llbracket A \rrbracket \arrow[rr, "\llbracket m \rrbracket"'] & & \llbracket A \rrbracket
\end{tikzcd}
$$

Soundness and adequacy

Theorem (Soundess)

Every model of Proto-Quipper-M extended with recursion is computationally sound.

Theorem (Termination)

Consider a computationally adequate model of PQM extended with recursion. For any well-typed configuration (C, m)*, if* $[\![(C, m)]\!] \neq 0$*, then* $(C, m) \Downarrow$*. (Proof in progress).*

Theorem (Adequacy)

Consider a computationally adequate model of PQM extended with recursion. For any well-typed configuration (C, m)*, where* m *is a term of parameter type:*

$$[\![(C, m)]\!] \neq 0 \textit{ iff } (C, m) \Downarrow$$

Concrete model of Proto-Quipper-M extended with recursion

Let $\mathbf{M}_*$ be the $\mathbf{DCPO}_{\perp!}$-category obtained by freely adding a zero object to $\mathbf{M}$ and $\overline{\mathbf{M}_*} = [\mathbf{M}_*^{\mathrm{op}}, \mathbf{DCPO}_{\perp!}]$ be the associated enriched functor category.

$$
\begin{tikzcd}
\mathbf{DCPO}_{\perp!} \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] \arrow[d, bend left, "U"] \arrow[d, phantom, "\dashv"] & & \overline{\mathbf{M}_*} \arrow[ll, bend left, "\overline{\mathbf{M}_*}(I, -)"] \arrow[d, bend left, "\overline{U}"] \arrow[d, phantom, "\dashv"] \\
\mathbf{DCPO} \arrow[u, bend left, "L"] \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] & & \overline{\mathbf{M}} \arrow[u, bend left, "\overline{L}"] \arrow[ll, bend left, "\overline{\mathbf{M}}(I, -)"]
\end{tikzcd}
$$

Remark

If $\mathbf{M} = \mathbf{1}$*, then the above model degenerates to the left vertical adjunction, which is a model of a simply-typed lambda calculus with term-level recursion.*

Original model revisited

Fix an arbitrary symmetric monoidal category $\mathbf{M}$.
Original Proto-Quipper-M model:

$$\begin{tikzcd} \mathbf{Set} \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] & & \mathbf{Fam}[\overline{\mathbf{M}}] \arrow[ll, bend left, "\mathbf{Fam}[\overline{\mathbf{M}}](I, -)"] \end{tikzcd}$$

Simpler model:

$$\begin{tikzcd} \mathbf{Set} \arrow[rr, bend left, "- \odot I"] \arrow[rr, phantom, "\perp"] & & \overline{\mathbf{M}} \arrow[ll, bend left, "\overline{\mathbf{M}}(I, -)"] \end{tikzcd}$$

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 $\mathbf{Fam}[\overline{\mathbf{M}}]$ *is a model of dependently typed intuitionistic linear logic* [3].

Conjecture

The symmetric monoidal adjunction:
$$\begin{tikzcd} \mathbf{Set} \arrow[r, bend left, "- \odot I"] \arrow[r, phantom, "\perp"] & \mathbf{Fam}[\overline{\mathbf{M}}] \arrow[l, bend left, "{\mathbf{Fam}[\overline{\mathbf{M}}](I, -)}"] \end{tikzcd}$$
is a model of Proto-Quipper-M extended with dependent types.

Remark

If $\mathbf{M} = \mathbf{1}$, *the above model degenerates to*
$\mathbf{Fam}[\overline{\mathbf{M}}] = \mathbf{Fam}[\mathbf{M}^{op}, \mathbf{Set}] \cong \mathbf{Fam}[\mathbf{Set}] \simeq [\mathbf{2}^{op}, \mathbf{Set}]$, *which is a closed comprehension category and thus a model of intuitionistic dependent type theory*[4].

[3] Matthijs Vákár. *In Search of Effectful Dependent Types*. PhD thesis, University of Oxford.
[4] 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, ...) [5].

Conjecture

An abstract model of Proto-Quipper-M extended with dependent types is given by an ***enriched*** *indexed monoidal category* [6] *with some additional structure (comprehension, strictness, ...).*

---

[5] Matthijs Vákár. *In Search of Effectful Dependent Types*. PhD thesis, University of Oxford.
[6] Michael Shulman. *Enriched Indexed Categories*. Theory and Application of Categories, 2013.

- This is the most complicated case by far.

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\mathbf{DCPO}_{\perp!} \arrow[r, bend left, "- \odot I"] \arrow[r, phantom, "\perp"] \arrow[d, bend left, "U"] \arrow[d, phantom, "\dashv"] & \mathbf{CFam}_{\perp!}[\overline{\mathbf{M}_*}] \arrow[l, bend left, "\mathbf{CFam}_{\perp!}[\overline{\mathbf{M}_*}](I,-)"] \arrow[d, bend left, "\overline{U}"] \arrow[d, phantom, "\dashv"] \\
\mathbf{DCPO} \arrow[u, bend left, "L"] \arrow[r, bend left, "- \odot I"] \arrow[r, phantom, "\perp"] & \mathbf{CFam}[\overline{\mathbf{M}}] \arrow[u, bend left, "\overline{L}"] \arrow[l, bend left, "\mathbf{CFam}[\overline{\mathbf{M}}](I,-)"]
\end{tikzcd}
$$

Remark

If $\mathbf{M} = \mathbf{1}$*, 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]*.*

[7] Erik Palmgren & Viggo Stoltenberg-Hansen. *Domain interpretations of Martin-Löf's partial type theory*. Annals of Pure and Applied Logic 1990.

Abstract model with recursion and dependent types?

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

- One can construct a model of PQM by categorically enriching certain denotational models.
- We described a sound abstract model for PQM.
- We described a sound and computationally adequate abstract model for PQM with general recursion.
- Systematic construction for concrete models that works for any circuit (string diagram) model described by a symmetric monoidal category.
- We have conjectured what possible models that support dependent types should look like.

Thank you for your attention!