Categorical models of circuit description languages

Bert Lindenhovius Michael Mislove Vladimir Zamdzhiev

Department of Computer Science
Tulane University

29 October 2017

Proto-Quipper-M

- 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.

Circuit Model

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** = **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.

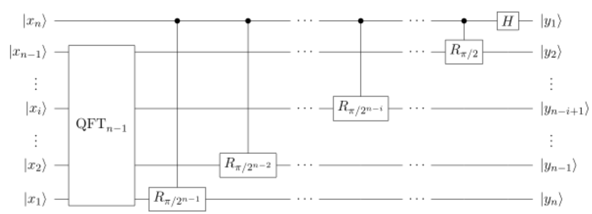


Figure: Quantum Fourier Transform on n qubits (subroutine in Shor's algorithm).[1]

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

Syntax of Proto-Quipper-M

The type system is given by:

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

The term language is given by:

$$
\begin{array}{llll}
\text{Terms} & m, n & ::= & x \mid \ell \mid c \mid \text{let } x = m \text{ in } n \\
& & & \mid \Box_A m \mid \text{left}_{A,B} m \mid \text{right}_{A,B} m \mid \text{case } m \text{ of } \{\text{left } x \to n \mid \text{right } y \to p\} \\
& & & \mid * \mid m; n \mid \langle m, n \rangle \mid \text{let } \langle x, y \rangle = m \text{ in } n \mid \lambda x^A.m \mid mn \\
& & & \mid \text{lift } m \mid \text{force } n \mid \mathbf{box_T m} \mid \mathbf{apply(m, n)} \mid (\tilde{\ell}, \mathbf{C}, \tilde{\ell}')
\end{array}
$$

Our approach

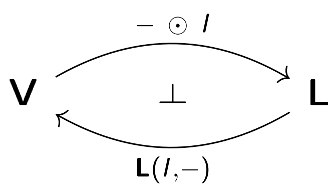
- Consider an *abstract* categorical model for the same language.
- Describe a *candidate* categorical model for each of the following language variants:
  - The original Proto-Quipper-M language (`base`).
  - Proto-Quipper-M extended with general recursion (`base+rec`).
  - Proto-Quipper-M extended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



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 $\mathbf{M}$, and embed it via the Yoneda embedding into $\overline{\mathbf{M}} = [\mathbf{M}^{\mathrm{op}}, \mathbf{Set}]$.

Concrete models of the base language

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

$$\mathbf{Set} \underset{\mathbf{Fam}[\overline{\mathbf{M}}](I,-)}{\overset{-\odot I}{\underset{\perp}{\rightleftarrows}}} \mathbf{Fam}[\overline{\mathbf{M}}]$$

Concrete models of the base language

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

$$\mathbf{Set} \underset{\mathbf{Fam}[\overline{\mathbf{M}}](I,-)}{\overset{-\odot I}{\rightleftarrows}} \mathbf{Fam}[\overline{\mathbf{M}}] \qquad (\bot)$$

Definition

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

- 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 $\varphi : A \to B \circ f$ 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 $\mathbf{M}$, and embed it via the Yoneda embedding into $\overline{\mathbf{M}} = [\mathbf{M}^{\mathrm{op}}, \mathbf{Set}]$.
The original Proto-Quipper-M model is given by the model of ILL

$$\mathbf{Set} \underset{\mathbf{Fam}[\overline{\mathbf{M}}](I,-)}{\overset{-\odot I}{\rightleftarrows}} \mathbf{Fam}[\overline{\mathbf{M}}] \qquad (\bot)$$

Definition

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

- 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 $\varphi : A \to B \circ f$ 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:

$$\mathbf{Set} \underset{\overline{\mathbf{M}}(I,-)}{\overset{-\odot I}{\rightleftarrows}} \overline{\mathbf{M}} \quad (\bot)$$

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

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:

$$\mathbf{Set} \underset{\overline{\mathbf{M}}(I,-)}{\overset{-\odot I}{\underset{\perp}{\rightleftarrows}}} \overline{\mathbf{M}}$$

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

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 $\mathbf{M}$.
A simpler model for the same language is given by the model of ILL:

$$\mathbf{Set} \underset{\overline{\mathbf{M}}(I,-)}{\overset{-\odot I}{\underset{\perp}{\rightleftarrows}}} \overline{\mathbf{M}}$$

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

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.

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

$$\mathbf{DCPO} \underset{\overline{\mathbf{M}}(I,-)}{\overset{-\odot I}{\underset{\perp}{\rightleftarrows}}} \overline{\mathbf{M}}$$

Original model revisited

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

$$\mathbf{Set} \underset{\mathbf{Fam}[\overline{\mathbf{M}}](I, -)}{\overset{- \odot I}{\rightleftarrows}} \mathbf{Fam}[\overline{\mathbf{M}}] \qquad (\bot)$$

Simpler model:

$$\mathbf{Set} \underset{\overline{\mathbf{M}}(I, -)}{\overset{- \odot I}{\rightleftarrows}} \overline{\mathbf{M}} \qquad (\bot)$$

Question: What does the extra layer of abstraction provide?
Conjecture: A model of the language extended with dependent types, since

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

is a fibration.

Dependent types

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

$$n : \mathbb{N} \vdash \mathrm{NatList}(n) : \mathrm{Type}.$$

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

$$\mathrm{NatList} = (\mathrm{NatList}(n))_{n:\mathbb{N}}.$$

- This is like sets depending on sets, i.e., $S = (S_x)_{x \in X}$ with $X \in \mathbf{Set}$, or equivalently, a pair (X, S) with $S : X \to \mathbf{Set}$ a functor,
- Hence fibrations as $\mathbf{Fam}[\mathbf{Set}] \to \mathbf{Set}$ can be used as models for dependent type theory.

Linear dependent types

Theorem

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

Conjecture

The symmetric monoidal adjunction:

$$\mathbf{Set} \underset{\mathbf{Fam}[\overline{\mathbf{M}}](I,-)}{\overset{-\odot I}{\rightleftarrows}} \mathbf{Fam}[\overline{\mathbf{M}}] \qquad (\bot)$$

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

Remark

If $\mathbf{M} = \mathbf{1}$, *the above model degenerates to* $\mathbf{Fam}[\overline{\mathbf{M}}] = \mathbf{Fam}[\mathbf{M}^{op}, \mathbf{Set}] \cong \mathbf{Fam}[\mathbf{Set}] \simeq [2^{op}, \mathbf{Set}]$, *which is a closed comprehension category and thus a model of intuitionistic dependent type theory*[3].

[2] Matthijs Vákár. *In Search of Effectful Dependent Types.* PhD thesis, University of Oxford.

[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]*.*

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

[5] Michael Shulman. *Enriched Indexed Categories.* Theory and Application of Categories, 2013.

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.*

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

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:

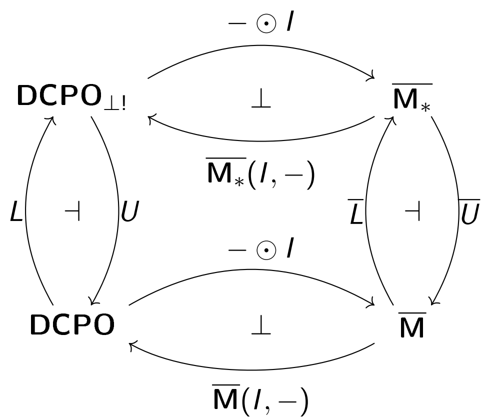
$$\mathbf{DCPO} \underset{\overline{\mathbf{M}}(I,-)}{\overset{-\odot I}{\rightleftarrows}} \overline{\mathbf{M}} \qquad (\perp)$$

    where $\overline{\mathbf{M}} = [\mathbf{M}^{\mathrm{op}}, \mathbf{DCPO}]$, and where the underlying induced (co)monad endofunctors are algebraically compact.
  - Thus, we add partiality to the above model:

$$\mathbf{DCPO}_{\perp!} \underset{\overline{\mathbf{M}_*}(I,-)}{\overset{-\odot I}{\rightleftarrows}} \overline{\mathbf{M}_*} \qquad (\perp)$$

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

Proposed concrete model of Proto-Quipper-M extended with recursion



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.

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:

$$\mathbf{V} \underset{G}{\overset{F}{\rightleftarrows}} \mathbf{L} \quad (F \dashv G)$$

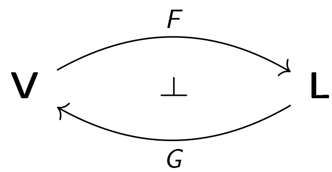
where FG (or equivalently GF) is algebraically compact [6].

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

Abstract model with recursion?

Theorem

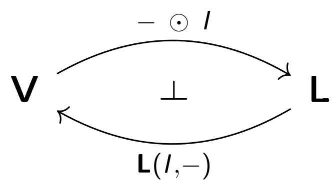
A categorical model of a linear/non-linear lambda calculus extended with recursion is given by a model of ILL:



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:



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?

- Idea: **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 **CFam[C]** and **DCPO**.
- The induced monad and comonad must be algebraically compact.
- The right adjoint of the adjuction must be a representable functor.
- For this reason **CFam[C]** must be **DCPO**-enriched.
- Must have a enriched monoidal fibration **CFam[C]** $\to$ **DCPO** with some extra structure, i.e., comprehension.

Definition **CFam**:

Construction: a generalization of the **CFam**[**DCPO**]-construction[7,8] with **DCPO** replaced by a **DCPO**-enriched category **C**.

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

[8] Bart Jacobs. *Categorical Logic and Type Theory.* 1999.

Definition **CFam**:

Construction: a generalization of the **CFam**[**DCPO**]-construction[7][8] with **DCPO** replaced by a **DCPO**-enriched category **C**.

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

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

[8] Bart Jacobs. *Categorical Logic and Type Theory.* 1999.

Definition **CFam**:

Construction: a generalization of the **CFam**[**DCPO**]-construction[7][8] with **DCPO** replaced by a **DCPO**-enriched category **C**.

- Objects are pairs (X, A) with $X \in \mathbf{DCPO}$ and $A : X \to \mathbf{C}$ 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 $A(x \leq y)^p$;
  - $A(\sup D) = \varinjlim_{d \in D} Ad$ for each directed $D \subseteq X$;

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

[8] Bart Jacobs. *Categorical Logic and Type Theory.* 1999.

Definition **CFam**:

Construction: a generalization of the **CFam**[**DCPO**]-construction[7][8] with **DCPO** replaced by a **DCPO**-enriched category **C**.

- Objects are pairs (X, A) with $X \in \mathbf{DCPO}$ and $A : X \to \mathbf{C}$ 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 $A(x \leq y)^p$;
  - $A(\sup D) = \varinjlim_{d \in D} Ad$ for each directed $D \subseteq X$;
- A morphism $(X, A) \to (Y, B)$ is a pair (f, φ) 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., φ 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.

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

[8] Bart Jacobs. *Categorical Logic and Type Theory*. 1999.

DCPO-enrichment of CFam[C]

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

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

in $\mathbf{C}\big(Ax, Bf(x)\big)$ for each $x \in X$.

DCPO-enrichment of CFam[C]

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

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

in $\mathbf{C}\big(Ax, Bf(x)\big)$ for each $x \in X$.

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

$$f = \sup_{i \in I} f_i$$

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

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

calculated in the dcpo $\mathbf{C}\big(Ax, Bf(x)\big)$ for each $x \in X$;

Monoidal structure

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

Monoidal structure

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

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

with

$$(A \multimap B)f = \oint_{x \in X} Ax \multimap Bf(x),$$

where $\oint$ denotes some kind of 'lax end' satisfying

$$\oint_{x \in X} \mathbf{C}(Fx, Gx) = \{\text{lax natural transformations } F \to G\}$$

for functors $F, G : X \to \mathbf{C}$.

Monoidal structure

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

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

with

$$(A \multimap B)f = \oint_{x \in X} Ax \multimap Bf(x),$$

where $\oint$ denotes some kind of 'lax end' satisfying

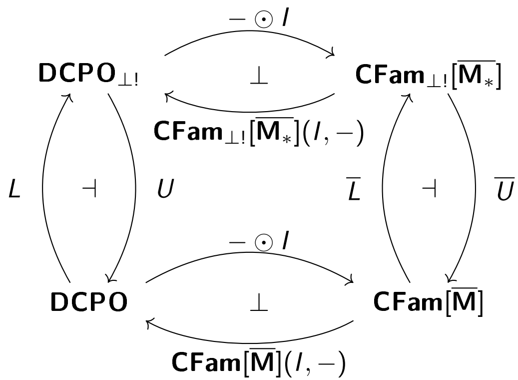
$$\oint_{x \in X} \mathbf{C}(Fx, Gx) = \{\text{lax natural transformations } F \to G\}$$

for functors $F, G : X \to \mathbf{C}$.

Question: what are the requirements on $\mathbf{C}$ to assure the existence of this 'lax end'.

Abstract model with recursion and dependent types?

- This is the most complicated case by far.



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 [9].

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

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.