A DCPO-enriched linear/non-linear model

Bert Lindenhovius
Department of Computer Science
Tulane University
New Orleans, LA, USA

Michael Mislove
Department of Computer Science
Tulane University
New Orleans, LA, USA

Vladimir Zamdzhiev
Department of Computer Science
Tulane University
New Orleans, LA, USA

Rios and Selinger have recently proposed a categorical model for the quantum programming language *Proto-Quipper-M*, which is an important fragment of the Quipper language. In this work, we describe an extension to their categorical model with the additional property that it is **DCPO**-enriched, bringing us closer to modeling general recursion in the language. Similar to their model, our model exhibits a symmetric monoidal adjunction between a cartesian closed category and a monoidal closed category that previously has been shown to be a sound categorical model for a mixed linear/non-linear (and not necessarily quantum) programming language by Benton.

1 Introduction and Background

The impact of Girard's linear logic on computer science and logic led to a search for semantic models that illuminate the relation between intuitionistic linear logic (ILL) and intuitionistic logic (IL). One such model was devised by Benton [3], who modeled ILL as a symmetric monoidal closed category, $\mathscr{L}$, IL as a cartesian closed category, $\mathscr{C}$, with the two categories related by a symmetric monoidal adjunction $F\colon \mathscr{C} \to \mathscr{L} \ \dashv G\colon \mathscr{L} \to \mathscr{C}$. Such linear/non-linear (LNL) models, as he called them, are receiving renewed attention with the emergence of high-level functional quantum programming languages [18, 20]. In this paper we explore one particular approach to building LNL models in which the linear category is $\mathscr{L} = \mathbf{Fam}(\mathbf{M})$, the free coproduct completion of a symmetric monoidal closed product-complete category $\mathbf{M}$ [11]. Alternatively, each object of the category $\mathbf{Fam}(\mathbf{M})$ can be seen as a set-indexed family of objects of $\mathbf{M}$ and each morphism is a set-indexed family of morphisms of $\mathbf{M}$ which compose in an obvious way.

The main result of this paper is to present an analogous construction $\mathbf{DFam}(\mathbf{M})$, which is a **DCPO**-enriched category in which each object is indexed by a directed complete partial order (dcpo). This leads to an adjunction between $\mathbf{DFam}(\mathbf{M})$ and **DCPO**, the category of directed complete partial orders and Scott-continuous maps, which is a step toward a model supporting recursion. An interesting and distinctive feature of our approach is that we do not assume an order structure on the circuit model. Because of the minimal assumptions we impose on $\mathbf{M}$, our construction may apply to a broad range of cases beyond the particular assumptions needed for modeling quantum circuits.

Plan of the paper In the rest of this section, we outline the basic concepts and background we need, first about Benton's LNL models, and then about dcpo's. We also describe related work. Section 2 is devoted to presenting our **DFam** construction and giving its basic properties, and in Section 3 we describe how **DFam** can be used as the linear component of a LNL model. We also give the syntax for a LNL term language from [3], and Benton's basic results show our construction serves as a sound model for this language. Section 4 describes additional properties of **DFam** and gives the precise relationship between our construction and the one used by Rios and Selinger [18], and Section 5 shows **DFam** is a fibration, which allows us to prove it is complete and cocomplete. Section 6 concludes the paper with a

Submitted to:
QPL 2017

© B. Lindenhovius, M. Mislove & V. Zamdzhiev
This work is licensed under the
Creative Commons Attribution License.

summary and a description of future work. Most proofs are elided for lack of space; they will appear in the full version of the paper.

1.1 Linear/non-linear models

If $(\mathscr{C},\otimes,I)$ and $(\mathscr{V},\odot,J)$ are symmetric monoidal categories, then a *symmetric monoidal functor* is a pair (F,m), where $F:\mathscr{C}\to\mathscr{V}$ is a functor and m is a pair of natural transformations $m_{A,B}:FA\odot FB\to F(A\otimes B)$ and $m_I:J\to FI$ satisfying the usual coherence conditions. If all maps $m_{A,B}$ and m_I are isomorphisms, then (F,m) is called a *strong symmetric monoidal functor*. The natural transformation m is called a *comparison map*. If $(F,m),(G,n):(\mathscr{C},\otimes,I)\to(\mathscr{V},\odot,J)$ are monoidal functors, a natural transformation $\alpha:F\to G$ is called *monoidal* if it is compatible with the comparison maps. We refer to [3, Definitions 5–7] or [12, Section 2] for the details.

Definition 1.1 ([4])**.** A linear/non-linear model consists of the following data:

1. a cartesian closed category $(\mathscr{C},1,\times,\to)$;
2. a symmetric monoidal closed category $(\mathscr{L},I,\otimes,\multimap)$
3. a pair of symmetric monoidal functors $(G,n):\mathscr{L}\to\mathscr{C}$ and $(F,m):\mathscr{C}\to\mathscr{L}$ that form a symmetric monoidal adjunction $F\dashv G$.

We will refer to linear/non-linear models $\mathscr{M}=(\mathscr{C},\mathscr{L},F,G)$ as *LNL* models for brevity.

If $F\dashv G$ is a symmetric monoidal adjunction, the underlying functors F and G form an adjunction with the unit $\eta:\mathrm{id}_{\mathscr{C}}\to G\circ F$ and the counit $\varepsilon:F\circ G\to\mathrm{id}_{\mathscr{L}}$ inducing monoidal natural transformations between the corresponding monoidal functors. It has been shown that the symmetric monoidal left adjoint (F,m) in a linear/non-linear model is always strong. The following lemma, which has been stated in greater generality by Kelly in [14, Theorem 1.5], asserts that the converse holds as well, and it considerably reduces the length of a proof that an adjunction is symmetric monoidal.

Lemma 1.2 ([14])**.** Let $(F,m):(\mathscr{V},\odot,J)\to(\mathscr{L},\otimes,I)$ be a strong symmetric monoidal functor. If F has a right adjoint $G:\mathscr{L}\to\mathscr{V}$, then there is a unique comparison map n for G such that $(F,m)\dashv(G,n)$ is a symmetric monoidal adjunction.

Benton [3] also shows that an LNL model gives rise to a strong commutative monad GF on the non-linear category $\mathscr{C}$ and a comonad FG on the linear one $\mathscr{L}$.

1.2 Categories of dcpo's

Let X be a partially ordered set (poset). A non-empty subset $D\subseteq X$ is *directed* if for each $x,y\in D$ there is some $z\in D$ such that $x,y\leq z$. We call X a *directed-complete partial order* (dcpo) if the supremum $\bigvee D$ of each directed subset $D\subseteq X$ exists. An order-preserving map $f:X\to Y$ between dcpo's X and Y is called *Scott continuous* if $f(\bigvee D)=\bigvee f(D)$ for each directed $D\subseteq X$.

We denote the category of dcpo's with Scott-continuous functions by **DCPO**. This category has all limits and colimts and we will now describe its products and coproducts in particular as we will need them later on. For a collection $\{X_i\}_{i\in I}$ of dcpo's, the product $\prod_{i\in I}X_i$ is the set-theoretic product of the X_i equipped with the *product order:* $(x_i)_{i\in I}\leq(y_i)_{i\in I}$ if and only if $x_i\leq y_i$ for each $i\in I$. Since the product order is the coarsest partial order for which the projections maps are order preserving, it follows routinely that $\prod_{i\in I}X_i$ is a dcpo. The coproduct $\coprod_{i\in I}X_i$ is the dcpo whose elements are pairs (i,x) where $i\in I$ and $x\in X_i$, ordered by $(i,x)\leq(j,y)$ if and only if $i=j$ and $x\leq y$. For each $j\in I$, we define an injection map $\mathrm{in}_j:X_j\to\coprod_{i\in I}X_i$ by $x\mapsto(j,x)$. It is obvious that each $z\in\coprod_{i\in I}X_i$ is of the form $\mathrm{in}_j(x)$ for some unique

$j \in I$ and a unique $x \in X_j$. In this case, the partial order is the finest one for which the injections are order preserving, so this defines a dcpo.

DCPO is also a cartesian closed category where the internal hom $[X \to Y]$ consists of all Scott-continuous functions $X \to Y$. We note that its full subcategory $\mathbf{DCPO}_\perp$ of dcpo's with a least element $\perp$ is cartesian closed, but is neither complete nor cocomplete. The subcategory $\mathbf{DCPO}_{\perp!}$ of dcpo's with a least element and *strict* Scott-continuous maps, i.e., maps that preserve the least element, becomes a symmetric monoidal closed category when equipped with the *smash product:* $X \otimes Y = (X \times Y)/R$, where $R = (\{\perp_X\} \times Y) \cup (X \times \{\perp_Y\})$.

We say that a category $\mathscr{C}$ is **DCPO**-*enriched* if $\mathscr{C}(X,Y)$ is a dcpo for each pair of $\mathscr{C}$-objects X and Y, and if in addition for any $\mathscr{C}$-object Z, the composition map $\mathscr{C}(Y,Z) \times \mathscr{C}(X,Y) \to \mathscr{C}(X,Z)$ is Scott continuous. A functor $F : \mathscr{C} \to \mathscr{D}$ between two **DCPO**-enriched categories $\mathscr{C}$ and $\mathscr{D}$ is called *locally continuous* if the assignment

$$\mathscr{C}(X,Y) \to \mathscr{D}(FX,FY), \quad f \mapsto Ff$$

is Scott continuous for each pair of objects X and Y in $\mathscr{C}$.

We refer to [1, 9] for a more detailed exposition of dcpo's.

1.3 Related work

This contribution is based on the model of Rios and Selinger [18] for Proto-Quipper-M, a prototypical high-level functional quantum programming language that embodies the principle of 'quantum data, classical control'. The first functional programming language relying on this principle is *QPL*, which is a first-order quantum programming language equipped with categorical semantics [19]. Alternative semantics for QPL were provided by Cho [6] based on his discovery that the category of W*-algebras with completely positive subunital maps is $\mathbf{DCPO}_{\perp!}$-enriched, a fact that was independently discovered by Rennela [15]. An approach to higher-order quantum computing was proposed by Selinger and Valiron [20]. Traditionally, higher-order programming languages are provided with denotational semantics using domain theory, so higher-order quantum programming languages require the development of an appropriate notion of 'quantum domain theory' for their denotational semantics. The present contribution suggests that linear/non-linear models where **DCPO** is the cartesian category and where the linear category is **DCPO**-enriched might be an appropriate notion. An alternative notion of quantum domains has been proposed by Rennela and Staton [16, 17].

2 Directed Families Construction

In this section we describe what we call the *directed families construction*, which allows us to define a family of LNL models $\mathscr{M}_{\mathbf{M}} = (\mathbf{DCPO}, \mathbf{DFam}(\mathbf{M}), F_{\mathbf{M}}, G_{\mathbf{M}})$, one for each choice of $\mathbf{M}$, provided $\mathbf{M}$ is a symmetric monoidal closed category that also is product-complete. The construction is similar to the standard families construction [5, p. 373],[11], which is done over **Set**, but differs in that our families are indexed by dcpo's and form a **DCPO**-enriched category. For brevity, we refer to these models simply as $\mathscr{M} = (\mathbf{DCPO}, \mathbf{DFam}, F, G)$, and the reader should remember that they are parametrized over $\mathbf{M}$.

The category **DCPO** should be thought of as the category in which we interpret the non-linear (classical) control of our programming language. The category $\mathbf{M}$ should be seen as the category in which we interpret the circuit model ($\mathbf{Q}$) and linear lambda abstractions. Finally, **DFam** allows us to interpret the rest of the linear constructs and can be seen as a category consisting of ordered families of objects of $\mathbf{M}$.

The required structure on $\mathbf{M}$ is not difficult to obtain. For example, $\mathbf{M}$ could be the presheaf category $\mathbf{Set}^{\mathbf{Q}^{\mathrm{op}}}$, where $\mathbf{Q}$ is any symmetric monoidal category. $\mathbf{M}$ is then complete and cocomplete, and becomes a symmetric monoidal closed category when equipped with the Day tensor product [7]. $\mathbf{Q}$ then can be embedded into $\mathbf{M}$ using the Yoneda embedding, which is a strong symmetric monoidal functor. In the context of quantum programming, a good choice for $\mathbf{Q}$ would be **FdStar**, the category of finite-dimensional C*-algebras and completely positive maps (sometimes also assumed to be unital or subunital). However, choosing $\mathbf{Q} = \mathbf{FdHilb}$, the category of finite-dimensional Hilbert spaces with linear maps, also satisfies the requirements. Our **DFam** construction also works with any other symmetric monoidal category, so our model might have applications outside of quantum computing as well.

We recall that objects in the usual set-indexed families construction have the form $(X,(A_x)_{x\in X})$, where X is a set and $A_x \in \mathbf{M}$ for each $x \in X$. A morphism between objects $(X,(A_x)_{x\in X})$ and $(Y,(B_y)_{y\in Y})$ is a pair $(f,(\varphi_x)_{x\in X})$ where $f : X \to Y$ is a function, and $\varphi_x : A_x \to B_{f(x)}$ is a morphism in $\mathbf{M}$. Regarding X as a discrete category, it follows that $f : X \to Y$ becomes a functor, and so do $A : X \to \mathbf{M}$ and $B : Y \to \mathbf{M}$, if we define $Ax = A_x$ and $By = B_y$. Moreover, φ_x forms the x-component of a natural transformation $A \to B \circ f$. This observation leads to the following:

Definition 2.1. The category $\mathbf{DFam}(\mathbf{M})$ (or briefly, **DFam**) has:

- Objects of the form (X,A), where $X \in \mathbf{DCPO}$ and $A : X \to \mathbf{M}$ is a functor, where X is regarded as a poset category;
- Morphisms $(X,A) \to (Y,B)$ of the form (f,φ), where $f : X \to Y$ is a Scott-continuous map, and $\varphi : A \to B \circ f$ is a natural transformation in the functor category $[X,\mathbf{M}]$;
- Composition of morphisms
$$(f,\varphi) : (X,A) \to (Y,B)$$
and
$$(g,\psi) : (Y,B) \to (Z,C)$$
is defined by
$$(g,\psi) \circ (f,\varphi) = (g \circ f, \psi f \circ \varphi),$$
where we recall that if $x \in X$, then the x-component of $\psi f \circ \varphi$ is given by
$$(\psi f \circ \varphi)_x = \psi_{f(x)} \circ \varphi_x.$$

For each $X \in \mathbf{DCPO}$, we define $I_X : X \to \mathbf{M}$ by $I_X(x) = I$, the tensor unit of $\mathbf{M}$, and $I_X(x \leq x') = \mathrm{id}_I$.

Proposition 2.2. **DFam** is a symmetric monoidal closed category if we define
$$(X,A) \otimes (Y,B) = (X \times Y, A \otimes B),$$
where $A \otimes B : X \times Y \to \mathbf{M}$ is defined by $(A \otimes B)(x,y) = A(x) \otimes B(y)$. The tensor unit is given by $(1,I_1)$. Its internal hom set is defined by
$$(X,A) \multimap (Y,B) = ([X \to Y], C),$$
where $C : [X \to Y] \to \mathbf{M}$ is the functor defined by
$$Cf = \prod_{x\in X} (Ax \multimap Bf(x)).$$

Theorem 2.3. **DFam** is **DCPO**-enriched. More specifically, given two **DFam**-objects (X,A) and (Y,B), we define an order $\sqsubseteq$ on $\mathbf{DFam}((X,A),(Y,B))$ by

$$(f,\varphi) \sqsubseteq (g,\psi) \iff f \leq g \text{ and } B\big(f(x) \leq g(x)\big) \circ \varphi_x = \psi_x \text{ for each } x \in X,$$

where $f \leq g$ refers to the pointwise order on the dcpo $[X \to Y]$. The supremum of a directed set $\{(f_i),\varphi_i\}_{i\in D}$ in $\mathbf{DFam}((X,A),(Y,B))$ is given by (f,φ), where $f = \bigvee_{i\in D} f_i$ in $[X \to Y]$, and the x-component of φ is defined by

$$\varphi_x = B\big(f_i(x) \leq f(x)\big) \circ (\varphi_i)_x,$$

and is independent of the choice of $i \in D$.

3 A linear/non-linear model

In this section we show that **DCPO** and **DFam** form a linear/non-linear model which also is **DCPO**-enriched. We begin by showing that there is an adjunction between the two categories:

$$\mathbf{DCPO} \underset{G}{\overset{F}{\perp}} \mathbf{DFam}$$

F is given by:

$$F(X) = (X, I_X)$$
$$F(f) = (f, \iota), \text{ where } \iota_x = \mathrm{id}_I$$

F can also be equipped with natural isomorphisms $m_{X,Y} : FX \otimes FY \to F(X \otimes Y)$ given by $m_{X,Y} = (\mathrm{id}_{X\times Y}, \lambda)$, where $\lambda_{x,y} = \lambda_I$ and $m : (1,I_1) \to F1$ by $m = \mathrm{id}_{(1,I_1)}$.

Lemma 3.1. $(F,m) : \mathbf{DCPO} \to \mathbf{DFam}$ is a strong symmetric monoidal and locally continuous functor.

Proof. Clearly, F is a functor that is strong symmetric monoidal. We prove that it is locally continuous. Given dcpo's X and Y, let $f,g \in [X \to Y]$ such that $f \leq g$. Then $Ff = (f,\iota)$, $Fg = (g,\iota)$ and for each $x \in X$, we have

$$I_Y(f(x) \leq g(x))\iota_x = \mathrm{id}_I \circ \iota_x = \mathrm{id}_I \circ \mathrm{id}_I = \mathrm{id}_I = \iota_x,$$

hence $(f,\iota) \sqsubseteq (g,\iota)$, i.e., $Ff \sqsubseteq Fg$. Now let $(f_i)_{i\in D}$ be directed in $[X \to Y]$ with supremum f. Note that $Ff_i : (X,I_X) \to (Y,I_Y)$ for each $i \in D$. Moreover, $(Ff_i)_{i\in D} = (f_i,\iota)_{i\in D}$ is directed; its supremum is of the form (f,φ), where for each $x \in X$, we have

$$\varphi_x = I_Y(f_i(x) \leq f(x))\iota_x = \mathrm{id}_I \circ \mathrm{id}_I = \mathrm{id}_I = \iota_x.$$

Hence $\bigvee_{i\in D} Ff_i = (f,\iota) = (\bigvee_{i\in D} f_i, \iota) = F\left(\bigvee_{i\in D} f_i\right)$. □

The functor $G : \mathbf{DFam} \to \mathbf{DCPO}$ can be defined by:

$$G(X,A) = \mathbf{DFam}\left((1,I_1),(X,A)\right)$$
$$G(f,\alpha) = \mathbf{DFam}\left((1,I_1),(f,\alpha)\right)$$

Since **DFam** is **DCPO**-enriched, it follows that G is locally continuous. In combination with the previous lemma, this implies that the comonad FG on **DFam** and the monad GF on **DCPO** are locally continuous.

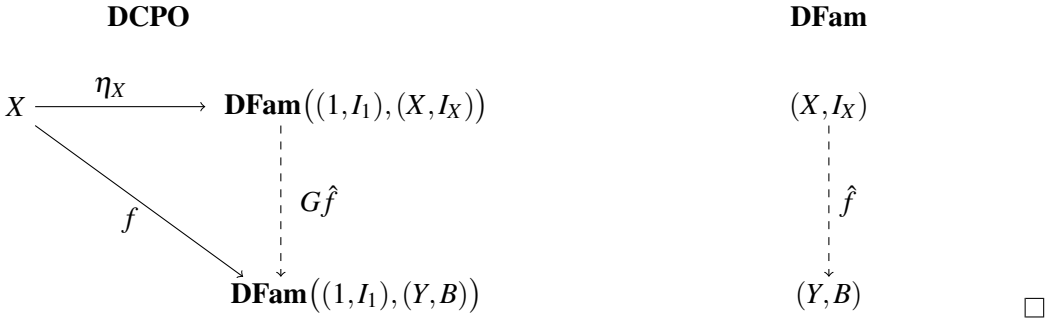
Remark 3.2. Since all elements of $G(X,A)$ are of the form (f,φ), with $f : 1 \to X$ and $\varphi_1 \in \mathbf{M}(I, Af(1))$, we can identify f with $f(1)$, whence it follows that

$$G(X,A) \cong \big(\{(x,\varphi_x) \mid x \in X \text{ and } \varphi_x \in \mathbf{M}(I,Ax)\}, \sqsubseteq\big)$$

where the order is given by $(x_1,\varphi_1) \sqsubseteq (x_2,\varphi_2)$ iff $(x_1 \leq x_2)$ and $A(x_1 \leq x_2) \circ \varphi_1 = \varphi_2$. Then $G(f,\alpha) = (z,\varphi) \mapsto (f(z), \alpha_z \circ \varphi)$.

Lemma 3.3. There exists an adjunction between **DCPO** and **DFam** with $F \dashv G$.

Proof. Using Remark 3.2, we can define the unit η_X to be the map $x \mapsto (x, \mathrm{id}_I)$ and for an arbitrary $f : X \to G(Y,B)$, the unique map $\hat{f}$ is given by $\hat{f} = (\pi_1 \circ f, \varphi)$, where $\varphi_x = \pi_2 \circ f(x)$ for each $x \in X$



□

Theorem 3.4. There exists a unique comparison map n, such that $(F,m) \dashv (G,n)$ is a symmetric monoidal adjunction.

Proof. This follows immediately from Lemma 3.3 and Lemma 3.1 when combined with Lemma 1.2. □

3.1 Modeling an LNL Term Language

Theorem 3.4 shows that we have an LNL model, with the cartesian closed category $\mathscr{C} = \mathbf{DCPO}$ as the non-linear category, the symmetric monoidal closed category $\mathscr{L} = \mathbf{DFam}$ as the linear category, and where the adjunction is given by $F \dashv G$. As Benton shows in [4, 3], this categorical model can be used to interpret an LNL term language, which we now describe.

The type system is given by:

$$A,B := A_0 \mid I \mid A \otimes B \mid A \multimap B \mid FX$$

$$X,Y := X_0 \mid 1 \mid X \times Y \mid X \to Y \mid GA$$

where A,B range over the objects of $\mathscr{L}$; X,Y range over the objects of $\mathscr{C}$; A_0 and X_0 are some sets of constants which may be interpreted in $\mathscr{L}$ or $\mathscr{C}$, respectively.

Then, the LNL term assignment system is given in Figure 1, where a,b,c are used to refer to linear variables; e,f,g,h refer to linear terms; w,x,y,z refer to non-linear variables; s,t,u,v refer to non-linear terms. Γ and Δ range over linear contexts and Θ,Φ ranger over non-linear contexts. A type judgement in the non-linear category is of the general form:

$$s_1 : X_1, \ldots, s_n : X_n \vdash_{\mathscr{C}} t : Y$$

$$\Theta; a\colon A \vdash_{\mathcal{L}} a\colon A \qquad \Theta, x\colon X \vdash_{\mathcal{C}} x\colon X$$

$$\frac{\Theta \vdash_{\mathcal{C}} s\colon X \qquad \Theta \vdash_{\mathcal{C}} t\colon Y}{\Theta \vdash_{\mathcal{C}} (s,t)\colon X \times Y} \qquad \frac{}{\Theta \vdash_{\mathcal{C}} ()\colon 1}$$

$$\frac{\Theta \vdash_{\mathcal{C}} s\colon X \times Y}{\Theta \vdash_{\mathcal{C}} \mathsf{fst}(s)\colon X} \qquad \frac{\Theta \vdash_{\mathcal{C}} s\colon X \times Y}{\Theta \vdash_{\mathcal{C}} \mathsf{snd}(s)\colon Y}$$

$$\frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon A \qquad \Theta; \Delta \vdash_{\mathcal{L}} f\colon B}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} e \otimes f\colon A \otimes B} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon A \otimes B \qquad \Theta; \Delta, a\colon A, b\colon B \vdash_{\mathcal{L}} f\colon C}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} \mathsf{let}\ a \otimes b = e\ \mathsf{in}\ f\colon C}$$

$$\frac{}{\Theta \vdash_{\mathcal{L}} *\colon I} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon I \qquad \Theta; \Delta \vdash_{\mathcal{L}} f\colon A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} \mathsf{let} * = e\ \mathsf{in}\ f\colon A}$$

$$\frac{\Theta, x\colon X \vdash_{\mathcal{C}} s\colon Y}{\Theta \vdash_{\mathcal{C}} (\lambda x\colon X.s)\colon X \to Y} \qquad \frac{\Theta \vdash_{\mathcal{C}} s\colon X \to Y \qquad \Theta \vdash_{\mathcal{C}} t\colon X}{\Theta \vdash_{\mathcal{C}} s\ t\colon Y}$$

$$\frac{\Theta; \Gamma, a\colon A \vdash_{\mathcal{L}} e\colon B}{\Theta; \Gamma \vdash_{\mathcal{L}} (\lambda a\colon A.e)\colon A \multimap B} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon A \multimap B \qquad \Theta; \Delta \vdash_{\mathcal{L}} f\colon A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} e\ f\colon B}$$

$$\frac{\Theta \vdash_{\mathcal{C}} s\colon X}{\Theta \vdash_{\mathcal{L}} \mathsf{F}(s)\colon FX} \qquad \frac{\Theta; \Gamma \vdash_{\mathcal{L}} e\colon FX \qquad \Theta, x\colon X; \Delta \vdash_{\mathcal{L}} f\colon A}{\Theta; \Gamma, \Delta \vdash_{\mathcal{L}} \mathsf{let}\ \mathsf{F}(x) = e\ \mathsf{in}\ f\colon A}$$

$$\frac{\Theta \vdash_{\mathcal{L}} e\colon A}{\Theta \vdash_{\mathcal{C}} \mathsf{G}(e)\colon GA} \qquad \frac{\Theta \vdash_{\mathcal{C}} s\colon GA}{\Theta \vdash_{\mathcal{L}} \mathsf{derelict}(s)\colon A}$$

Figure 1: LNL term assignment system ([3], Fig 8)

whereas a type judgement in the linear category is of the form:

$$s_1 : X_1, \ldots, s_n : X_n, a_1 : A_1, \ldots, a_m : A_m \vdash_{\mathscr{L}} b : B$$

Benton also defines β-reduction rules for the language given in Figure 2, as well as another reduction relation, called *commuting conversion* reduction, whose rules are shown in Figure 3. Both reductions are well-typed and it is shown that LNL models are sound with respect to these reductions.

Theorem 3.5 ([3])**.** Both the β-reductions and the commuting conversions are soundly modeled in any LNL model.

1. If $\Theta;\Gamma \vdash_{\mathscr{L}} e : A$ and $e \to_{\beta,c} e'$, then $[\![\Theta;\Gamma \vdash_{\mathscr{L}} e : A]\!] = [\![\Theta;\Gamma \vdash_{\mathscr{L}} e' : A]\!]$
2. If $\Theta \vdash_{\mathscr{C}} s : X$ and $s \to_{\beta,c} s'$, then $[\![\Theta \vdash_{\mathscr{C}} s : X]\!] = [\![\Theta \vdash_{\mathscr{C}} s' : X]\!]$

4 Additional properties of $\mathscr{M}$

We saw in the previous section that our model $\mathscr{M}$ is an LNL model that allows us to soundly interpret the LNL term calculus of Benton. In this section we list some additional properties of $\mathscr{M}$ and discuss how they allow us to extended the term language.

$$\mathsf{fst}(s,t) \to_\beta s$$
$$\mathsf{snd}(s,t) \to_\beta t$$
$$\mathsf{let}\ a \otimes b = e \otimes f\ \mathsf{in}\ g \to_\beta g[e/a, f/b]$$
$$\mathsf{let} * = *\ \mathsf{in}\ e \to_\beta e$$
$$(\lambda x\colon X.s)\ t \to_\beta s[t/x]$$
$$(\lambda a\colon A.e)\ f \to_\beta e[f/a]$$
$$\mathsf{let}\ \mathsf{F}(x) = \mathsf{F}(s)\ \mathsf{in}\ e \to_\beta e[s/x]$$
$$\mathsf{derelict}(\mathsf{G}(e)) \to_\beta e$$

Figure 2: LNL term calculus β-reductions ([3], Fig 9)

$$\mathsf{let}\ a \otimes b = (\mathsf{let} * = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let} * = e\ \mathsf{in}\ (\mathsf{let}\ a \otimes b = f\ \mathsf{in}\ g)$$
$$\mathsf{let} * = (\mathsf{let} * = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let} * = e\ \mathsf{in}\ (\mathsf{let} * = f\ \mathsf{in}\ g)$$
$$(\mathsf{let} * = e\ \mathsf{in}\ f)\ g \to_c \mathsf{let} * = e\ \mathsf{in}\ (f\ g)$$
$$\mathsf{let}\ F(x) = (\mathsf{let} * = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let} * = e\ \mathsf{in}\ (\mathsf{let}\ F(x) = f\ \mathsf{in}\ g)$$
$$\mathsf{let}\ a \otimes b = (\mathsf{let}\ c \otimes d = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ c \otimes d = e\ \mathsf{in}\ (\mathsf{let}\ a \otimes b = f\ \mathsf{in}\ g)$$
$$\mathsf{let} * = (\mathsf{let}\ a \otimes b = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ a \otimes b = e\ \mathsf{in}\ (\mathsf{let} * = f\ \mathsf{in}\ g)$$
$$(\mathsf{let}\ a \otimes b = e\ \mathsf{in}\ f)\ g \to_c \mathsf{let}\ a \otimes b = e\ \mathsf{in}\ (f\ g)$$
$$\mathsf{let}\ F(x) = (\mathsf{let}\ a \otimes b = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ a \otimes b = e\ \mathsf{in}\ (\mathsf{let}\ F(x) = f\ \mathsf{in}\ g)$$
$$\mathsf{let}\ a \otimes b = (\mathsf{let}\ F(x) = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ F(x) = e\ \mathsf{in}\ (\mathsf{let}\ a \otimes b = f\ \mathsf{in}\ g)$$
$$\mathsf{let} * = (\mathsf{let}\ F(x) = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ F(x) = e\ \mathsf{in}\ (\mathsf{let} * = f\ \mathsf{in}\ g)$$
$$(\mathsf{let}\ F(x) = e\ \mathsf{in}\ f)\ g \to_c \mathsf{let}\ F(x) = e\ \mathsf{in}\ (f\ g)$$
$$\mathsf{let}\ F(y) = (\mathsf{let}\ F(x) = e\ \mathsf{in}\ f)\ \mathsf{in}\ g \to_c \mathsf{let}\ F(x) = e\ \mathsf{in}\ (\mathsf{let}\ F(y) = f\ \mathsf{in}\ g)$$

Figure 3: LNL term calculus commuting conversions ([3], Fig 10)

Theorem 4.1. **DFam** has all coproducts.

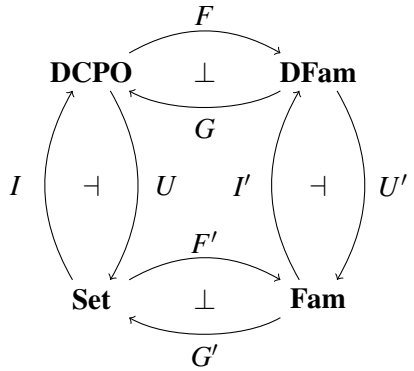
Proof. The initial object in **DFam** is $(\emptyset, I_\emptyset)$. Given a collection of objects (X_i, A_i) indexed by I, then their coproduct is given by the object $(\coprod_{i\in I} X_i, \coprod_{i\in I} A_i)$, where we define

$\coprod_{i\in I} A_i : \coprod_{i\in I} X_i \to \mathbf{M}$
$\coprod_{i\in I} A_i(z) = A_j(x)$, if $z = \mathrm{in}_j(x)$, for some $x \in X_j$
$\coprod_{i\in I} A_i(z \leq z') = A_j(x \leq x')$, if $z = \mathrm{in}_j(x)$, for some $x \in X_j$ and $z' = \mathrm{in}_j(x')$, for some $x' \in X_j$

The functor $\coprod_{i\in I} A_i$ is well-defined on morphisms, because $z \leq z'$ in the coproduct object in **DCPO** implies that z and z' are in the image of the same injection. Finally, for an object (X_j, A_j) in the family, its injection map is then $(\mathrm{in}_j, \mathrm{id})$, where id is the identity natural transformation $\mathrm{id}_x : A_j(x) \to A_j(x)$. □

Thus, the category **DFam** inherits coproducts from **DCPO**. This allows us to extend our type system by adding sum types and to extend the term language of Section 3.1 by adding support for conditional branching. Then, the categorical interpretation is done in the standard way. The model proposed by Rios and Selinger [18] also has this property, because **Fam** is the free coproduct completion of **M**.

The following diagram describes the relationship between their model (bottom adjunction) and the model which we have described in this paper (top adjunction):



where all the adjunctions are symmetric monoidal and where **DFam** and **Fam** are defined over the same category **M**. The functors I, I' are the obvious inclusion functors, whereas the functors U, U' are the obvious forgetful functors. The adjoint functors F' and G' can be equivalently defined by:

$$F' = U' \circ F \circ I$$

$$G' = U \circ G \circ I'$$

The composition of the left adjunction with the top one commutes with the composition of the other two adjunctions, so we also get another LNL model between **Set** and **DFam**.

In addition, as shown in Theorem 2.3, and Section 3 our model $\mathscr{M}$ is **DCPO**-enriched in the sense that both the linear and non-linear categories are **DCPO**-enriched and moreover, the adjoint functors F and G are both locally continuous.

5 Limits, colimits and fibrations

It turns out that the functor $\Phi : \mathbf{DFam} \to \mathbf{DCPO}$, $(X, A) \mapsto X$ is a *fibration*. This is a very useful fact, since the theory of fibrations (which are discussed in detail in [5, 10, 13, 21]) allows us to prove:

Theorem 5.1. Recall that $\mathbf{DFam} = \mathbf{DFam}(\mathbf{M})$.

(1) If $\mathbf{M}$ has all limits, so does **DFam**.

(2) If $\mathbf{M}$ has all colimits, so does **DFam**.

This theorem generalizes Theorem 4.1, but at the cost of imposing extra conditions on $\mathbf{M}$, which are not required for Theorem 4.1.

The rest of this section is devoted to the outline of a proof of Theorem 5.1. We begin with some terminology, starting with a functor $\Phi : \mathscr{A} \to \mathscr{X}$. Given an object $X \in \mathscr{X}$, we define the *fiber* of Φ over X as the subcategory $\mathscr{A}_X$ of $\mathscr{A}$ whose objects are the objects $A \in \mathscr{A}$ such that $\Phi(A) = X$, and whose morphisms are $\mathscr{A}$-morphisms $\varphi : A \to B$ such that $\Phi(\varphi) = \mathrm{id}_X$. The next lemma is immediate:

Lemma 5.2. Let $\Phi : \mathbf{DFam} \to \mathbf{DCPO}$ be given by $(X,A) \mapsto X$ on objects, and $(f,\varphi) \mapsto f$ on morphisms. Then the objects of $\mathbf{DFam}_X$ are of the form (X,A), where $A : X \to \mathbf{M}$ is a functor; all morphisms of $\mathbf{DFam}_X$ are of the form $(\mathrm{id}_X, \varphi) : (X,A) \to (X,B)$, where $\varphi : A \to B$ is a natural transformation.

Definition 5.3. Given $\Phi : \mathscr{A} \to \mathscr{X}$, let $f : X \to Y$ be a morphism in $\mathscr{X}$. Then the morphism $\varphi : A \to B$ in $\mathscr{A}$ is *cartesian over* f if

(1) $\Phi(\varphi) = f$;

(2) If $\psi : C \to B$ is a morphism in $\mathscr{A}$ such that $\Phi(\psi) = f \circ h$ for some morphism in $\mathscr{X}$, then there is a unique morphism $\theta : C \to A$ in $\mathscr{A}$ such that $\Psi(\theta) = h$ and $\psi = \varphi \circ \theta$.

Thus a morphism $\varphi : A \to B$ over f is cartesian if any other morphism with codomain B factorizes through φ in a unique way if the corresponding morphism in the base category factorizes through f.

Definition 5.4. $\Phi : \mathscr{A} \to \mathscr{X}$ is called a *fibration* if for each morphism $f : X \to Y$ in $\mathscr{X}$ and each $B \in \mathscr{A}_Y$, there exists a cartesian morphism $\varphi : A \to B$ over f. Φ is called a *opfibration* if $\Phi^{\mathrm{op}} : \mathscr{A}^{\mathrm{op}} \to \mathscr{X}^{\mathrm{op}}$ is a fibration. If Φ is both a fibration and an opfibration, then Φ is called a *bifibration*.

The intuition behind a fibration $\mathscr{A} \to \mathscr{X}$ is that it allows us to regard $\mathscr{A}$ as a category consisting of categories $\mathscr{A}_X$ indexed by objects $X \in \mathscr{X}$, and glued together by cartesian morphisms.

Proposition 5.5. The functor Φ in Lemma 5.2 is a fibration. More specifically, let $f : X \to Y$ be a Scott-continuous function between dcpo's X and Y. Let $(Y,B) \in \mathbf{DFam}$. Then $(f, \mathrm{id}_{Bf}) : (X, B \circ f) \to (Y,B)$ is cartesian over f.

Let $\Phi : \mathscr{A} \to \mathscr{X}$ be a fibration, and $f : X \to Y$ a morphism in $\mathscr{X}$. Given $B \in \mathscr{A}_Y$, and cartesian morphisms $\varphi : A \to B$, $\psi : C \to B$ over f, the objects A and C are isomorphic [5, Lemma 8.1.4]. It follows that the cartesian morphism $\varphi : A \to B$ in the definition of a fibration is unique up to isomorphism. If we choose for each $B \in \mathscr{A}_Y$ a specific cartesian morphism $\varphi : A \to B$, then the assignment $B \mapsto A$ defines a functor $f^* : \mathscr{A}_Y \to \mathscr{A}_X$. For our particular fibration $\Phi : \mathbf{DFam} \to \mathbf{DCPO}$, we can define f^* as the functor $(Y,B) \mapsto (X, B \circ f)$. If (id_Y, φ) is a morphism in $\mathbf{DFam}_Y$, then $f^*(\mathrm{id}_Y, \varphi) = (\mathrm{id}_X, \varphi f)$. Lemma 5.2 now allows us to describe $\mathbf{DFam}_X$ and f^* as follows:

Lemma 5.6. Given the fibration Φ defined in Lemma 5.2, the fiber $\mathbf{DFam}_X$ is equivalent to the category $[X, \mathbf{M}]$ of functors $X \to \mathbf{M}$. If $f : X \to Y$ is Scott continuous, then $f^* : \mathbf{DFam}_Y \to \mathbf{DFam}_X$ corresponds to the functor $[Y, \mathbf{M}] \to [X, \mathbf{M}]$, $B \mapsto B \circ f$.

It follows from the previous lemma that the fibers of **DFam** have all limits if $\mathbf{M}$ has all limits. Similarly, the fibers have all colimits if $\mathbf{M}$ has all colimits. This fact will be useful if we want to apply the following proposition, which can be found in [10] as Theorem 4.2. A more accessible source is [13, Propositions 9.2.1 & 9.2.2], where special cases are stated involving finite products and coproducts instead of limits and colimits.

Proposition 5.7. Let $\Phi : \mathscr{A} \to \mathscr{X}$ be a fibration.

(1) $\mathscr{A}$ is complete and Φ preserves all limits if and only if $\mathscr{X}$ and all fibers $\mathscr{A}_X$ are complete.

(2) If Φ is a bifibration, then $\mathscr{A}$ is cocomplete and Φ preserves all colimits if and only if $\mathscr{X}$ and all fibers $\mathscr{A}_X$ are cocomplete.

Proof of Theorem 5.1: Since **DCPO** has all limits, it follows that **DFam** has all limits if **M** has all limits, which proves the first statement in Theorem 5.1. **DCPO** also has all colimits, hence the proposition also assures that **DFam** has all colimits if **M** has all colimits and $\Phi : \mathbf{DFam} \to \mathbf{DCPO}$ is a bifibration. It has been shown in [21, Proposition 3.7] that a fibration $\Phi : \mathscr{A} \to \mathscr{X}$ is a bifibration if all functors $f^* : \mathscr{A}_Y \to \mathscr{A}_X$ have a left adjoint $f_! : \mathscr{A}_X \to \mathscr{A}_Y$. Given the fibration $\Phi : \mathbf{DFam} \to \mathbf{DCPO}$, and a Scott-continuous function $f : X \to Y$, cocompleteness of **M** (which we already need in order to apply the second statement in Proposition 5.7) assures the existence of $f_! : [X,\mathbf{M}] \to [Y,\mathbf{M}]$. Concretely, given $A \in [X,\mathbf{M}]$, we define $f_!A \in [Y,\mathbf{M}]$ as a colimit: $(f_!A)y = \varinjlim_{x \in f^{-1}[\downarrow y]} Ax$ for each $y \in Y$. This proves the second statement in Theorem 5.1. □

Remark 5.8. We note that Theorem 5.1 can be formulated for the category **Fam** in [18] by observing that $\Phi : \mathbf{Fam} \to \mathbf{Set}$, $(X,(A_x)_{x\in X}) \mapsto X$ also is a fibration. If **M** is cocomplete, then Φ is a bifibration.

Remark 5.9. Since $\mathbf{DCPO}_\perp$ is not (co)complete, Proposition 5.7 implies that if there is a linear/non-linear model with $\mathbf{DCPO}_\perp$ as the non-linear category, $\mathscr{L}$ as the linear category, and for which there is a fibration $\mathscr{L} \to \mathbf{DCPO}_\perp$, then either $\mathscr{L}$ is not (co)complete or the fibration does not preserve all existing limits and colimits.

6 Conclusion and Future Work

Our main contribution is an extension of the families construction $\mathbf{Fam}(\mathbf{M})$ to the category $\mathbf{DFam}(\mathbf{M})$, which is a **DCPO**-enriched category whose objects are **DCPO**-indexed families of objects of **M**. The impetus for this work has been to extend the linear/non-linear model of Proto-Quipper-M being developed by Rios and Selinger [18] so that the "classical parameters" form a dcpo, which in principle would allow extending their model to accommodate general recursion. But, as we noted in Section 2, the conditions required on **M** for our construction apply more generally than those needed for models of high-level functional quantum programming languages, and we suspect **DFam** may find other applications. In particular, we do not need to assume that our circuit model has an order structure which is the distinguishing feature of our model.

Extending the categorical model of Proto-Quipper-M [18] to support general recursion is an open problem. Our proposed model is **DCPO**-enriched and we believe it is an important step towards solving this problem. As future work we hope to identify classes of algebraically compact endofunctors [8, 2] on **DFam** that can be used to extend our model to support recursive types and recursive terms.

In our approach, the order on **DFam** is inherited from the category **DCPO**. It also is worth investigating whether one can define an order on **DFam** by combining the order on **DCPO** with an order defined on **M** so that we still get an LNL model.

Acknowledgements. We gratefully acknowledge support of the MURI project "Semantics, Formal Reasoning, and Tool Support for Quantum Programming" sponsored by the U.S. Department of Defense and the U.S. Air Force Office of Scientific Research. We would like to thank Peter Selinger and Francisco Rios for introducing us to their model of Proto-Quipper-M. We also wish to thank Mathys Rennela, Chris Heunen and Kenta Cho for stimulating discussions, and Bart Jacobs for pointing out relevant references on fibrations. This work was done in part while the authors were participating in the program on Logical Structures for Computation at the Simons Institute for the Theory of Computing at UC Berkeley.

References

[1] S. Abramsky & A. Jung (1994): *Handbook of Logic in Computer Science (Vol. 3)*. chapter Domain Theory, Oxford University Press, Oxford, UK, pp. 1–168.

[2] M. Barr (1992): *Algebraically compact functors*. Journal of Pure and Applied Algebra 82(3), pp. 211 – 231, doi:10.1016/0022-4049(92)90169-G.

[3] P.N. Benton (1994): *A mixed linear and non-linear logic: Proofs, terms and models*. Available at `http://www.cl.cam.ac.uk/~am21/grants/lomaps/pnbpapers/mixed3.ps.gz`. Technical Report.

[4] P.N. Benton (1995): *A mixed linear and non-linear logic: Proofs, terms and models*. In: Computer Science Logic: 8th Workshop, CSL '94 Kazimierz, Poland, September 25–30, 1994 Selected Papers.

[5] F. Borceux (1994): *Handbook of Categorical Algebra 2: Categories and Structures*. Cambridge University Press.

[6] K. Cho (2016): *Semantics for a Quantum Programming Language by Operator Algebras*. New Generation Comput. 34(1-2), pp. 25–68, doi:10.1007/s00354-016-0204-3.

[7] B. Day (1974): *On closed categories of functors II*. In: Category Seminar: Proceedings Sydney Category Theory Seminar 1972/1973, pp. 20–54, doi:10.1007/BFb0063098.

[8] P. Freyd (1991): *Algebraically complete categories*. In: Category Theory: Proceedings of the International Conference held in Como, Italy, July 22–28, 1990, pp. 95–104, doi:10.1007/BFb0084215.

[9] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. W. Mislove & D. S. Scott (2003): *Continuous lattices and domains*. Cambridge University Press.

[10] J.W. Gray (1966): *Fibred and Cofibred Categories*. In: Proceedings of the Conference on Categorical Algebra, La Jolla 1965, Springer Verlag, Berlin, Heidelberg, pp. 21–83.

[11] H. Hu & W. Tholen (1995): *Limits in free coproduct completions*. Journal of Pure and Applied Algebra 105(3), pp. 277 – 291, doi:10.1016/0022-4049(94)00153-7.

[12] G.B. Im & G.M. Kelly (1986): *A universal property of the convolution monoidal structure*. Journal of Pure and Applied Algebra 43, pp. 75–88.

[13] B. Jacobs (1999): *Categorical Logic and Type Theory*. Elsevier Science.

[14] G.M. Kelly (1974): *Doctrinal adjunction*. In: Category Seminar: Proceedings Sydney Category Theory Seminar 1972/1973, pp. 257–280, doi:10.1007/BFb0063105.

[15] M. Rennela (2014): *Towards a quantum domain theory: order-enrichment and fixpoints in W*-algebras*. In: MFPS 30, Electronic Notes in Theoretical Computer Science 308, pp. 289–307.

[16] M. Rennela & S. Staton (2015): *Completely Positivity and Natural Representation of Quantum Computation*. Electronic Notes in Theoretical Computer Science 319, pp. 369–385.

[17] M. Rennela & S. Staton (2017): *Classical control and quantum circuits in enriched category theory*. Available at `http://www.cs.ru.nl/M.Rennela/papers/ewire.pdf`.

[18] F. Rios & P. Selinger (2017): *A Categorical Model for a Quantum Circuit Description Language*. Submitted to QPL 2017.

[19] P. Selinger (2004): *Towards a quantum programming language*. Mathematical Structures in Computer Science 14(4), pp. 527–586.

[20] P. Selinger & B. Valiron (2006): *A lambda calculus for quantum computation with classical control*. Mathematical Structures in Computer Science 16(3), pp. 527–552.

[21] Michael Shulman (2008): *Framed Bicategories and Monoidal Fibrations*. Theory and Applications of Categories 20(18), pp. 650–738.