Presheaf models of quantum computation: an outline

Octavio Malherbe[1], Philip Scott[2], and Peter Selinger[3]

[1] IMERL-FING, Universidad de la República, Montevideo, Uruguay
malherbe@fing.edu.uy
[2] Dept. of Mathematics and Statistics, University of Ottawa, Canada
phil@site.uottawa.ca
[3] Dept. of Mathematics and Statistics, Dalhousie University, Halifax, Canada
selinger@mathstat.dal.ca

Abstract. This paper outlines the construction of categorical models of higher-order quantum computation. We construct a concrete denotational semantics of Selinger and Valiron's quantum lambda calculus, which was previously an open problem. We do this by considering presheaves over appropriate base categories arising from first-order quantum computation. The main technical ingredients are Day's convolution theory and Kelly and Freyd's notion of continuity of functors. We first give an abstract description of the properties required of the base categories for the model construction to work. We then exhibit a specific example of base categories satisfying these properties.

1 Introduction

Quantum computing is based on the laws of quantum physics. While no actual general-purpose quantum computer has yet been built, research in the last two decades indicates that quantum computers would be vastly more powerful than classical computers. For instance, Shor [34] proved in 1994 that the integer factoring problem can be solved in polynomial time on a quantum computer, while no efficient classical algorithm is known.

Logic has played a key role in the development of classical computation theory, starting with the foundations of the subject in the 1930's by Church, Gödel, Turing, and Kleene. For example, the pure untyped lambda calculus, one of the first models of computation invented by Church, can be simultaneously regarded as a prototypical functional programming language as well as a formalism for denoting proofs. This is the so-called *proofs-as-programs* paradigm. Indeed, since the 1960's, various systems of typed and untyped lambda calculi have been developed, which on the one hand yield proofs in various systems of constructive and/or higher-order logic, while on the other hand denoting functional programs. Modern programming languages such as ML, Haskell, and Coq are often viewed in this light.

Recent research by Selinger, Valiron, and others [30,33] in developing "quantum lambda calculi" has shown that Girard's *linear logic* [12] is a logical system that corresponds closely to the demands of quantum computation. Linear logic, a resource sensitive logic, turns out to formalize one of the central principles of quantum physics, the so-called *no-cloning property*, which asserts that a given unknown quantum state cannot be replicated. This property is reflected on the logical side by the requirement that a given logical assumption (or “resource”) can only be used once. However, until now, the correspondence between linear logic and quantum computation has mainly been explored at the syntactic level.

In this paper we construct mathematical (semantic) models of higher-order quantum computation. The basic idea is to start from existing low level models, such as the category of superoperators, and to use a Yoneda type presheaf construction to adapt and extend these models to a higher order quantum setting. To implement the latter, we use Day’s theory of monoidal structure in presheaf categories, as well as the Freyd-Kelly theory of continuous functors, to lift the required quantum structure [6,11]. Finally, to handle the probabilistic aspects of quantum computation, we employ Moggi’s computational monads [24].

Our model construction depends on a sequence of categories and functors $\mathcal{B} \to \mathcal{C} \to \mathcal{D}$, as well as a collection Γ of cones in $\mathcal{D}$. We use this data to obtain a pair of adjunctions

$$\begin{tikzcd}[\mathcal{B}^{op}, \mathbf{Set}] \arrow[r, shift left, "L"] \arrow[r, phantom, "\perp"] & {[\mathcal{C}^{op}, \mathbf{Set}]} \arrow[l, shift left, "\Phi^*"] \arrow[r, shift left, "F"] \arrow[r, phantom, "\perp"] & {[\mathcal{D}^{op}, \mathbf{Set}]_{\Gamma}} \arrow[l, shift left, "G"]\end{tikzcd}$$

in which the left-hand adjunction gives an appropriate categorical model of the underlying linear logic, and the right-hand adjunction gives a Moggi monad for probabilistic effects. We then give sufficient conditions on $\mathcal{B} \to \mathcal{C} \to \mathcal{D}$ and Γ so that the resulting structure is a model of the quantum lambda calculus. One can describe various classes of concrete models by appropriate choices of diagrams $\mathcal{B} \to \mathcal{C} \to \mathcal{D}$ and cones Γ.

In this paper, we focus on the categorical aspects of the model construction. Thus, we will not review the syntax of the quantum lambda calculus itself (see [30] and [33] for a quick review). Instead, we take as our starting point Selinger and Valiron’s definition of a *categorical model of the quantum lambda calculus* [33]. It was proven in [33] that the quantum lambda calculus forms an internal language for the class of such models. This is similar to the well-known interplay between typed lambda calculus and cartesian closed categories [19]. What was left open in [33] was the construction of a concrete such model (other than the one given by the syntax itself). This is the question whose answer we sketch here. Further details can be found in the first author’s PhD thesis [22].

2 Categories of completely positive maps and superoperators

We first recall various categories of finite dimensional Hilbert spaces that we use in our study. Let V be a finite dimensional Hilbert space, i.e., a finite dimensional complex inner product space. We write $\mathcal{L}(V)$ for the space of linear functions $\rho : V \longrightarrow V$.

Definition 2.1. Let V, W be finite dimensional Hilbert spaces. A linear function $F : \mathcal{L}(V) \longrightarrow \mathcal{L}(W)$ is said to be *completely positive* if it can be written in the form $F(\rho) = \sum_{i=1}^{m} F_i \rho F_i^{\dagger}$, where $F_i : V \longrightarrow W$ is a linear function and $F_i^{\dagger}$ denotes the linear adjoint of F_i for $i = 1, \dots, m$.

Definition 2.2. The category $\mathbf{CPM}_s$ of *simple completely positive maps* has finite dimensional Hilbert spaces as objects, and the morphisms $F : V \longrightarrow W$ are completely positive maps $F : \mathcal{L}(V) \longrightarrow \mathcal{L}(W)$.

Definition 2.3. The category **CPM** of *completely positive maps* is defined as $\mathbf{CPM} = \mathbf{CPM}_s^{\oplus}$, the biproduct completion of $\mathbf{CPM}_s$. Specifically, the objects of **CPM** are finite sequences $(V_1, \dots, V_n)$ of finite-dimensional Hilbert spaces, and a morphism $F : (V_1, \dots, V_n) \longrightarrow (W_1, \dots, W_m)$ is a matrix (F_{ij}), where each $F_{ij} : V_j \longrightarrow W_i$ is a completely positive map. Composition is defined by matrix multiplication.

Remark 2.4. The category **CPM** is the same (up to equivalence) as the category **W** of [28] and the category $\mathbf{CPM}(\mathbf{FdHilb})^{\oplus}$ of [29].

Note that for any two finite dimensional Hilbert spaces V and W, there is a canonical isomorphism $\varphi_{V,W} : \mathcal{L}(V \otimes W) \longrightarrow \mathcal{L}(V) \otimes \mathcal{L}(W)$.

Remark 2.5. The categories $\mathbf{CPM}_s$ and **CPM** are symmetric monoidal. For $\mathbf{CPM}_s$, the tensor product is given on objects by the tensor product of Hilbert spaces $V \bar{\otimes} W = V \otimes W$, and on morphisms by the following induced map $f \bar{\otimes} g := \mathcal{L}(V \otimes W) \xrightarrow{\varphi_{V,W}} \mathcal{L}(V) \otimes \mathcal{L}(W) \xrightarrow{f \otimes g} \mathcal{L}(X) \otimes \mathcal{L}(Y) \xrightarrow{\varphi_{X,Y}^{-1}} \mathcal{L}(X \otimes Y)$. The remaining structure (units, associativity, symmetry maps) is inherited from Hilbert spaces. Similarly, for the symmetric monoidal structure on **CPM**, define $(V_i)_{i \in I} \otimes (W_j)_{j \in J} = (V_i \otimes W_j)_{i \in I, j \in J}$. This extends to morphisms in an obvious way. For details, see [28].

Definition 2.6. We say that a linear map $F : \mathcal{L}(V) \to \mathcal{L}(W)$ is *trace preserving* when it satisfies $\mathrm{tr}_W(F(\rho)) = \mathrm{tr}_V(\rho)$ for all positive $\rho \in \mathcal{L}(V)$. F is called *trace non-increasing* when it satisfies $\mathrm{tr}_W(F(\rho)) \leqslant \mathrm{tr}_V(\rho)$ for all positive $\rho \in \mathcal{L}(V)$.

Definition 2.7. A linear map $F : \mathcal{L}(V) \to \mathcal{L}(W)$ is called a *trace preserving superoperator* if it is completely positive and trace preserving, and it is a *trace non-increasing superoperator* if it is completely positive and trace non-increasing.

Definition 2.8. A completely positive map $F : (V_1, \dots, V_n) \longrightarrow (W_1, \dots, W_m)$ in the category **CPM** is called a *trace preserving superoperator* if for all j and all positive $\rho \in \mathcal{L}(V_j)$, $\sum_i \mathrm{tr}(F_{ij}(\rho)) = \mathrm{tr}(\rho)$, and a *trace non-increasing superoperator* if for all j and all positive $\rho \in \mathcal{L}(V_j)$, $\sum_i \mathrm{tr}(F_{ij}(\rho)) \leqslant \mathrm{tr}(\rho)$.

We now define four symmetric monoidal categories of superoperators. All of them are symmetric monoidal subcategories of **CPM**.

Definition 2.9.

- $\mathbf{Q}_s$ and $\mathbf{Q}'_s$ have the same objects as $\mathbf{CPM}_s$, and $\mathbf{Q}$ and $\mathbf{Q}'$ have the same objects as $\mathbf{CPM}$.
- The morphisms of $\mathbf{Q}_s$ and $\mathbf{Q}$ are trace non-increasing superoperators, and the morphisms of $\mathbf{Q}'_s$ and $\mathbf{Q}'$ are trace preserving superoperators.

Remark 2.10. The categories $\mathbf{Q}_s$, $\mathbf{Q}$, $\mathbf{Q}'_s$, and $\mathbf{Q}'$ are all symmetric monoidal. The symmetric monoidal structure is inherited from $\mathbf{CPM}_s$ and $\mathbf{CPM}$, respectively, and it is easy to check that all the structural maps are trace preserving.

Lemma 2.11. $\mathbf{Q}$ *and* $\mathbf{Q}'$ *have finite coproducts.*

Proof. The injection/copairing maps are as in $\mathbf{CPM}$ and are trace preserving.

3 Presheaf models of a quantum lambda calculus

Selinger defined an elementary quantum flow chart language in [28], and gave a denotational model in terms of superoperators. This axiomatic framework captures the behaviour and interconnection between the basic concepts of quantum computation (for example, the manipulation of quantum bits under the basic operations of measurement and unitary transformation) in a lower-level language. In particular, the semantics of this framework is very well understood: each program corresponds to a concrete superoperator.

Higher-order functions are functions that can input or output other functions. In order to deal with such functions, Selinger and Valiron introduced, in a series of papers [31,32,33], a typed lambda calculus for quantum computation and investigated several aspects of its semantics. In this context, they combined two well-established areas: the intuitionistic fragment of Girard's linear logic [12] and Moggi's computational monads [24].

The type system of Selinger and Valiron's quantum lambda calculus is based on intuitionistic linear logic. As is usual in linear logic, the logical rules of weakening and contraction are introduced in a controlled way by an operator "!" called "of course" or "exponential". This operator creates a bridge between two different kinds of computation. More precisely, a value of a general type A can only be used once, whereas a value of type $!A$ can be copied and used multiple times. As mentioned in the introduction, the impossibility of copying quantum information is one of the fundamental differences between quantum information and classical information, and is known as the *no-cloning property.* From a logical perspective, this is related to the failure of the contraction rule; thus it seems natural to use linear logic in discussing quantum computation. It is also well known that categorically, the operator "!" satisfies the properties of a *comonad* (see [23]).

Since the quantum lambda calculus has higher-order functions, as well as probabilistic operations (namely measurements), it must be equipped with an evaluation strategy in order to be consistent. Selinger and Valiron addressed this by choosing the *call-by-value* evaluation strategy. This introduces a distinction between *values* and *computations*. At the semantic level, Moggi [24] proposed using the notion of monad as an appropriate tool for interpreting computational behaviour. In our case, this will be a strong monad.

So let us now describe Selinger and Valiron's notion of a *categorical model of the quantum lambda calculus* [33].

3.1 Categorical models of the quantum lambda calculus

In what follows, let $(\mathcal{C}, \otimes, I, \alpha, \rho, \lambda, \sigma)$ be a symmetric monoidal category [21].

Definition 3.1. A *symmetric monoidal comonad* $(!, \delta, \varepsilon, m_{A,B}, m_I)$ is a comonad $(!, \delta, \varepsilon)$ where the functor ! is a monoidal functor $(!, m_{A,B}, m_I)$, i.e., with natural transformations $m_{A,B} : !A \otimes !B \longrightarrow !(A \otimes B)$ and $m_I : I \longrightarrow !I$, satisfying appropriate coherence axioms [16] such that δ and ε are symmetric monoidal natural transformations.

Definition 3.2. A *linear exponential comonad* is a symmetric monoidal comonad $(!, \delta, \varepsilon, m_{A,B}, m_I)$ such that for every $A \in \mathcal{C}$, there exists a commutative comonoid (A, d_A, e_A) satisfying some technical requirements (see [4,33]).

Definition 3.3. Let (T, η, μ) be a strong monad on $\mathcal{C}$. We say that $\mathcal{C}$ has *Kleisli exponentials* if there exists a functor $[-,-]_k : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$ and a natural isomorphism: $\mathcal{C}(A \otimes B, TC) \cong \mathcal{C}(A, [B, C]_k)$.

Definition 3.4 (Selinger and Valiron [33]). A *linear category for duplication* consists of a symmetric monoidal category $(\mathcal{C}, \otimes, I)$ with the following structure:

- an idempotent, strongly monoidal, linear exponential comonad $(!, \delta, \varepsilon, d, e)$,
- a strong monad (T, μ, η, t) such that $\mathcal{C}$ has Kleisli exponentials.

Further, if the unit I is a terminal object we shall speak of an *affine linear category for duplication.*

Remark 3.5. Perhaps surprisingly, following the work of Benton, a linear category for duplication can be obtained from a structure that is much easier to describe, namely, a pair of monoidal adjunctions [2,23,17]

$$\begin{tikzcd}
(\mathcal{B}, \times, 1) \arrow[r, shift left, "(L,l)"] \arrow[r, phantom, "\perp"] & (\mathcal{C}, \otimes, I) \arrow[l, shift left, "(I,i)"] \arrow[r, shift left, "(F,m)"] \arrow[r, phantom, "\perp"] & (\mathcal{D}, \otimes, I), \arrow[l, shift left, "(G,n)"]
\end{tikzcd}$$

where the category $\mathcal{B}$ has finite products and $\mathcal{C}$ and $\mathcal{D}$ are symmetric monoidal closed. The monoidal adjoint pair of functors on the left represents a linear-non-linear model of linear logic in the sense of Benton [2], in which we obtain a monoidal comonad by setting $! = L \circ I$. The monoidal adjoint pair on the right gives rise to a strong monad $T = G \circ F$ in the sense of Kock [16,17], which is also a computational monad in the sense of Moggi [24].

We now state the main definition of a model of the quantum lambda calculus.

Definition 3.6 (Models of the quantum lambda calculus [33]). An *abstract model of the quantum lambda calculus* is an affine linear category for duplication $\mathcal{C}$ with finite coproducts, preserved by the comonad !. A *concrete model of the quantum lambda calculus* is an abstract model such that there exists a full and faithful embedding $\mathbf{Q} \hookrightarrow \mathcal{C}_T$, preserving tensor $\otimes$ and coproduct $\oplus$ up to isomorphism, from the category $\mathbf{Q}$ of norm non-increasing superoperators (see Definition 2.9) into the Kleisli category of the monad T.

Remark 3.7. To make the connection to quantum lambda calculus: the category $\mathcal{C}$, the Kleisli category $\mathcal{C}_T$, and the co-Kleisli category $\mathcal{C}_!$ all have the same objects, which correspond to *types* of the quantum lambda calculus. The morphisms $f : A \longrightarrow B$ of $\mathcal{C}$ correspond to *values* of type B (parameterized by variables of type A). A morphism $f : A \longrightarrow B$ in $\mathcal{C}_T$, which is really a morphism $f : A \longrightarrow TB$ in $\mathcal{C}$, corresponds to a *computation* of type B (roughly, a probability distribution of values). Finally, a morphism $f : A \longrightarrow B$ in $\mathcal{C}_!$, which is really a morphism $f : !A \longrightarrow B$ in $\mathcal{C}$, corresponds to a *classical value* of type B, i.e., one that only depends on classical variables. The idempotence of "!" implies that morphisms $!A \longrightarrow B$ are in one-to-one correspondence with morphisms $!A \longrightarrow !B$, i.e., classical values are duplicable. For details, see [33].

3.2 Outline of the procedure for obtaining a concrete model

We construct the model in two stages. The first (more elaborate) stage constructs *abstract* models by applying certain general presheaf constructions to diagrams of functors $\mathcal{B} \to \mathcal{C} \to \mathcal{D}$. In Section 3.8 we find the precise conditions required on diagrams $\mathcal{B} \to \mathcal{C} \to \mathcal{D}$ to obtain a valid abstract model. In the second stage, we construct a *concrete* model of the quantum lambda calculus by identifying particular base categories so that the remaining conditions of Definition 3.6 are satisfied. This is the content of Sections 3.9 and 3.10.

We divide the two stages of construction into eight main steps.

1. The basic idea of the construction is to lift a sequence of functors $\mathcal{B} \xrightarrow{\Phi} \mathcal{C} \xrightarrow{\Psi} \mathcal{D}$ into a pair of adjunctions between presheaf categories

$$[\mathcal{B}^{op}, \mathbf{Set}] \underset{\Phi^*}{\overset{L}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} [\mathcal{C}^{op}, \mathbf{Set}] \underset{\Psi^*}{\overset{F_1}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} [\mathcal{D}^{op}, \mathbf{Set}].$$

   Here, Φ^* and Ψ^* are the precomposition functors, and L and F_1 are their left Kan extensions. By Remark 3.5, such a pair of adjunctions potentially yields a linear category for duplication, and thus, with additional conditions, an abstract model of quantum computation. Our goal is to identify the particular conditions on $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, Φ, and Ψ that make this construction work.
2. By Day's convolution construction (see [6]), the requirement that $[\mathcal{C}^{op}, \mathbf{Set}]$ and $[\mathcal{D}^{op}, \mathbf{Set}]$ are monoidal closed can be achieved by requiring $\mathcal{C}$ and $\mathcal{D}$ to be monoidal. The requirement that the adjunctions $L \dashv \Phi^*$ and $F_1 \dashv \Psi^*$ are monoidal is directly related to the fact that the functors Ψ and Φ are

strong monoidal. More precisely, this implies that the left Kan extension is a strong monoidal functor [10] which in turn determines the enrichment of the adjunction [14]. We also note that the category $\mathcal{B}$ must be cartesian.

3. One important complication with the model, as discussed so far, is the following. The Yoneda embedding $Y : \mathcal{D} \to [\mathcal{D}^{op}, \mathbf{Set}]$ is full and faithful, and by Day's result, also preserves the monoidal structure $\otimes$. Therefore, if one takes $\mathcal{D} = \mathbf{Q}$, all but one of the conditions of a concrete model (from Definition 3.6) are automatically satisfied. Unfortunately, however, the Yoneda embedding does not preserve coproducts, and therefore the remaining condition of Definition 3.6 fails. For this reason, we modify the construction and use a modified presheaf category with a coproduct preserving Yoneda embedding. More specifically, we choose a set Γ of cones in $\mathcal{D}$, and use the theory of continuous functors by Lambek [18] and Freyd and Kelly [11] to construct a reflective subcategory $[\mathbf{Q}^{op}, \mathbf{Set}]_\Gamma$ of $[\mathbf{Q}^{op}, \mathbf{Set}]$, such that the modified Yoneda embedding $\mathbf{Q} \longrightarrow [\mathbf{Q}^{op}, \mathbf{Set}]_\Gamma$ is coproduct preserving. Our adjunctions, and the associated Yoneda embeddings, now look like this:

$$\begin{tikzcd}
{[\mathcal{B}^{op}, \mathbf{Set}]} \arrow[rr, "L \dashv \Phi^*"] & & {[\mathcal{C}^{op}, \mathbf{Set}]} \arrow[rr, "F \dashv G"] & & {[\mathcal{D}^{op}, \mathbf{Set}]_\Gamma} \\
\mathcal{B} \arrow[u, "Y"] \arrow[rr, "\Phi"] & & \mathcal{C} \arrow[u, "Y"] \arrow[rr, "\Psi"] & & \mathcal{D} \arrow[u, "Y_\Gamma"']
\end{tikzcd}$$

The second pair of adjoint functors $F \dashv G$ is itself generated by the composition of two adjunctions:

$$\begin{tikzcd}
{[\mathcal{C}^{op}, \mathbf{Set}]} \arrow[r, shift left, "F_1"] \arrow[r, phantom, "\perp"] & {[\mathcal{D}^{op}, \mathbf{Set}]} \arrow[l, shift left, "\Psi^*"] \arrow[r, shift left, "F_2"] \arrow[r, phantom, "\perp"] & {[\mathcal{D}^{op}, \mathbf{Set}]_\Gamma} \arrow[l, shift left, "G_2"]
\end{tikzcd}$$

Here $\mathcal{D} = \mathbf{Q}$ and the pair of functors $F_2 \dashv G_2$ arises from the reflection of $[\mathbf{Q}^{op}, \mathbf{Set}]_\Gamma$ in $[\mathbf{Q}^{op}, \mathbf{Set}]$. The structure of the modified Yoneda embedding $\mathbf{Q} \longrightarrow [\mathbf{Q}^{op}, \mathbf{Set}]_\Gamma$ depends crucially on general properties of functor categories [18,11]. Full details are given in [22].

To ensure that the reflection functor remains strongly monoidal, we will use Day's reflection theorem [7], which yields necessary conditions for the reflection to be strong monoidal, by inducing a monoidal structure from the category $[\mathbf{Q}^{op}, \mathbf{Set}]$ into its subcategory $[\mathbf{Q}^{op}, \mathbf{Set}]_\Gamma$. In particular, this induces a constraint on the choice of Γ: all the cones considered in Γ must be preserved by the opposite functor of the tensor functor in $\mathcal{D}$.

4. Notice that the above adjunctions are examples of what in topos theory are called *essential geometric* morphisms, in which both functors are left adjoint to some other two functors: $L \dashv \Phi^* \dashv \Phi_*$. Therefore, this shows that the comonad "!" obtained will preserve finite coproducts.
5. The condition for the comonad "!" to be idempotent turns out to depend on the fact that the functor Φ is full and faithful; see Section 3.4.
6. In addition to requiring that "!" preserves coproducts, we also need "!" to preserve the tensor, i.e., to be strongly monoidal, as required in Definition 3.6. This property is unusual for models of intuitionistic linear logic and

restricts the possible choices for the category $\mathcal{C}$. In brief, since the left Kan extension along Φ is a strong monoidal functor, we find a concrete condition on the category $\mathcal{C}$ that is necessary to ensure that the property holds when we lift the functor Φ to the category of presheaves; see Section 3.5.

7. Our next task is to translate these categorical properties to the Kleisli category. We use the comparison Kleisli functor to pass from the framework we have already established to the Kleisli monoidal adjoint pair of functors. Also, at the same time, we shall find it convenient to characterize the functor $H : \mathcal{D} \to [\mathcal{C}^{op}, \mathbf{Set}]_T$ as a strong monoidal functor. The above steps yield an abstract model of quantum computation, parameterized by $\mathcal{B} \to \mathcal{C} \to \mathcal{D}$ and Γ.
8. Finally, in Section 3.9, we will identify specific categories $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ that yield a concrete model of quantum computation. We let $\mathcal{D} = \mathbf{Q}$, the category of superoperators. We let $\mathcal{B}$ be the category of finite sets. Alas, identifying a suitable candidate for $\mathcal{C}$ is difficult. For example, two requirements are that $\mathcal{C}$ must be affine monoidal and must satisfy the condition of equation (1) in Section 3.5 below. We construct such a $\mathcal{C} = \mathbf{Q}''$ related to the category $\mathbf{Q}$ of superoperators with the help of some universal constructions.

The above base category $\mathbf{Q}''$ plays a central role in our construction. While the higher-order structural properties of the quantum lambda calculus hold at the pure functor category level, the interpretation of concrete quantum operations takes place mostly at this base level.

Let us now discuss some details of the construction.

3.3 Categorical models of linear logic on presheaf categories

The first categorical models of linear logic were given by Seely [27]. The survey by Melliès is an excellent introduction [23]. Current state-of-the-art definitions are Bierman's definition of a linear category [4], simplified yet more by Benton's definition of a linear-non-linear category ([2], cf. Remark 3.5 above). Benton proved the equivalence of these two notions [2,23].

Definition 3.8 (Benton [2]). A *linear-non-linear category* consists of:

(1) a symmetric monoidal closed category $(\mathcal{C}, \otimes, I, \multimap)$,
(2) a category $(\mathcal{B}, \times, 1)$ with finite products,
(3) a symmetric monoidal adjunction: $\begin{tikzcd}(\mathcal{B}, \times, 1) \arrow[r, shift left, "{(F,m)}"] \arrow[r, phantom, "\perp"] & (\mathcal{C}, \otimes, I) \arrow[l, shift left, "{(G,n)}"]\end{tikzcd}$.

Remark 3.9. We use Kelly's characterization of monoidal adjunctions to simplify condition (3) in Definition 3.8 above to:

(3') an adjunction: $\begin{tikzcd}(\mathcal{B}, \times, 1) \arrow[r, shift left, "F"] \arrow[r, phantom, "\perp"] & (\mathcal{C}, \otimes, I) \arrow[l, shift left, "G"]\end{tikzcd}$, and there exist isomorphisms $m_{A,B} : FA \otimes FB \to F(A \times B)$ and $m_I : I \to F(1)$, making $(F, m_{A,B}, m_I) : (\mathcal{B}, \times, 1) \to (\mathcal{C}, \otimes, I)$ a strong symmetric monoidal functor.

Details of this characterization can found in [23].

We can characterize Benton's linear-non-linear models (Definition 3.8) on presheaf categories using Day's monoidal structure [6]. This is an application of monoidal enrichment of the Kan extension, see [10]. We use the following:

Proposition 3.10 (Day-Street[10]). *Suppose we have a strong monoidal functor* $\Phi : (\mathcal{A}, \otimes, 1) \to (\mathcal{B}, \otimes, I)$ *between two monoidal categories, i.e., we have natural isomorphisms* $\Phi(a) \otimes \Phi(b) \cong \Phi(a \otimes b)$ *and* $I \cong \Phi(I)$*. Consider the left Kan extension along* Φ *in the functor category* $[\mathcal{B}^{op}, \mathbf{Set}]$*, where the copower is the cartesian product on sets:* $Lan_\Phi(F) = \int^a \mathcal{B}(-, \Phi(a)) \times F(a)$*. Then* Lan_Φ *is strong monoidal.*

Remark 3.11. If $\mathcal{A}$ is cartesian then the Day tensor (convolution) $[\mathcal{A}^{op}, \mathbf{Set}] \times [\mathcal{A}^{op}, \mathbf{Set}] \xrightarrow{\otimes_D} [\mathcal{A}^{op}, \mathbf{Set}]$ is a pointwise product of functors. Also if the unit of a monoidal category $\mathcal{C}$ is a terminal object then the unit of $\otimes_D$ is also terminal.

3.4 Idempotent comonad in the functor category

A comonad $(!, \epsilon, \delta)$ is said to be *idempotent* if $\delta : ! \Rightarrow !!$ is an isomorphism. Let $(!, \epsilon, \delta)$ be the comonad generated by an adjunction $(\mathfrak{B}, \times, 1) \underset{G}{\overset{F}{\rightleftarrows}} (\mathfrak{C}, \otimes, I)$, with $F \dashv G$. Then $\delta = F\eta_G$ with $\eta : I \to GF$. Thus if η is an isomorphism then δ is also an isomorphism. In the context of our model construction, how can we guarantee that η is an isomorphism? Consider the unit $\eta_B : B \Rightarrow \Phi^*(Lan_\Phi(B))$ of the adjunction generated by the Kan extension:

$$[\mathcal{B}^{op}, \mathbf{Set}] \underset{\Phi^*}{\overset{Lan_\Phi}{\rightleftarrows}} [\mathcal{C}^{op}, \mathbf{Set}].$$

Proposition 3.12 ([5]). *If* Φ *is full and faithful then* $\eta_B : B \Rightarrow \Phi^*(Lan_\Phi(B))$ *is an isomorphism.*

3.5 A strong comonad

In this section we study conditions that force the idempotent comonad above to be a strong monoidal functor. This property is part of the model we are building and is one of the main differences with previous models of intuitionistic linear logic [23].

To achieve this, consider a fully faithful functor $\Phi : \mathcal{B} \longrightarrow \mathcal{C}$, as in Section 3.2. Let $[\mathcal{C}^{op}, \mathbf{Set}] \xrightarrow{\Phi^*} [\mathcal{B}^{op}, \mathbf{Set}]$ be the precomposition functor, i.e., the right adjoint of the left Kan extension.

Lemma 3.13 ([9]). *If there exists a natural isomorphism*

$$\mathcal{C}(\Phi(b), c) \times \mathcal{C}(\Phi(b), c') \cong \mathcal{C}(\Phi(b), c \otimes c'), \tag{1}$$

where $b \in \mathcal{B}$ *and* $c, c' \in \mathcal{C}$ *and* Φ *is a fully faithful functor satisfying* $\Phi(1) = I$*, then* Φ^* *is a strong monoidal functor.*

In Section 3.9 we shall build a category satisfying this specific requirement among others. More precisely, from our viewpoint, this will depend on the construction of a certain category that we will name $\mathbf{Q}''$, which is a modification of the category $\mathbf{Q}$ of superoperators. Also, we will consider a fully faithful strong monoidal functor $\Phi : (\mathbf{FinSet}, \times, 1) \to (\mathcal{C}, \otimes_{\mathcal{C}}, I)$ that generates the first adjunction in Section 3.2, where $\mathcal{C} = \mathbf{Q}''$.

3.6 The functor $H : \mathcal{D} \to \hat{\mathcal{C}}_T$

Let $\mathcal{C}$ and $\mathcal{D}$ be categories. Consider an adjoint pair of functors $[\mathcal{C}^{op}, \mathbf{Set}] \underset{G}{\overset{F}{\underset{\perp}{\rightleftarrows}}} [\mathcal{D}^{op}, \mathbf{Set}]_\Gamma$, as mentioned in Section 3.2, item 3. Let $T = G \circ F$ and $\hat{\mathcal{C}} = [\mathcal{C}^{op}, \mathbf{Set}]$. In this section we consider the construction of a coproduct preserving and tensor preserving functor $H : \mathcal{D} \to \hat{\mathcal{C}}_T$ with properties similar to the Yoneda embedding, for a general category $\mathcal{D}$.

Let $F_1 \dashv G_1$ and $F_2 \dashv G_2$ be two monoidal adjoint pairs with associated natural transformations (F_1, m_1), (G_1, n_1) and (F_2, m_2), (G_2, n_2). We shall use the following notation: $F = F_2 \circ F_1$, $G = G_1 \circ G_2$, and $T = G \circ F$. We now describe a typical situation of this kind generated by a functor $\Psi : \mathcal{C} \to \mathcal{D}$.

Let us consider $F_1 = Lan_\Psi$ and $G_1 = \Psi^*$. With some co-completeness condition assumed, we can express $F_1(A) = \int^c \mathcal{D}(-, \Psi(c)) \otimes A(c)$ and $G_1 = \Psi^*$.

On the other hand we consider $F_2 = Lan_Y(Y_\Gamma) : [\mathcal{D}^{op}, \mathbf{Set}] \to [\mathcal{D}^{op}, \mathbf{Set}]_\Gamma$, where $Y_\Gamma : \mathcal{D} \to [\mathcal{D}^{op}, \mathbf{Set}]_\Gamma$ is the co-restriction of the Yoneda functor given by $Y_\Gamma(d) = \mathcal{D}(-, d)$. Thus we have $F_2(D) = \int^d D(d) \otimes Y_\Gamma(d)$. Assuming that $[\mathcal{D}^{op}, \mathbf{Set}]_\Gamma$ is co-complete and contains the representable presheaves, then the right adjoint G_2 is isomorphic to the inclusion functor.

Definition of H.

We want to study the following situation:

$$\begin{tikzcd}
 & \hat{\mathcal{C}} \arrow[r, shift left, "F_1"] \arrow[r, phantom, "\perp"] \arrow[d, shift left, "F_T"] \arrow[d, phantom, "\vdash"] & \hat{\mathcal{D}} \arrow[l, shift left, "G_1"] \arrow[r, shift left, "F_2"] \arrow[r, phantom, "\perp"] & \hat{\mathcal{D}}_\Gamma \arrow[l, shift left, "G_2"] \\
\mathcal{C} \arrow[ur, bend left, "Y"] \arrow[dr, bend right, "\Psi"'] & \hat{\mathcal{C}}_T \arrow[u, shift left, "G_T"] \arrow[urr, "C"] & & \\
 & \mathcal{D} \arrow[u, dashed, "H"] \arrow[uurr, bend right, "Y_\Gamma"'] & &
\end{tikzcd}$$

The goal is to determine a full and faithful functor, denoted H in this diagram, that preserves tensor and coproduct.

First, notice that the perimeter of this diagram commutes on objects: $F_1(\mathcal{C}(-,c)) = \int^{c'} \mathcal{D}(-,\Psi(c')) \otimes \mathcal{C}(c',c) = \mathcal{D}(-,\Psi(c))$ and when we evaluate again, using F_2, we obtain:

$$F_2(\mathcal{D}(-,\Psi(c))) = \int^{d'} \mathcal{D}(d',\Psi(c)) \otimes Y_\Gamma(d') = Y_\Gamma(\Psi(c)) = \mathcal{D}(-,\Psi(c)).$$

Summing up, we have that $F(\mathcal{C}(-,c)) = \mathcal{D}(-,\Psi(c))$ up to isomorphism.

Suppose now that Ψ is essentially onto on objects and we have that:

$$\mathcal{D}(-,d) \cong \mathcal{D}(-,\Psi(c))$$

for some $c \in \mathcal{C}$, i.e., we can make a choice, for every $d \in |\mathcal{D}|$, of some $c \in |\mathcal{C}|$ such that $\Psi(c) \cong d$. Let us call this choice a "choice of preimages". We can therefore define a map $H : |\mathcal{D}| \to |\hat{\mathcal{C}}_T|$ by $H(d) = \mathcal{C}(-,c)$ on objects.

Then we can define a functor $H : \mathcal{D} \to \hat{\mathcal{C}}_T$ in the following way: let $d \xrightarrow{f} d'$ be an arrow in the category $\mathcal{D}$. We apply Y_Γ obtaining $\mathcal{D}(-,d) \xrightarrow{Y_\Gamma(f)} \mathcal{D}(-,d')$. This arrow is equal to $\mathcal{D}(-,\Psi(c)) \xrightarrow{Y_\Gamma(f)} \mathcal{D}(-,\Psi(c'))$ for some $c, c' \in \mathcal{C}$, and for the reason stipulated above is equal to $F(\mathcal{C}(-,c)) \xrightarrow{Y_\Gamma(f)} F(\mathcal{C}(-,c'))$. Now we use the fact that the comparison functor $C : \hat{\mathcal{C}}_T \to \hat{\mathcal{D}}_\Gamma$, i.e.,

$$C : \hat{\mathcal{C}}_T(\mathcal{C}(-,c), \mathcal{C}(-,c')) \to \hat{\mathcal{D}}_\Gamma(F(\mathcal{C}(-,c)), F(\mathcal{C}(-,c'))),$$

is full and faithful. Thus there is a unique $\gamma : \mathcal{C}(-,c) \to \mathcal{C}(-,c')$ such that $C(\gamma) = Y_\Gamma(f)$. Then we can define $H(f) = \gamma$ on morphisms and (as mentioned above) $H(d) = \mathcal{C}(-,c)$ on objects, where c is given by our choice of preimages.

$C : \hat{\mathcal{C}}_T \to \hat{\mathcal{D}}_\Gamma$ is a strong monoidal functor

We define $C(A) \otimes_{\hat{\mathcal{D}}_\Gamma} C(B) \xrightarrow{u_{AB}} C(A \otimes_{\mathcal{C}_T} B)$ as $F(A) \otimes_{\hat{\mathcal{D}}_\Gamma} F(B) \xrightarrow{m_{AB}} F(A \otimes B)$. It is easy to check naturality. Also define $I \xrightarrow{u_I = m_I} C(I) = F(I)$. Since m_{AB} and m_I are invertible in $\hat{\mathcal{D}}_\Gamma$, we have that u_{AB} and u_I are invertible. This implies that (C, m) is a strong functor. Also, coherence of isomorphisms is easily checked.

$H : \mathcal{D} \to \hat{\mathcal{C}}_T$ is a strong monoidal functor

We want to define a natural transformation $H(A) \otimes_{\hat{\mathcal{C}}_T} H(B) \xrightarrow{\psi_{A,B}} H(A \otimes_{\mathcal{D}} B)$ that makes H into a strong monoidal functor.

We begin by recalling that (C, u) and (Y_Γ, y) are strong monoidal, i.e., u and y are isomorphisms. Since C is fully faithful, this allows us to define $\psi_{A,B}$ as the unique map making the following diagram commute:

$$\begin{tikzcd}
Y_\Gamma(A) \otimes Y_\Gamma(B) \arrow[rr, "y_{A,B}"] \arrow[dr, "u_{HA,HB}"'] & & Y_\Gamma(A \otimes B) = C \circ H(A \otimes B). \\
& C(H(A) \otimes H(B)) \arrow[ur, "C(\psi_{A,B})"'] &
\end{tikzcd}$$

We define ψ_I similarly. Furthermore, ψ is a natural transformation and satisfies all the axioms of a monoidal structure. We refer to [22] for the details.

H preserves coproducts

Here we focus on the specific problem of the preservation of finite coproducts by the functor H defined in Section 3.6. First, note that the category $[\mathcal{C}^{op}, \mathbf{Set}]$ has finite coproducts, computed pointwise. Moreover, the Kleisli category $\hat{\mathcal{C}}_T$ inherits the coproduct structure from $\hat{\mathcal{C}}$ since:

Proposition 3.14. *If $\mathcal{C}$ has finite coproducts, then so does $\mathcal{C}_T$.*

Therefore, $[\mathcal{C}^{op}, \mathbf{Set}]_T$ has finite coproducts. Recall that the comparison functor $C : [\mathcal{C}^{op}, \mathbf{Set}]_T \to [\mathcal{D}^{op}, \mathbf{Set}]_\Gamma$ is fully faithful. Also, by a well known property of representable functors (see [18]), we have that $H : \mathcal{D} \to [\mathcal{C}^{op}, \mathbf{Set}]_T$ preserves coproducts iff $[\mathcal{C}^{op}, \mathbf{Set}]_T(H-, A) : \mathcal{D}^{op} \to \mathbf{Set}$ preserves products for every $A \in [\mathcal{C}^{op}, \mathbf{Set}]_T$. Using these two facts we prove the following:

Proposition 3.15. *If the class Γ contains all the finite product cones, then H preserves finite coproducts.*

We refer to [22] for the details. From this, we impose that Γ contains all the finite product cones. This is another requirement to obtain a model.

3.7 $F_T \dashv G_T$ is a monoidal adjunction

We recall how a monoidal adjoint pair $(F, m) \dashv (G, n)$ induces a monoidal structure for the adjunction $F_T \dashv G_T$ associated with the Kleisli construction.

Lemma 3.16. *Let $F \dashv G$ be a monoidal adjunction, let $T = GF$, and consider the Kleisli adjunction* $\begin{tikzcd} \mathcal{C} \arrow[r, shift left, "F_T"] \arrow[r, phantom, "\perp"] & \mathcal{C}_T \arrow[l, shift left, "G_T"] \end{tikzcd}$ *generated by this adjunction. Then $\mathcal{C}_T$ is a monoidal category and $F_T \dashv G_T$ is a monoidal adjunction.*

Proof. Since $F \dashv G$ is a monoidal adjunction, it follows that $T = GF$ is a monoidal monad. The result then follows by general properties of monoidal monads and monoidal adjunctions.

3.8 Abstract model of the quantum lambda calculus

Summing up the parts from previous sections, we have the following theorem.

Theorem 3.17. *Given categories $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$, and functors $\mathcal{B} \xrightarrow{\Phi} \mathcal{C} \xrightarrow{\Psi} \mathcal{D}$, satisfying*

- *$\mathcal{B}$ has finite products, $\mathcal{C}$ and $\mathcal{D}$ are symmetric monoidal,*
- *$\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ have coproducts, which are distributive w.r.t. tensor,*
- *$\mathcal{C}$ is affine,*
- *Φ and Ψ are strong monoidal,*

– *Φ and Ψ preserve coproducts,*
– *Φ is full and faithful,*
– *Ψ is essentially surjective on objects,*
– *for every $b \in \mathcal{B}$, $c, c' \in \mathcal{C}$ we have $\mathcal{C}(\Phi(b), c) \times \mathcal{C}(\Phi(b), c') \cong \mathcal{C}(\Phi(b), c \otimes c')$.*

Let Γ be any class of cones preserved by the opposite tensor functor, including all the finite product cones. Let Lan_Φ, Φ^*, F and G be defined as in Section 3.2 and subsequent sections. Then

$$\begin{tikzcd} {[\mathcal{B}^{op}, \mathbf{Set}]} \arrow[r, shift left, "Lan_\Phi"] \arrow[r, phantom, "\perp"] & {[\mathcal{C}^{op}, \mathbf{Set}]} \arrow[l, shift left, "\Phi^*"] \arrow[r, shift left, "F"] \arrow[r, phantom, "\perp"] & {[\mathcal{D}^{op}, \mathbf{Set}]_\Gamma} \arrow[l, shift left, "G"] \end{tikzcd}$$

forms an abstract model of the quantum lambda calculus.

Proof. Relevant propositions from previous sections.

3.9 Towards a concrete model: constructing $\mathbf{FinSet} \xrightarrow{\Phi} \mathbf{Q}'' \xrightarrow{\Psi} \mathbf{Q}$

The category $\mathbf{Q}$ of superoperators was defined in Section 2. Here, we discuss a category $\mathbf{Q}''$ related to $\mathbf{Q}$, together with functors $\mathbf{FinSet} \xrightarrow{\Phi} \mathbf{Q}'' \xrightarrow{\Psi} \mathbf{Q}$. The goal is to choose $\mathbf{Q}''$ and the functors Φ and Ψ carefully so as to satisfy the requirements of Theorem 3.17.

Recall the definition of the free affine monoidal category $(\mathcal{F}wm(\mathcal{K}), \otimes, I)$:

– Objects are finite sequences of objects of $\mathcal{K}$: $\{V_i\}_{i\in[n]} = \{V_1, \dots, V_n\}$.
– Maps $(\varphi, \{f_i\}_{i\in[m]}) : \{V_i\}_{i\in[n]} \longrightarrow \{W_i\}_{i\in[m]}$ are determined by:
  (i) an injective function $\varphi : [m] \to [n]$,
  (ii) a family of morphisms $f_i : V_{\varphi(i)} \to W_i$ in the category $\mathcal{K}$.
– Tensor $\otimes$ is given by concatenation, with unit I given by the empty sequence.

Proposition 3.18. *There is a canonical inclusion $Inc : \mathcal{K} \to \mathcal{F}wm(\mathcal{K})$ satisfying: for any symmetric monoidal category $\mathcal{A}$ whose tensor unit is terminal and any functor $F : \mathcal{K} \to \mathcal{A}$, there is a unique strong monoidal functor $G : \mathcal{F}wm(\mathcal{K}) \to \mathcal{A}$, up to isomorphism, such that $G \circ Inc = F$.*

We apply this universal construction to the situation where $\mathcal{K}$ is a discrete category. For later convenience, we let $\mathcal{K}$ be the discrete category with finite dimensional Hilbert spaces as objects. Then $\mathcal{F}wm(\mathcal{K})$ has sequences of Hilbert spaces as objects and dualized, compatible, injective functions as arrows.

Now consider the identity-on-objects inclusion functor $F : \mathcal{K} \to \mathbf{Q}'_s$, where $\mathbf{Q}'_s$ is the category of simple trace-preserving superoperators defined in Section 2. Since $\mathbf{Q}'_s$ is affine, by Proposition 3.18 there exists a unique (up to natural isomorphism) strong monoidal functor $\hat{F}$ such that:

$$\begin{tikzcd} \mathcal{K} \arrow[r, "F"] \arrow[d, "Inc"'] & \mathbf{Q}'_s \\ \mathcal{F}wm(\mathcal{K}) \arrow[ur, "\hat{F}"'] & \end{tikzcd}$$

Remark 3.19. This reveals the purpose of using equality instead of $\leqslant$ in the definition of a trace-preserving superoperator (Definition 2.9). When the codomain is the unit, there is only one map $f(\rho) = \mathrm{tr}(\rho)$, and therefore $\mathbf{Q}'_s$ is affine.

Now we remind the reader about the general properties of the free finite coproduct completion $\mathcal{C}^+$ of a category $\mathcal{C}$. The category $\mathcal{C}^+$ has as its objects finite families of objects of $\mathcal{C}$, say $V = \{V_a\}_{a\in A}$, with A a finite set. A morphism from $V = \{V_a\}_{a\in A}$ to $W = \{W_b\}_{b\in B}$ consists of the following:

– a function $\varphi : A \to B$,
– a family $f = \{f_a\}_{a\in A}$ of morphisms of $\mathcal{C}$, where $f_a : V_a \to W_{\varphi(a)}$.

The coproduct in $\mathcal{C}^+$ is just concatenation of families of objects of $\mathcal{C}$.

Proposition 3.20. *Given any category $\mathcal{A}$ with finite coproducts and any functor $F : \mathcal{C} \to \mathcal{A}$, there is a unique finite coproduct preserving functor $G : \mathcal{C}^+ \to \mathcal{A}$, up to natural isomorphism, such that $G \circ Inc = F$.*

$$\begin{tikzcd} \mathcal{C} \arrow[r, "F"] \arrow[d, "Inc"'] & \mathcal{A} \\ \mathcal{C}^+ \arrow[ur, "G"'] & \end{tikzcd}$$

If $\mathcal{C}$ is a symmetric monoidal category then $\mathcal{C}^+$ is also symmetric monoidal. In addition, if we assume that the categories $\mathcal{C}$ and $\mathcal{A}$ are symmetric monoidal, then Inc is a symmetric monoidal functor. If F is a symmetric monoidal functor and tensor distributes over coproducts in $\mathcal{A}$, then G is a symmetric monoidal functor. Moreover, if F is strong monoidal then so is G.

In the sequel we want to apply Proposition 3.20 to a concrete category, but first:

Remark 3.21. By definition, $\mathbf{Q}_s$ is a full subcategory of $\mathbf{Q}$, and the inclusion functor $In : \mathbf{Q}_s \to \mathbf{Q}$ is strong monoidal. Also, since every trace preserving superoperator is trace non-increasing, $\mathbf{Q}'_s$ is a subcategory of $\mathbf{Q}_s$, and the inclusion functor $E : \mathbf{Q}'_s \to \mathbf{Q}_s$ is strong monoidal as well.

We apply the machinery of Proposition 3.20 to the composite functor

$$\mathcal{F}wm(\mathcal{K}) \xrightarrow{\hat{F}} \mathbf{Q}'_s \xrightarrow{E} \mathbf{Q}_s \xrightarrow{In} \mathbf{Q},$$

where In and E are as defined in Remark 3.21.

Definition 3.22. Let $\mathbf{Q}'' = (\mathcal{F}wm(\mathcal{K}))^+$ and let Ψ be the unique finite coproduct preserving functor making the following diagram commute:

$$\begin{tikzcd} \mathcal{F}wm(\mathcal{K}) \arrow[r, "\hat{F}"] \arrow[d, "Inc"'] & \mathbf{Q}'_s \arrow[r, "E"] & \mathbf{Q}_s \arrow[r, "In"] & \mathbf{Q}. \\ (\mathcal{F}wm(\mathcal{K}))^+ \arrow[urrr, "\Psi"'] & & & \end{tikzcd} \tag{2}$$

Note that such a functor exists by Proposition 3.20, and it is strong monoidal.

Remark 3.23. Since $\Psi\{\{V_i^a\}_{i\in[n_a]}\}_{a\in A} = \coprod_{a\in A}\{(V_1^a\otimes\ldots\otimes V_{n_a}^a)_*\}_{*\in 1}$, the functor Ψ is essentially onto objects. Specifically, given any object $\{V_a\}_{a\in A}\in|\mathbf{Q}|$, we can choose a preimage (up to isomorphism) as follows:

$$\Psi\{\{V_i^a\}_{i\in[1]}\}_{a\in A} = \coprod_{a\in A}\{(V_1^a)_*\}_{*\in 1} \cong \{V_a\}_{a\in A}. \tag{3}$$

Lemma 3.24. *Let $\mathcal{C}$ be an affine category. Then there is a fully faithful strong monoidal functor $\Phi : (\mathbf{FinSet}, \times, 1) \to (\mathcal{C}^+, \otimes_{\mathcal{C}^+}, I)$ that preserves coproducts.*

Definition 3.25. Recall that $\mathcal{F}wm(\mathcal{K})$ is an affine category and $\mathbf{Q}'' = \mathcal{F}wm(\mathcal{K})^+$. Let $\Phi : \mathbf{FinSet} \to \mathbf{Q}''$ be the functor defined by Lemma 3.24.

Remark 3.26. With the above choice of $\Phi : \mathbf{FinSet} \to \mathbf{Q}''$, equation (1) in Lemma 3.13 is just the characterization of cartesian products in **FinSet** using representable functors.

Theorem 3.27. *The choice $\mathcal{B} = \mathbf{FinSet}$, $\mathcal{C} = \mathbf{Q}''$, $\mathcal{D} = \mathbf{Q}$, with the functors Φ as in Definition 3.25 and Ψ as in Definition 3.22, and with Γ the class of all finite product cones in $\mathcal{D}^{op}$, satisfies all the properties required by Theorem 3.17.*

3.10 A concrete model

Theorem 3.28. *Let $\mathbf{Q}$, $\mathbf{Q}''$, Φ, Ψ, and Γ be defined as in Sections 2 and 3.9. Then*

$$\begin{tikzcd}[\mathbf{FinSet}^{op}, \mathbf{Set}] \arrow[r, shift left, "Lan_\Phi"] \arrow[r, phantom, "\perp"] & {[(\mathbf{Q}'')^{op}, \mathbf{Set}]} \arrow[l, shift left, "\Phi^*"] \arrow[r, shift left, "F"] \arrow[r, phantom, "\perp"] & {[\mathbf{Q}^{op}, \mathbf{Set}]_\Gamma} \arrow[l, shift left, "G"]\end{tikzcd}$$

forms a concrete model of the quantum lambda calculus.

Proof. This follows from Theorems 3.17 and 3.27.

4 Conclusions and future work

We have constructed mathematical (semantic) models of higher-order quantum computation, specifically for the quantum lambda calculus of Selinger and Valiron. The central idea of our model construction was to apply the presheaf construction to a sequence of three categories and two functors, and to find a set of sufficient conditions for the resulting structure to be a valid model. The construction depends crucially on properties of presheaf categories, using Day's convolution theory and the Kelly-Freyd notion of continuity of functors.

We then identified specific base categories and functors that satisfy these abstract conditions, based on the category of superoperators. Thus, our choice of base categories ensures that the resulting model has the "correct" morphisms at base types, whereas the presheaf construction ensures that it has the "correct" structure at higher-order types.

Our work has concentrated solely on the existence of such a model. One question that we have not yet addressed is specific properties of the interpretation of quantum lambda calculus in this model. It would be interesting, in future work, to analyze whether this particular interpretation yields new insights into the nature of higher-order quantum computation, or to use this model to compute properties of programs.

Acknowledgements. This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Program for the Development of Basic Sciences, Uruguay (PEDECIBA).

References

1. S. Abramsky, B. Coecke. A categorical semantics of quantum protocols. In *Proc. 19th Annual IEEE Symp. on Logic in Computer Science (LICS 2004)*, IEEE Computer Soc. Press, 2004, 415–425.
2. N. Benton. A mixed linear and non-linear logic: proofs, terms and models (extended abstract). In L. Pacholski and J. Tiuryn, editors. *Computer Science Logic, CSL' 94 Selected Papers*, *LNCS*, Springer, v. 933 (1994), 121–135.
3. G. Bierman. *On intuitionistic linear logic.* Ph.D. thesis, Computer Science department, Cambridge University, 1993.
4. G. Bierman. What is a categorical model of intuitionistic linear logic?, in: *Proc. Typed Lambda Calculi and Applications, TLCA '95*, *LNCS*, Springer, v. 902 (1995).
5. F. Borceux. *Handbook of Categorical Algebra 1.* Cambridge University Press, 1994.
6. B. Day. On closed categories of functors. *Lecture Notes in Math.* 137 (1970), Springer, 1–38.
7. B. Day. A reflection theorem for closed categories. *J. Pure Appl. Algebra* 2 (1972), 1–11.
8. B. Day. Note on monoidal localisation. *Bull. Austral. Math. Soc.* 8 (1973), 1–16.
9. B. Day. Monoidal functor categories and graphic Fourier transforms (2006). ArXiv:math/0612496.
10. B. Day, R. Street. Kan extensions along promonoidal functors. *Theory and Applications of Categories* 1 (4) (1995), 72–78.
11. P. Freyd, G. M. Kelly. Categories of continuous functors I, *J. Pure and Appl. Algebra* 2 (1972), 169–191.
12. J. Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1), 1987, 1–101.
13. G. B. Im, G. M. Kelly. A universal property of the convolution monoidal structure. *J. Pure and Appl. Algebra* 43 (1986), 75–88.
14. G. M. Kelly. Doctrinal adjunction, *Lecture Notes in Math.* 420 (1974), Springer, 257–280.
15. G. M. Kelly. *Basic Concepts of Enriched Category Theory*, LMS Lecture Notes 64, Cambridge University Press, 1982.
16. A. Kock. Monads on symmetric monoidal closed categories. *Arch. Math.* 21 (1970), 1–10.
17. A. Kock. Strong functors and monoidal monads. *Archiv der Mathematik* 23, 1972.
18. J. Lambek. *Completions of Categories*, Lecture Notes in Math. 24, Springer, 1966.
19. J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic.* Cambridge University Press, 1986.

20. M. L. Laplaza. Coherence for distributivity. Springer *Lecture Notes in Math.* 281, 1972, 29–65.
21. S. Mac Lane. *Categories for the Working Mathematician*, 2nd ed., Springer, 1998.
22. O. Malherbe. *Categorical models of computation: partially traced categories and presheaf models of quantum computation*. Ph.D. thesis, University of Ottawa, 2010. Available from arXiv:1301.5087.
23. P.-A. Melliès. Categorical models of linear logic revisited. Preprint, 2002. Appeared as: Categorical semantics of linear logic, in *Interactive models of computation and program behaviour*, P.-L. Curien, H. Herbelin, J.-L. Krivine, P.-A. Melliès, eds., Panoramas et Synthèses 27, Société Mathématique de France, 2009.
24. E. Moggi. Computational lambda-calculus and monads. Technical Report ECS-LFCS-88-66, Lab. for Foundations of Computer Science, U. Edinburgh, 1988.
25. E. Moggi. Notions of computation and monads. *Information and Computation* 93(1), 1991, 55–92.
26. A. Nielsen, I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
27. R. Seely. Linear logic, *-autonomous categories and cofree coalgebras. In J. W. Gray and A. Scedrov, eds., *Categories in Computer Science and Logic*, Volume 92 of *Contemporary Mathematics*, Amer. Math. Soc. 1989, 371–382.
28. P. Selinger. Towards a quantum programming language, *Math. Structures in Comp. Sci.* 14(4), 2004, 527–586.
29. P. Selinger. Dagger compact closed categories and completely positive maps. In P. Selinger, ed., *Proceedings of the Third International Workshop on Quantum Programming Languages*, (QPL 2005), Chicago. ENTCS 170 (2007), 139–163.
30. P. Selinger, B. Valiron. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science* 16 (2006), 527–552.
31. P. Selinger, B. Valiron. On a fully abstract model for a quantum functional language. *Proceedings of the Fourth International Workshop on Quantum Programming Languages*, Springer ENTCS 210 (2008), 123–137.
32. P. Selinger, B. Valiron. A linear-non-linear model for a computational call-by-value lambda calculus. In *Proc. of the Eleventh International Conference on Foundations of Software Science and Computation Structures* (FOSSACS 2008), Budapest, Springer *LNCS* 4962 (2008), 81–96.
33. P. Selinger, B. Valiron. Quantum lambda calculus. In: *Semantic Techniques in Quantum Computation*, Cambridge University Press, S. Gay and I. Mackie, eds., 2009, 135–172.
34. P. W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, *Proc. 35nd Annual Symposium on Foundations of Computer Science*, S. Goldwasser, ed., IEEE Computer Society Press (1994), 124–134.
35. B. Valiron. *Semantics for a higher order functional programming language for quantum computation*. Ph.D. thesis, University of Ottawa, 2008.