# Applying Quantitative Semantics to Higher-Order Quantum Computing*


Michele Pagani
Université Paris 13, Sorbonne Paris Cité
Villetaneuse, France
michele.pagani@lipn.univ-paris13.fr

Peter Selinger
Dalhousie University
Halifax, Canada
selinger@mathstat.dal.ca

Benoît Valiron
CIS Dept, University of Pennsylvania
Philadelphia, U.S.A.
benoit.valiron@monoidal.net





## Abstract

Finding a denotational semantics for higher order quantum computation is a long-standing problem in the semantics of quantum programming languages. Most past approaches to this problem fell short in one way or another, either limiting the language to an unusably small finitary fragment, or giving up important features of quantum physics such as entanglement. In this paper, we propose a denotational semantics for a quantum lambda calculus with recursion and an infinite data type, using constructions from quantitative semantics of linear logic.


## 1 Introduction

Type theory and denotational semantics have been successfully used to model, design, and reason about programming languages for almost half a century. The application of such methods to quantum computing is much more recent, going back only about 10 years [17].

An important problem in the semantics of quantum computing is how to combine quantum computing with higher-order functions, or in other words, how to design a functional quantum programming language. A syntactic answer to this question was arguably given with the design of the quantum lambda calculus [22, 19]. The quantum lambda calculus has a well-defined syntax and operational semantics, with a strong type system and a practical type inference algorithm. However, the question of how to give a *denotational* semantics to the quantum lambda calculus turned out to be difficult, and has remained open for many years [18, 21]. One reason that designing such a semantics is difficult is that quantum computation is inherently defined on *finite dimensional* Hilbert spaces, whereas the semantics of higher-order functional programming languages, including such features as infinite data types and recursion, is inherently infinitary.

In recent years, a number of solutions have been proposed to the problem of finding a denotational semantics of higher-order quantum computation, with varying degrees of success. The first approach [20] was to restrict the language to strict linearity, meaning that each function had to use each argument exactly once, in the spirit of linear logic. In this way, all infinitary concepts (such as infinite types and recursion) were eliminated from the language. Not surprisingly, the resulting finitary language permitted a fully abstract semantics in terms of finite dimensional spaces; this was hardly an acceptable solution to the general problem. The second approach [13] was to construct a semantics of higher-order quantum computation by methods from category theory; specifically, by applying a presheaf construction to a model of first-order quantum computation. This indeed succeeds in yielding a model of the full quantum lambda calculus, albeit without recursion. The main drawbacks of the presheaf model are the absence of recursion, and the fact that such models are relatively difficult to reason about. The third approach [6] was based on the Geometry of Interaction. Starting from a traced monoidal category of basic quantum operations, Hasuo and Hoshino applied a sequence of categorical constructions, which eventually yielded a model of higher-order quantum computation. The problem with this approach is that the tensor product constructed from the geometry-of-interaction construction does not coincide with the tensor product of the underlying physical data types. Therefore, the model drops the possibility of entangled states, and thereby fails to model one of the defining features of quantum computation.

**Our contribution.** In this paper, we give a novel denotational semantics of higher-order quantum computation, based on methods from *quantitative semantics*. Quantitative semantics refers to a family of semantics of linear logic that interpret proofs as linear mappings between vector spaces (or more generally, modules), and standard lambda terms as power series. The original idea comes from Girard's normal functor semantics [4]. More recently, quantitative semantics has been used to give a solid, denotational semantics for various algebraic extensions of lambda calculus, such as probabilistic and differential lambda calculi (e.g. [1], [2]).

One feature of our model is that it can represent *infinite dimensional* structures, and is expressive enough to describe recursive types, such as lists of qubits, and to model recursion. This is achieved by providing an exponential structure *à la* linear logic. Unlike the Hasuo-Hoshino model, our model permits general entanglement. We interpret (a minor variant of) the quantum lambda calculus in this model. Our main result is the adequacy of the model with respect to the operational semantics.

The model is the juxtaposition of a simple, finite-dimensional model of quantum computation together with a canonical completion yielding the structures of linear logic. Our model demonstrates that the quantum and the classical "universes" work well

*Partially founded by French ANR project COQUAS (number 12 JS02 006 01) and CNRS chair "Logique linéaire et calcul".

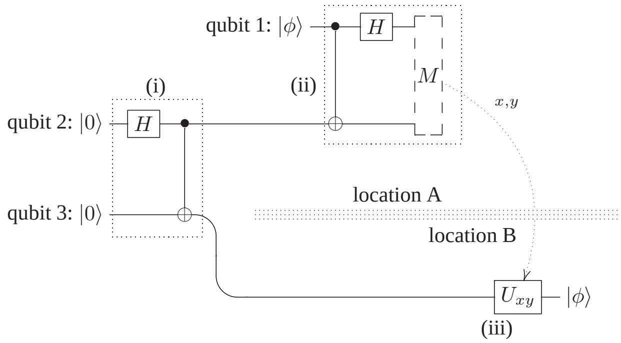


Figure 1: The quantum teleportation protocol.

together, but also – surprisingly – that they do not mix too much, even at higher order types.

**Outline.** In Section 2, we briefly review some background. Section 3 presents the version of the quantum lambda calculus that we use in this paper, including its operational semantics. Section 4 presents the denotational semantics of the quantum lambda calculus, and Section 5 proves the adequacy theorem. Section 6 concludes with some properties of the representable elements.

# 2 Background

## 2.1 Quantum computation in a nutshell

Quantum computation is a computational paradigm based on the laws of quantum physics. We briefly recall some basic notions; please see [16] for a more complete treatment. The basic unit of information in quantum computation is a *quantum bit* or *qubit*, whose state is given by a normalized vector in the two-dimensional Hilbert space $\mathbb{C}^2$. It is customary to write the canonical basis of $\mathbb{C}^2$ as $\{|0\rangle, |1\rangle\}$, and to identify these basis vectors with the booleans false and true, respectively. The state of a qubit can therefore be thought of as a complex linear combination $\alpha|0\rangle + \beta|1\rangle$ of booleans, called a *quantum superposition*. More generally, the state of $n$ qubits is an element of the $n$-fold tensor product $\mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2$.

There are three kinds of basic operations on quantum data: initializations, unitary maps and measurements. Initialization prepares a new qubit in state $|0\rangle$ or $|1\rangle$. A unitary map, or *gate*, is an invertible linear map $U$ such that $U^* = U^{-1}$; here $U^*$ denotes the complex conjugate transpose of $U$. Finally, the operation of measurement consumes a qubit and returns a classical bit. If $n$ qubits are in state $\alpha|0\rangle \otimes \phi_0 + \beta|1\rangle \otimes \phi_1$, where $\phi_0$ and $\phi_1$ are normalized states of $n-1$ qubits, then measuring the leftmost qubit yields false with probability $|\alpha|^2$, leaving the remaining qubits in state $\phi_0$, and true with probability $|\beta|^2$, leaving the remaining qubits in state $\phi_1$.

**Example 1.** A small algorithm is the simulation of an unbiased coin toss: initialize one quantum bit to $|0\rangle$, apply the Hadamard gate sending $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, then measure. The result is true with probability $\frac{1}{2}$ and false with probability $\frac{1}{2}$.

**Example 2.** A slightly more involved algorithm is the *quantum teleportation algorithm* (see [16] for details). The procedure is summarized in Figure 1. Wires represent the path of quantum bits in the computation, and time flows from left to right. The gate $\boxed{H}$ stands for an application of the Hadamard gate, whereas the gate is a controlled-not: it negates the bottom qubit if the upper one is in state $|1\rangle$. The box $M$ is a measurement. The unitaries $U_{xy}$ are

$$U_{00} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; U_{01} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; U_{10} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \; U_{11} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

The goal is to send a quantum bit in an unknown state $|\phi\rangle$ from Location A to Location B using two classical bits. The procedure can be reversed to send two classical bits using a quantum bit. In this case it is called the *dense coding algorithm* [16].

The algorithm consists of three parts. In (i), two quantum bits (qubits 2 and 3) are entangled in state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. In (ii), the input qubit 1 in state $|\phi\rangle$ is entangled with qubit 2, then both are measured. The result is sent over location B, where in (iii) an correction $U_{xy}$ is applied on qubit 3, setting it to state $|\phi\rangle$.

## 2.2 Density matrices and completely positive maps

If we identify $|0\rangle$ and $|1\rangle$ with the standard basis vectors $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, the state of a qubit can be expressed as a two-dimensional vector $v = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$. Similarly, the state of an $n$-qubit system can be expressed as an $2^n$-dimensional column vector. Often, it is necessary to consider *probability distributions* on quantum states; these are also known as *mixed states*. Consider a quantum system that is in one of several states $v_1, \ldots, v_k$ with probabilities $p_1, \ldots, p_k$, respectively. The *density matrix* of this mixed state is defined to be $A = \sum_i p_i v_i v_i^*$, where $(-)^*$ denotes the adjoint operator. By a theorem of Von Neumann, the density matrix is a good representation of mixed states, in the following sense: two mixed states are indistinguishable by any physical experiment if and only if they have the same density matrix [16]. Note that $\operatorname{tr} A = p_1 + \ldots + p_k$. For our purposes, it is often convenient to permit sub-probability distributions, so that $p_1 + \ldots + p_k \leqslant 1$.

Let us write $\mathbb{C}^{n \times n}$ for the space of $n \times n$-matrices. Recall that a matrix $A \in \mathbb{C}^{n \times n}$ is called *positive* if $v^* A v \geqslant 0$ for all $v \in \mathbb{C}^n$. Given $A, B \in \mathbb{C}^{n \times n}$, we write $A \sqsubseteq B$ iff $B - A$ is positive; this is the so-called *Löwner partial order*. A linear map $F : \mathbb{C}^{n \times n} \to \mathbb{C}^{m \times m}$ is called *positive* if $A \sqsupseteq 0$ implies $F(A) \sqsupseteq 0$, and *completely positive* if $F \otimes \mathrm{id}_k$ is positive for all $k$, where $\mathrm{id}_k$ is the identity function on $\mathbb{C}^{k \times k}$. If $F$ moreover satisfies $\operatorname{tr}(F(A)) \leqslant \operatorname{tr} A$ for all positive $A$, then it is called a *superoperator*. The density matrices are precisely the positive matrices $A$ of trace $\leqslant 1$. Moreover, the superoperators correspond precisely to those functions from mixed states to mixed states that are physically possible [16, 17].

## 2.3 The category CPM

The category $\mathbf{CPM}_s$ is defined as follows: the objects are natural numbers, and a morphism $F : n \to m$ is a completely positive map $F : \mathbb{C}^{n \times n} \to \mathbb{C}^{m \times m}$. Let **CPM** be the free completion of $\mathbf{CPM}_s$ under finite biproducts; specifically, the objects of **CPM** are sequences $\vec{n} = (n_1, \ldots, n_k)$ of natural numbers, and a morphism $F : \vec{n} \to \vec{m}$ is a matrix $(F_{ij})$ of morphisms $F_{ij} : n_j \to m_i$ of $\mathbf{CPM}_s$. The categories $\mathbf{CPM}_s$ and **CPM** are symmetric monoidal, and in fact, compact closed [17].

### 2.4 Limitations of CPM as a model

The category **CPM** can serve as a fully abstract model for a simple, strictly linear, finitary quantum lambda calculus [20]. For example, the type **bit** is interpreted as $(1, 1)$, and the type **qubit** is interpreted as $(2)$. Measurement, as a map from **qubit** to **bit**, sends $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ to $(a, d)$. The coin toss is a map $(1) \to (1, 1)$ sending $(p)$ to $(\frac{p}{2}, \frac{p}{2})$. Function spaces are interpreted via the compact closed structure.

As mentioned in the introduction, the semantics of [20] is extremely limited, because it is completely finitary. Thus recursion, infinite data types, and non-linear functions (i.e., those that can use their argument more than once) had to be completely removed from the language in order to fit the model. For example, even the simple squaring function $f \mapsto \lambda x.f(f\,x)$ is not representable in **CPM**.

The purpose of the present paper is to remove all of these restrictions. As an example, consider the following pseudo-code (in ML-style):

```
val qlist : qubit -> qubit list
let rec qlist q = if (cointoss) then [q]
            else let (x,y) = entangle q in x::(qlist y)
```

Here, `cointoss` is a fair coin toss, and the function `entangle` sends $\alpha|0\rangle + \beta|1\rangle$ to $\alpha|00\rangle + \beta|11\rangle$.

So if the function **qlist** is applied to a qubit $\alpha|0\rangle + \beta|1\rangle$, the output is $\alpha|0\rangle + \beta|1\rangle$ with probability $\frac{1}{2}$, $\alpha|00\rangle + \beta|11\rangle$ with probability $\frac{1}{4}$, $\alpha|000\rangle + \beta|111\rangle$ with probability $\frac{1}{8}$, and so on. Its semantics should be of type $2 \to (2, 4, 8, \ldots)$, mapping

$$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \mapsto \left(\frac{1}{2}\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right), \frac{1}{4}\left(\begin{smallmatrix} a & 0 & 0 & b \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ c & 0 & 0 & d \end{smallmatrix}\right), \ldots\right).$$

The category **CPM** is "almost" capable of handling this case, but not quite, because it cannot express infinite tuples of matrices. The model we propose in this paper is essentially an extension of **CPM** to infinite biproducts, using methods developed in [5, 15, 11, 12].

## 3 A quantum lambda calculus

We define a variant of the typed quantum lambda calculus of [21]. The main difference is that the language in this present paper is a true extension of linear logic (see the type assignment system of Table 2). In particular, in contrast with [21], $!(A \otimes B) \multimap !A \otimes !B$ is not provable and there is no need for a subtyping relation. The operational semantics implements a call-by-value strategy. An untyped call-by-name variant has been studied in [10].

The classes of *terms*, *values* and *types* are defined in Table 1. The symbol $c$ ranges over the set of term constants $\{\mathtt{skip}, \mathtt{split}^A, \mathtt{meas}, \mathtt{new}, U\}$. The constant $U$ ranges over a set of elementary unitary transformations on quantum bits. In the

$$\begin{array}{l} \textit{Terms} \quad M, N, P \quad ::= \\ \qquad x \mid \lambda x^A.M \mid MN \mid \mathtt{skip} \mid M;N \mid \\ \qquad M \otimes N \mid \mathtt{let}\ x^A \otimes y^B = M\ \mathtt{in}\ N \mid \\ \qquad \mathtt{in}_\ell\ M \mid \mathtt{in}_r\ M \mid \mathtt{match}\ P\ \mathtt{with}\ (x^A : M \mid y^B : N) \mid \\ \qquad \mathtt{split}^A \mid \mathtt{letrec}\ f^{A \multimap B} x = M\ \mathtt{in}\ N \mid \mathtt{meas} \mid \mathtt{new} \mid U \\ \\ \textit{Values} \quad V, W \quad ::= \\ \qquad x \mid c \mid \lambda x^A.M \mid V \otimes W \mid \mathtt{in}_\ell\ V \mid \mathtt{in}_r\ W \\ \\ \textit{Types} \quad A, B, C \quad ::= \\ \qquad \mathbf{qubit} \mid A \multimap B \mid !(A \multimap B) \mid 1 \mid A \otimes B \mid A \oplus B \mid A^\ell. \end{array}$$

Table 1: Grammars of terms, values and types.

examples below, we will be using the Hadamard gate $H$ and the controlled-not gate $N_c$, defined as follows [16]:

$$H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right) \qquad N_c = \left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right) \tag{1}$$

Notice that bound variables are given in Church style, i.e., with a type annotation. This enables Proposition 4, and simplifies the semantic interpretation of the typed terms. We omit such annotations in the sequel if uninteresting or obvious.

We have two kinds of arrows: the linear arrow $A \multimap B$, and the intuitionistic arrow $!(A \multimap B)$, which is obtained by the call-by-value translation of the intuitionistic implication into linear logic [3]. Intuitively, only the terms of type $!(A \multimap B)$ represent functions that can be used repeatedly, whereas terms of type $A \multimap B$ must be used exactly once. A type of the form $!A$ is called a $!$-*type* or *non-linear* type, and all other types are called *linear*. The distinction between linear and non-linear types is crucial for allowing the type system to enforce the no-cloning property of quantum physics.

By convention, $\multimap$ is associative to the right, while application and tensor are associative to the left. We use the notation $A^{\otimes n}$ for $A$ tensored $n$ times. The type $A^\ell$ denotes finite lists of type $A$. When doing structural induction on types, we assume that $A^\ell$ is greater than $A^{\otimes n}$, for any $n \in \mathbb{N}$.

The set of terms and types is somewhat spartan; however it can be easily extended by introducing syntactic sugar. Note that, for technical convenience, we have only allowed types of the form $!A$ when $A$ is an arrow type. However, for an arbitrary type $A$, the type $!A$ can be simulated by using $!(1 \multimap A)$ instead.

**Notation 3.** We write $\mathbf{bit} = 1 \oplus 1$, $\mathtt{tt} = \mathtt{in}_r\ \mathtt{skip}$, $\mathtt{ff} = \mathtt{in}_\ell\ \mathtt{skip}$, $\mathtt{nil} = \mathtt{in}_\ell\ \mathtt{skip}$ and $M :: N = \mathtt{in}_r\ (M \otimes N)$. We write $\lambda\mathtt{skip}.M$ for the term $\lambda z^1.(z;M)$, where $z$ is a fresh variable, and $\mathtt{if}\ P\ \mathtt{then}\ M\ \mathtt{else}\ N$ for $\mathtt{match}\ P\ \mathtt{with}\ (x^1 : N \mid y^1 : M)$.

A *context* $\Delta$ is a function from a finite set of variables to types. We denote the domain of $\Delta$ by $|\Delta|$, and we write $\Delta = x_1 : A_1, \ldots, x_n : A_n$ whenever $|\Delta| = \{x_1, \ldots, x_n\}$ and $\Delta(x_i) =$

$$\frac{A \text{ linear}}{!\Delta, x : A \vdash x : A}\ ax \qquad \frac{}{!\Delta, x : !(A \multimap B) \vdash x : A \multimap B}\ axd \qquad \frac{!\Delta \vdash V : A \multimap B \quad V \text{ value}}{!\Delta \vdash V : !(A \multimap B)}\ p \qquad \frac{}{!\Delta \vdash \mathtt{skip} : 1}\ 1_I$$

$$\frac{\Delta, x : A \vdash M : B}{\Delta \vdash \lambda x^A.M : A \multimap B}\ \multimap_I \qquad \frac{!\Delta, \Gamma \vdash M : A \multimap B \quad !\Delta, \Sigma \vdash N : A}{!\Delta, \Gamma, \Sigma \vdash MN : B}\ \multimap_E \qquad \frac{!\Delta, \Gamma \vdash M : 1 \quad !\Delta, \Sigma \vdash N : A}{!\Delta, \Gamma, \Sigma \vdash M;N : A}\ 1_E$$

$$\frac{!\Delta, \Gamma \vdash M : A \quad !\Delta, \Sigma \vdash N : B}{!\Delta, \Gamma, \Sigma \vdash M \otimes N : A \otimes B}\ \otimes_I \qquad \frac{!\Delta, \Gamma \vdash M : A \otimes B \quad !\Delta, \Sigma, x : A, y : B \vdash N : C}{!\Delta, \Gamma, \Sigma \vdash \mathtt{let}\ x^A \otimes y^B \ = \ M \ \mathtt{in}\ N : C}\ \otimes_E$$

$$\frac{!\Delta, \Gamma \vdash M : A}{!\Delta, \Gamma \vdash \mathtt{in}_\ell\ M : A \oplus B}\ \oplus_I^\ell \qquad \frac{!\Delta, \Gamma \vdash M : B}{!\Delta, \Gamma \vdash \mathtt{in}_r\ M : A \oplus B}\ \oplus_I^r \qquad \frac{!\Delta, \Gamma \vdash P : A \oplus B \quad \begin{array}{c} !\Delta, \Sigma, x : A \vdash M : C \\ !\Delta, \Sigma, y : B \vdash N : C \end{array}}{!\Delta, \Gamma, \Sigma \vdash \mathtt{match}\ P\ \mathtt{with}\ (x^A : M \mid y^B : N) : C}\ \oplus_E$$

$$\frac{!\Delta, \Gamma \vdash M : 1 \oplus (A \otimes A^\ell)}{!\Delta, \Gamma \vdash M : A^\ell}\ -_I^\ell \qquad \frac{}{!\Delta \vdash \mathtt{split}^A : A^\ell \multimap 1 \oplus (A \otimes A^\ell)}\ \mathtt{split} \qquad \frac{!\Delta, f : !(A \multimap B), x : A \vdash M : B \quad !\Delta, \Gamma, f : !(A \multimap B) \vdash N : C}{!\Delta, \Gamma \vdash \mathtt{letrec}\ f^{A \multimap B}\ x = M\ \mathtt{in}\ N : C}\ \mathtt{rec}$$

$$\frac{}{!\Delta \vdash \mathtt{meas} : \mathbf{qubit} \multimap \mathbf{bit}}\ \mathtt{meas} \qquad \frac{}{!\Delta \vdash \mathtt{new} : \mathbf{bit} \multimap \mathbf{qubit}}\ \mathtt{new} \qquad \frac{U \text{ of arity } n}{!\Delta \vdash U : \mathbf{qubit}^{\otimes n} \multimap \mathbf{qubit}^{\otimes n}}\ U$$

Table 2: Typing rules. The contexts $\Gamma$ and $\Sigma$ are assumed to be linear.

$A_i$. We call $\Delta$ *exponential* (resp. *linear*) whenever all $A_i$ are !-types (resp. no $A_i$ is a !-type). We write $!\Delta$ for a context that is exponential. The notation $\Gamma, \Sigma$ refers to the union of the two contexts $\Gamma$ and $\Sigma$ and assumes that $|\Gamma|$ and $|\Sigma|$ are disjoint.

A *judgement* is a triple $\Gamma \vdash M : A$ of a context $\Gamma$, a term $M$ and a type $A$. A judgement is called *valid* if it can be inferred from the typing rules in Figure 2, using the convention that the contexts $\Gamma$ and $\Sigma$ are linear.

**Proposition 4.** *There is at most one derivation inferring a given typing judgement* $\Gamma \vdash M : A$. □

**Example 5.** In Section 2.4, we wrote the informal program **qlist**. Our language is expressive enough to represent it. The term `cointoss` can be defined as $\mathtt{meas}(H(\mathtt{new}\,\mathtt{tt}))$, and it has type **bit**. The term `entangle` is $\lambda x^{\mathbf{qubit}}.N_c(x \otimes (\mathtt{new}\,\mathtt{ff}))$, which has type **qubit** $\multimap$ **qubit** $\otimes$ **qubit**. Then, **qlist** is

$$\begin{array}{l} \mathtt{letrec}\ f^{\mathbf{qubit} \multimap \mathbf{qubit}^\ell}\ q = \\ \qquad \mathtt{if}\ \mathtt{cointoss}\ \mathtt{then}\ q :: \mathtt{nil} \\ \qquad \mathtt{else}\ \mathtt{let}\ x^{\mathbf{qubit}} \otimes y^{\mathbf{qubit}} = \mathtt{entangle}\ q\ \mathtt{in}\ x :: fy \end{array}$$

which has type **qubit** $\multimap$ $\mathbf{qubit}^\ell$. In Examples 9 and 28 we discuss its operational and denotational semantics, respectively.

**Example 6.** In Example 2 and Figure 1, we sketched the quantum teleportation algorithm. We said that the algorithm can be decomposed into 3 parts. Each of these parts can be described and typed in the quantum lambda calculus, yielding a higher-order term. This is an adaptation of an example provided in [19].

(i) generates an EPR pair of entangled quantum bits. Its type is therefore $1 \multimap \mathbf{qubit} \otimes \mathbf{qubit}$. The corresponding term is

$$\mathbf{EPR} = \lambda \mathtt{skip}.N_c\left((H(\mathtt{new}\,\mathtt{ff})) \otimes (\mathtt{new}\,\mathtt{ff})\right).$$

(ii) performs a Bell measurement on two quantum bits and outputs two classical bits $x, y$. Its type is thus $\mathbf{qubit} \multimap \mathbf{bit} \otimes \mathbf{bit}$, and the term **BellMeasure** is defined as

$$\lambda q_1.\lambda q_2.\left(\begin{array}{l} \mathtt{let}\ x \otimes y = N_c\,(q_1 \otimes q_2) \\ \mathtt{in}\ (\mathtt{meas}\,(H\,x)) \otimes (\mathtt{meas}\,y) \end{array}\right).$$

(iii) performs a correction. It takes one quantum bit, two classical bits, and outputs a quantum bit. It has a type of the form $\mathbf{qubit} \multimap \mathbf{bit} \otimes \mathbf{bit} \multimap \mathbf{qubit}$. The term is

$$\begin{array}{rl} \mathbf{U} = \lambda q.\lambda x \otimes y.\mathtt{if}\ x\ \mathtt{then}\ (\mathtt{if}\ y\ \mathtt{then}\ U_{11}\,q\ \mathtt{else}\ U_{10}\,q) & \\ \mathtt{else}\ (\mathtt{if}\ y\ \mathtt{then}\ U_{01}\,q\ \mathtt{else}\ U_{00}\,q). & \end{array}$$

We can now write the term

$$\begin{array}{rll} \mathbf{telep} = & \lambda \mathtt{skip}.\mathtt{let}\ x \otimes y & = \mathbf{EPR}\ \mathtt{skip}\ \mathtt{in} \\ & \mathtt{let}\ f & = \mathbf{BellMeasure}\ x\ \mathtt{in} \\ & \mathtt{let}\ g & = \mathbf{U}\ y \\ & \mathtt{in}\ f \otimes g. & \end{array}$$

It can then be shown that

$$\vdash \mathbf{telep} :!(1 \multimap (\mathbf{qubit} \multimap \mathbf{bit} \otimes \mathbf{bit}) \otimes (\mathbf{bit} \otimes \mathbf{bit} \multimap \mathbf{qubit}))$$

is a valid typing judgement. In other words, the teleportation algorithm produces a pair of entangled functions $f : \mathbf{qubit} \to \mathbf{bit} \otimes \mathbf{bit}$ and $g : \mathbf{bit} \otimes \mathbf{bit} \to \mathbf{qubit}$. These functions have the property that $g(f(|\phi\rangle)) = |\phi\rangle$ for all qubits $|\phi\rangle$, and $f(g(x \otimes y)) = (x \otimes y)$ for all booleans $x$ and $y$. These two functions are each other's inverse, but because they contain an embedded qubit each, they can only be used once. They can be said to form a "single-use isomorphism" between the (otherwise non-isomorphic) types **qubit** and **bit** $\otimes$ **bit**. However, the whole procedure is duplicable: one can generate as many one-time-use isomorphism pairs as desired.

## 3.1 Operational semantics

The operational semantics is defined in terms of an abstract machine simulating the behavior of Knill's QRAM model [8]. It is similar to the semantics given in [21].

**Definition 7.** A *quantum closure* is a triple $[q, \ell, M]$ where

- $q$ is a normalized vector of $\mathbb{C}^{2^n}$, for some integer $n \geqslant 0$. The vector $q$ is called the *quantum state*;
- $M$ is a term, not necessarily closed;
- $\ell$ is a one-to-one map from the set of free variables of $M$ to the set $\{1, \dots, n\}$. It is called the *linking function*.

We write $|\ell|$ for the domain of $\ell$. By abuse of language we may call a closure $[q, \ell, V]$ a *value* when the term $V$ is a value. We denote the set of quantum closures by Cl and the set of quantum closures that are values by Val. We write $\ell|_M$ for the linking function whose domain is restricted to the set of free variables of $M$. We say that the quantum closure $[q, \ell, M]$ is *total* when $|\ell|$ has cardinality $n$, the size of the quantum state. In that case, if $|\ell| = \{x_1, \dots, x_n\}$ and $\ell(x_i) = i$, we write $\ell$ as $|x_1, \dots, x_n\rangle$. A quantum closure $[q, |x_1, \dots, x_n\rangle, M]$ *has a type* $A$, whenever $x_1 : \mathbf{qubit}, \dots, x_n : \mathbf{qubit} \vdash M : A$. In case $\ell = |x_1, \dots, x_n\rangle$ we can also write $\ell \vdash M : A$.

The purpose of a quantum closure is to provide a mechanism to talk about terms with embedded quantum data. The idea is that a variable $y \in \mathrm{FV}(M)$ is bound in the closure $[q, \ell, M]$ to qubit number $\ell(y)$ of the quantum state $q$. So for example, the quantum closure $[\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), |x_1, x_2\rangle, \lambda y^A.yx_1x_2]$ denotes a term $\lambda y^A.yx_1x_2$ with two embedded qubits $x_1, x_2$ in the entangled state $|x_1x_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

The notion of $\alpha$-equivalence extends naturally to quantum closures, for instance, the states $[q, |x\rangle, \lambda y^A.x]$ and $[q, |z\rangle, \lambda y^A.z]$ are equivalent. From now on, we tacitly identify quantum closures up to renaming of bound variables.

The evaluation of a term is defined as a probabilistic rewriting procedure on quantum closures, using a call-by-value reduction strategy. We use the notation $[q, \ell, M] \xrightarrow{p} [q', \ell', M']$ to mean that the left-hand side closure reduces in one step to the right-hand side with probability $p \in [0, 1]$.

**Definition 8.** The reduction rules are shown in Table 3. The rules split into three categories: (a) rules handling the classical part of the calculus; (b) rules dealing with quantum data; and (c) congruence rules for the call-by-value strategy. Note that in the statement of the rules, $V$ and $W$ refer to values.

In the rules in Table 3(b), the quantum state $q$ has size $n$. The quantum state $q'$ in the first rule is obtained by applying the $k$-ary unitary gate $U$ to the qubits $\ell(x_1), \dots, \ell(x_k)$. Precisely, $q' = (\sigma \circ (U \otimes \mathrm{id}) \circ \sigma^{-1})(q)$, where $\sigma$ is the action on $\mathbb{C}^{2^n}$ of any permutation over $\{1, \dots, n\}$ such that $\sigma(i) = \ell(x_i)$ whenever $i \leqslant k$. In the rules about measurements, we assume that if $q_0$ and $q_1$ are normalized quantum states of the form

$$\textstyle\sum_j \alpha_j |\phi_j\rangle \otimes |0\rangle \otimes |\psi_j\rangle, \ \sum_j \beta_j |\phi_j\rangle \otimes |1\rangle \otimes |\psi_j\rangle, \qquad (2)$$

then $q_0'$ and $q_1'$ are respectively

$$\textstyle\sum_j \alpha_j |\phi_j\rangle \otimes |\psi_j\rangle, \ \sum_j \beta_j |\phi_j\rangle \otimes |\psi_j\rangle, \qquad (3)$$

where the vectors $\phi_j$ have dimension $\ell(x) - 1$ (so that the measured qubit is $\ell(x)$).

In summary, the quantum state acts as a shared global store that is updated destructively by the various quantum operations.

Note that the only probabilistic reduction step is the one corresponding to measurement. Also, we underline that the hypothesis associated with a congruence rule $[q, \ell, C[M]] \xrightarrow{p} [q', \ell', C[M']]$ takes into account the whole quantum states $q$ and $q'$. In fact, because of the entanglement, the evaluation of $[q, \ell|_M, M]$ may have a side-effect on the state of the qubits pointed to by the variables occurring in the context $C[\,]$.

The rules assume that the involved closures are well-defined. In particular, whenever $[q, \ell, M] \xrightarrow{p} [q, \ell, M']$, the two terms $M$ and $M'$ have the same free variables. For example, the closure $[|00\rangle, |yz\rangle, (\lambda x.y)z]$ cannot reduce and it represents an error: it would reduce to the erroneous quantum closure $[|00\rangle, |yz\rangle, z]$, where the domain of the linking function is not the set of free variables, as specified by Definition 7. The type system will prevent such an error as proven in Proposition 12.

**Example 9.** Recall Example 5. We have $[|\rangle, |\rangle, \texttt{cointoss}] \xrightarrow{1} [|1\rangle, |x\rangle, \texttt{meas}(Hx)] \xrightarrow{1} [\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |x\rangle, \texttt{meas}\ x]$, the latter reducing to either $[|\rangle, |\rangle, \texttt{tt}]$ or $[|\rangle, |\rangle, \texttt{ff}]$, with equal probability $\frac{1}{2}$. As for $\texttt{entangle}$, we have that

$$\begin{aligned} & [\alpha|0\rangle + \beta|1\rangle, |x\rangle, \texttt{entangle}\ x] \\ \xrightarrow{1}\ & [\alpha|0\rangle + \beta|1\rangle, |x\rangle, N_c(x \otimes (\texttt{new ff}))] \\ \xrightarrow{1}\ & [\alpha|00\rangle + \beta|10\rangle, |xy\rangle, N_c(x \otimes y)] \\ \xrightarrow{1}\ & [\alpha|00\rangle + \beta|11\rangle, |xy\rangle, x \otimes y]. \end{aligned}$$

Similarly, one can check that $[\alpha|0\rangle + \beta|1\rangle, |q\rangle, \texttt{qlist}\, q]$ behaves as described in Section 2.4, reducing to $[\alpha|0\rangle + \beta|1\rangle, |q\rangle, q :: \texttt{nil}]$ with probability $\frac{1}{2}$, to $[\alpha|00\rangle + \beta|11\rangle, |qq'\rangle, q' :: q :: \texttt{nil}]$ with probability $\frac{1}{4}$, etc. In particular, notice that in any single reduction sequence the variable $q$ has not been duplicated, as correctly asserted by the type of $\texttt{qlist}$.

**Lemma 10** (Substitution)**.** *Suppose* $!\Delta, \Gamma, x : A \vdash M : B$ *and* $!\Delta, \Sigma \vdash V : A$*, where* $\Gamma$ *and* $\Sigma$ *are linear contexts with disjoint domain. Then* $!\Delta, \Gamma, \Sigma \vdash M\{V/x\} : B$. ☐

**Proposition 11** (Subject reduction)**.** *When* $[q, |y_1 \dots y_n\rangle, M] \xrightarrow{p} [q', |x_1 \dots x_{n'}\rangle, M']$ *and* $y_1 : \mathbf{qubit}, \dots, y_n : \mathbf{qubit} \vdash M : A$*, then* $x_1 : \mathbf{qubit}, \dots, x_{n'} : \mathbf{qubit} \vdash M' : A$. ☐

**Proposition 12** (Type safety)**.** *If* $[q, \ell, M]$ *is typable then either* $M$ *is a value or there is a closure* $[q', \ell', M']$ *such that* $[q, \ell, M] \xrightarrow{p} [q', \ell', M']$*. Moreover, if* $M$ *is not a value, the total probability of all possible single-step reductions from* $[q, \ell, M]$ *is* 1. ☐

**Lemma 13** (Totality)**.** *If* $[q, \ell, M] \xrightarrow{p} [q', \ell', M']$ *and* $[q, \ell, M]$ *is total, then* $[q', \ell', M']$ *is total too.*

*Proof.* By induction on a derivation of $[q, \ell, M] \xrightarrow{p} [q', \ell', M']$, one proves that $\dim(q') = \dim(q) + \dim(\ell') - \dim(\ell)$ where $\dim(q)$ is the size of the quantum state $q$ and $\dim(\ell)$ is the cardinality of the domain set of the linking function $\ell$. Then, one gets the statement, since $[q, \ell, M]$ is total iff $\dim(q) = \dim(\ell)$. ☐

$$[q,\ell,(\lambda x^A.M)\,V] \xrightarrow{1} [q,\ell,M\{V/x\}] \qquad [q,\ell,\texttt{let}\ x^A\otimes y^B \ = \ V\otimes W\ \texttt{in}\ N] \xrightarrow{1} [q,\ell,N\{V/x,W/y\}]$$

$$[q,\ell,\texttt{skip};N] \xrightarrow{1} [q,\ell,N] \qquad [q,\ell,\texttt{match}\ (\texttt{in}_\ell\ V)\ \texttt{with}\ (x^A:M \mid y^B:N)] \xrightarrow{1} [q,\ell,M\{V/x\}]$$

$$[q,\ell,\texttt{split}\,V] \xrightarrow{1} [q,\ell,V] \qquad [q,\ell,\texttt{match}\ (\texttt{in}_r\ V)\ \texttt{with}\ (x^A:M \mid y^B:N)] \xrightarrow{1} [q,\ell,N\{V/y\}]$$

$$[q,\ell,\texttt{letrec}\ f^{A\multimap B}\ x = M\ \texttt{in}\ N] \xrightarrow{1} [q,\ell,N\{(\lambda x^A.\texttt{letrec}\ f^{A\multimap B}\ x = M\ \texttt{in}\ M)/f\}]$$

(a) Classical control.

$$[q,\ell,U(x_1\otimes\cdots\otimes x_k)] \xrightarrow{1} [q',\ell,x_1\otimes\cdots\otimes x_k]$$

$$[q,\emptyset,\texttt{new ff}] \xrightarrow{1} [q\otimes|0\rangle,\{y\mapsto n+1\},y] \qquad [\alpha q_0+\beta q_1,\{x\mapsto i\},\texttt{meas}\ x] \xrightarrow{|\beta|^2} [q_1',\emptyset,\texttt{tt}]$$

$$[q,\emptyset,\texttt{new tt}] \xrightarrow{1} [q\otimes|1\rangle,\{y\mapsto n+1\},y] \qquad [\alpha q_0+\beta q_1,\{x\mapsto i\},\texttt{meas}\ x] \xrightarrow{|\alpha|^2} [q_0',\emptyset,\texttt{ff}]$$

(b) Quantum data. The variable $y$ is fresh. The decomposition of the quantum array in the case of $\texttt{meas}\ x$ is explained in Definition 8.

$$[q,\ell,MN] \xrightarrow{p} [q',\ell',M'N] \qquad [q,\ell,M\otimes N] \xrightarrow{p} [q',\ell',M'\otimes N] \qquad [q,\ell,\texttt{in}_\ell\ M] \xrightarrow{p} [q',\ell',\texttt{in}_\ell\ M']$$

$$[q,\ell,VM] \xrightarrow{p} [q',\ell',VM'] \qquad [q,\ell,V\otimes M] \xrightarrow{p} [q',\ell',V\otimes M'] \qquad [q,\ell,\texttt{in}_r\ M] \xrightarrow{p} [q',\ell',\texttt{in}_r\ M']$$

$$[q,\ell,M;N] \xrightarrow{p} [q',\ell',M';N] \qquad [q,\ell,\texttt{let}\ x^A\otimes y^B \ = \ M\ \texttt{in}\ N] \xrightarrow{p} [q',\ell',\texttt{let}\ x^A\otimes y^B \ = \ M'\ \texttt{in}\ N]$$

$$[q,\ell,\texttt{match}\ M\ \texttt{with}\ (x^A:P \mid y^B:N)] \xrightarrow{p} [q',\ell',\texttt{match}\ M'\ \texttt{with}\ (x^A:P \mid y^B:N)]$$

(c) Congruence rules, under the hypothesis that for some $\ell_0$ we have $\ell=\ell_0\uplus\ell|_M$, $\ell'=\ell_0\uplus\ell'|_{M'}$ and $[q,\ell|_M,M]\xrightarrow{p}[q',\ell'|_{M'},M']$.

Table 3: Reduction rules on closures.

**Notation 14.** The reduction relation $\to$ defines the probability that a closure reduces to another one in a single step. We extend this relation to an arbitrary large (but finite) number of reduction steps with the notation $\mathrm{Red}^n_{[q,\ell,M],[q',\ell',V]}$: it is the total probability of $[q,\ell,M]$ reducing to a value $[q',\ell',V]$. It is defined as the sum of all $\prod_{i=1}^m p_i$, where $[q,\ell,M] \xrightarrow{p_1} [q_1,\ell_1,M_1]\cdots \xrightarrow{p_m} [q,\ell',V]$ is a finite reduction sequence of $m\leqslant n$ steps. We write $\mathrm{Red}^\infty_{[q,\ell,M],[q',\ell',V]}$ for the sup over $n$ of $\mathrm{Red}^n_{[q,\ell,M],[q',\ell',V]}$. Finally, we define the *total probability* $\mathrm{Halt}_{[q,\ell,M]}$ of $[q,\ell,M]$ converging to any value as $\sum_{[q',\ell',V]\in\mathrm{Val}}\mathrm{Red}^\infty_{[q,\ell,M],[q',\ell',V]}$.

# 4 Denotational semantics

We interpret the quantum lambda calculus in a suitable extension $\overline{\mathbf{CPMs}}^{\oplus}$ of the category **CPM** described in Section 2. What **CPM** essentially misses is the linear logic exponential $!A$, and our plan is to introduce it via the equation

$$!A := \bigoplus_{k=0}^{\infty} A^{\odot k}, \tag{4}$$

where $\bigoplus_{k=0}^{\infty}$ is the infinite biproduct of the family $\{A^{\odot k}\}_k$, each $A^{\odot k}$ being the symmetric $k$-fold tensor power of $A$, i.e., the equalizer of the $k!$ symmetries of the $k$-ary tensor $A^{\otimes k} := A\otimes\cdots\otimes A$.

The category **CPM** cannot express this equation because it lacks both infinite biproducts and a convenient definition of symmetric tensor powers. The category $\overline{\mathbf{CPMs}}^{\oplus}$ is in some sense the minimal extension of **CPM** having these two missing ingredients.

The plan of the section is as follows. Section 4.1 presents some preliminary material. Section 4.2 defines $\overline{\mathbf{CPMs}}^{\oplus}$ and Section 4.3 develops the categorical structure allowing us to interpret the quantum lambda calculus. Section 4.4 sketches the proof of the soundness of the model with respect to the operational semantics. Finally, Section 4.5 discusses the denotations of the programs **qlist** and **teleport**.

## 4.1 Preliminaries: from CPM to $\overline{\mathbf{CPMs}}$

**Permutation groups.** Let $S_n$ be the symmetric group of degree $n$, i.e., the group of permutations of $n=\{0,\ldots,n-1\}$. Any permutation $g\in S_n$ gives rise to a matrix $P_g\in\mathbb{C}^{n\times n}$, defined by $P_g(e_i)=e_{g(i)}$, where $e_i$ is the $i$th standard basis vector. We define an action of $g$ on $\mathbb{C}^{n\times n}$ by $g\cdot M := P_g M P_g^{-1}$. Moreover, for a subgroup $G\subseteq S_n$, we define

$$G\cdot M := \frac{1}{\#G}\sum_{g\in G} g\cdot M, \tag{5}$$

where $\#G$ is the number of elements of $G$.

**Lemma 15.** *Given a subgroup $G\subseteq S_n$, its action on $\mathbb{C}^{n\times n}$ is idempotent (i.e., $G\cdot G\cdot M = G\cdot M$ for all $M$) and completely positive.*

*Proof.* For the idempotence, notice that for every $g\in G$, $gG=G$, therefore: $G\cdot G\cdot M = \frac{1}{\#G}\sum_{g\in G} gG\cdot M = G\cdot M$. The complete positivity of $G$ is derived from the complete positivity of each map $M\mapsto g\cdot M = P_g M P_g^{-1}$. ☐

In the sequel, we use the notation $G$ both for a subgroup of $S_n$ and for the completely positive map defined by it. The above Lemma allows us to define the set of completely positive maps from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{m\times m}$ invariant under the actions of two subgroups $G\subseteq S_n$, $H\subseteq S_m$ by

$$\mathbf{CPMs}(G,H) := \{f\in\mathbf{CPM}(n,m) \mid G\,;f\,;H = f\},$$

where $f;g$ is the diagrammatic composition $(f;g)(x) = g(f(x))$, and $\mathbf{CPM}(n,m)$ is the set of completely positive maps from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{m\times m}$.

**Completion of the Löwner positive cone.** The set $\mathbf{CPMs}(G,H)$ is a module over the semi-ring $\mathbb{R}^+$ of the non-negative real numbers. The Löwner order $\sqsubseteq$ on completely positive maps [17] endows this module with the structure of a *bounded* directed complete partial order (bdcpo), i.e., there is a minimum element (the zero function $\mathbf{0}$), and any directed set $D$ that is bounded (i.e., such that there exists $f \in \mathbf{CPMs}(G,H)$ such that for all $g \in D$, $g \sqsubseteq f$) has a least upper bound $\bigvee D \in \mathbf{CPMs}(G,H)$. However there exist unbounded directed subsets in $\mathbf{CPMs}(G,H)$. We therefore need to complete $\mathbf{CPMs}(G,H)$ to a dcpo.

The relevant construction is the *D-completion* of [23], which we briefly recall. Given any poset $P$, say that a subset $X$ is *Scott-closed* if it is down-closed and for every directed $I \subseteq S$, if the least upper bound $\bigvee I$ exists in $P$, then $\bigvee I \in S$. We say that a monotone function between posets $f : P \to Q$ is *Scott-continuous* if it preserves all *existing* least upper bounds of directed subsets. Let $\Gamma(P)$ be the set of Scott-closed subsets of $P$; this forms a dcpo under the subset ordering. The *D-completion* $c(P)$ is defined to be the smallest sub-dcpo of $\Gamma(P)$ containing all sets of the form $\downarrow x$. Then $c(P)$ is a dcpo, and there is a canonical injective Scott-continuous map $\iota : P \to c(P)$, defined by $\iota(x) = \downarrow x$, which allows us to regard $P$ as a subset of $c(P)$. The D-completion preserves all existing least upper bounds of directed sets, is idempotent, and satisfies the following universal property: given any other dcpo $E$ and Scott-continuous map $f : P \to E$, there exists a unique Scott-continuous $g : c(P) \to E$ such that $f = \iota\,;g$. It follows that the D-completion is functorial. Moreover, if $P$ is a bounded directed complete partial order, then $P$ is an initial subset of $c(P)$, i.e., the only new elements added by the completion are "at infinity". We call these the *infinite* elements of $c(P)$.

The homset $\mathbf{CPMs}(G,H)$ is then extended by D-completion, namely, $\overline{\mathbf{CPMs}}(G,H) := c(\mathbf{CPMs}(G,H))$. The categorical operations are extended in the unique Scott-continuous way, using the universal property of D-completion. This allows us to define indexed sums over $\overline{\mathbf{CPMs}}(G,H)$, as follows. If $\{f_i\}_{i\in I} \subseteq \overline{\mathbf{CPMs}}(G,H)$ is a (possibly infinite) indexed family, $\sum_{i\in I} f_i$ is defined as $\bigvee_{F\subseteq_{\text{fin}} I}(\sum_{i\in F} f_i)$. Indeed, the set $\{\sum_{i\in F} f_i \;;\; F \subseteq_{\text{fin}} I\}$ is always directed, so has a least upper bound in the order completion $\overline{\mathbf{CPMs}}(G,H)$ of $\mathbf{CPMs}(G,H)$.

## 4.2 The category $\overline{\mathbf{CPMs}}^{\oplus}$

Given a set $A$ and $a, a' \in A$, define the *Kronecker symbol* $\delta_{a,a'} \in \mathbb{N}$ which takes value 1 if $a = a'$ and 0 if $a \neq a'$.

**Objects** are given by indexed families $\mathfrak{A} = \{(d^{\mathfrak{A}}_a, G^{\mathfrak{A}}_a)\}_{a\in|\mathfrak{A}|}$, where the index set $|\mathfrak{A}|$ is called the *web* of $\mathfrak{A}$ and, for every $a \in |\mathfrak{A}|$, $d^{\mathfrak{A}}_a$ is a natural non-negative integer, and $G^{\mathfrak{A}}_a$ a subgroup of permutations of degree $d^{\mathfrak{A}}_a$, called respectively the *dimension* and the *permutation group* of $\mathfrak{A}_a$.

**Morphisms** from $\mathfrak{A}$ to $\mathfrak{B}$ are matrices $\phi$ indexed by $|\mathfrak{A}| \times |\mathfrak{B}|$ and such that $\phi_{a,b} \in \overline{\mathbf{CPMs}}(G^{\mathfrak{A}}_a, G^{\mathfrak{B}}_b)$.

**Composition** of $\phi \in \overline{\mathbf{CPMs}}^{\oplus}(\mathfrak{A},\mathfrak{B})$ and $\psi \in \overline{\mathbf{CPMs}}^{\oplus}(\mathfrak{B},\mathfrak{C})$ is the matrix $\phi\,;\psi$ defined by, for $a \in |\mathfrak{A}|$ and $c \in |\mathfrak{C}|$, $(\phi\,;\psi)_{a,c} := \sum_{b\in|\mathfrak{B}|} \phi_{a,b}\,;\psi_{b,c}$.

**Identity** is the diagonal matrix built with the symmetries of $\mathfrak{A}$, i.e., for $a, a' \in |\mathfrak{A}|$, $\mathrm{id}^{\mathfrak{A}}_{a,a'} := \delta_{a,a'} G^{\mathfrak{A}}_a$.

The description of the objects and the morphisms as indexed families is crucial for inferring the structure of a compact closed Lafont category (Section 4.3). However, it is worthwhile to notice that $\overline{\mathbf{CPMs}}^{\oplus}$ can also be presented as a concrete category of modules and linear maps between modules. Let us sketch such an alternative presentation.

Let $\mathfrak{A}$ be an object of $\overline{\mathbf{CPMs}}^{\oplus}$. We define a module $\mathrm{Pos}(\mathfrak{A})$ over $\overline{\mathbb{R}^+} = \mathbb{R}^+ \cup \{\infty\}$ as follows. For every $a$ in $|\mathfrak{A}|$, let us write $\mathrm{Pos}(a)$ for the cone of the positive matrices in $G^{\mathfrak{A}}_a(\mathbb{C}^{d^{\mathfrak{A}}_a\times d^{\mathfrak{A}}_a})$, this latter being the subspace of the matrices in $\mathbb{C}^{d^{\mathfrak{A}}_a\times d^{\mathfrak{A}}_a}$ invariant under $G^{\mathfrak{A}}_a$. This positive cone $\mathrm{Pos}(a)$ is an $\mathbb{R}^+$-module. We then define:

$$\mathrm{Pos}(\mathfrak{A}) := \bigoplus_{a\in|\mathfrak{A}|} (c(\mathrm{Pos}(a))\}). \tag{6}$$

In fact, we have that $\mathrm{Pos}(a) \simeq \mathbf{CPMs}(S_1, G^{\mathfrak{A}}_a)$ and $\mathrm{Pos}(\mathfrak{A}) \simeq \bigoplus_{a\in|\mathfrak{A}|} \overline{\mathbf{CPMs}}(S_1, G^{\mathfrak{A}}_a)$. Hence, $\mathrm{Pos}(\mathfrak{A})$ is a continuous module over $\overline{\mathbb{R}^+}$: addition and scalar multiplication are defined pointwise and are continuous operations with respect to the Löwner order.

Let $f : \mathrm{Pos}(\mathfrak{A}) \to \mathrm{Pos}(\mathfrak{B})$ be a continuous module homomorphism. We say that $f$ is *completely positive* if all the module homomorphisms $f_{a,b} = \iota^a\,;f\,;\pi^b$ are completely positive maps, for all $a \in |\mathfrak{A}|$ and $b \in |\mathfrak{B}|$. (Indeed, since the positive matrices span the complex vector space of square matrices (of corresponding size), one can canonically extend the definition of complete positivity to module homomorphisms $\mathrm{Pos}(a) \to \mathrm{Pos}(b)$).

**Proposition 16.** *There is an isomorphism between the homset $\overline{\mathbf{CPMs}}^{\oplus}(\mathfrak{A},\mathfrak{B})$ and the continuous module homomorphisms from $\mathrm{Pos}(\mathfrak{A})$ to $\mathrm{Pos}(\mathfrak{B})$ that are completely positive.* □

## 4.3 $\overline{\mathbf{CPMs}}^{\oplus}$ as a model of the quantum lambda calculus

A compact closed category is a special case of symmetric monoidal closed category. A symmetric monoidal closed category with finite products, such that each object has a corresponding free commutative comonoid, is called a *Lafont category*, which is known to be a model of intuitionistic linear logic [9, 14]. The category $\overline{\mathbf{CPMs}}^{\oplus}$ can be endowed with such a structure, as we will show in Sections 4.3.1–4.3.4 below. We can therefore interpret the quantum lambda calculus in $\overline{\mathbf{CPMs}}^{\oplus}$.

The denotation $[\![A]\!]$ of a type $A$ is an object of $\overline{\mathbf{CPMs}}^{\oplus}$. In case $A$ is the ground type (i.e., $1$, $\mathbf{qubit}$), its denotation is:

$$|[\![\mathbf{qubit}]\!]| := \{\star\}, \quad d^{[\![\mathbf{qubit}]\!]}_{\star} := 2, \quad G^{[\![\mathbf{qubit}]\!]}_{\star} := \{\mathrm{id}\},$$
$$|[\![1]\!]| := \{\star\}, \quad d^{[\![1]\!]}_{\star} := 1, \quad G^{[\![1]\!]}_{\star} := \{\mathrm{id}\}.$$

The denotation of the other types is given by structural induction, following the compact closed Lafont structure of $\overline{\mathbf{CPMs}}^{\oplus}$. We

$$!\Delta \otimes A \xrightarrow{\mathtt{w}\otimes\mathrm{id}} 1 \otimes A \simeq A$$
(a) $!\Delta, x : A \vdash x : A$

$$!\Delta \otimes !A \xrightarrow{\mathtt{w}\otimes\mathtt{d}} 1 \otimes A \simeq A$$
(b) $!\Delta, x : !A \vdash x : A$

$$!\Delta \xrightarrow{\mathtt{dig}} !!\Delta \xrightarrow{\mathtt{m}} !(!\Delta) \xrightarrow{!\phi} !A$$
(c) $!\Delta, \vdash V : !A$

$$!\Delta \otimes \Gamma \xrightarrow{\Lambda(\phi)} A \multimap B$$
(d) $!\Delta, \Gamma \vdash \lambda x^A.M : A \multimap B$

$$!\Delta \otimes \Gamma \otimes \Sigma \xrightarrow{\mathtt{c}\otimes\mathrm{id}} !\Delta \otimes \Gamma \otimes !\Delta \otimes \Sigma \xrightarrow{\phi\otimes\psi} A \otimes A \multimap B \xrightarrow{\mathtt{eval}} B$$
(e) $!\Delta, \Gamma, \Sigma \vdash MN : B$

$$!\Delta \xrightarrow{\mathtt{w}} 1$$
(f) $!\Delta \vdash \mathtt{skip} : 1$

$$!\Delta \otimes \Gamma \otimes \Sigma \xrightarrow{\mathtt{c}\otimes\mathrm{id}} !\Delta \otimes \Gamma \otimes !\Delta \otimes \Sigma \xrightarrow{\phi\otimes\mathrm{id}} 1 \otimes !\Delta \otimes \Sigma \simeq !\Delta \otimes \Sigma \xrightarrow{\psi} A$$
(g) $!\Delta, \Gamma, \Sigma \vdash M;N : A$

$$!\Delta \otimes \Gamma \otimes \Sigma \xrightarrow{\mathtt{c}\otimes\mathrm{id}} !\Delta \otimes \Gamma \otimes !\Delta \otimes \Sigma \xrightarrow{\phi\otimes\psi} A \otimes B$$
(h) $!\Delta, \Gamma, \Sigma \vdash M \otimes N : A \otimes B$

$$!\Delta \otimes \Gamma \otimes \Sigma \xrightarrow{\mathtt{c}\otimes\mathrm{id}} !\Delta \otimes \Gamma \otimes !\Delta \otimes \Sigma \xrightarrow{\phi\otimes\mathrm{id}} A \otimes B \otimes !\Delta \otimes \Sigma \xrightarrow{\psi} C$$
(i) $!\Delta, \Gamma, \Sigma \vdash \mathtt{let}\; x^A \otimes y^B \;=\; M \;\mathtt{in}\; N : C$

$$!\Delta \otimes \Gamma \xrightarrow{\phi} A \xrightarrow{\iota^\ell} A \oplus B$$
(j) $!\Delta, \Gamma \vdash \mathtt{in}_\ell\; M : A \oplus B$

$$!\Delta \otimes \Gamma \xrightarrow{\phi} B \xrightarrow{\iota^r} A \oplus B$$
(k) $!\Delta, \Gamma \vdash \mathtt{in}_r\; M : A \oplus B$

$$!\Delta \otimes \Gamma \otimes \Sigma \xrightarrow{\mathtt{c}\otimes\mathrm{id}} !\Delta \otimes \Gamma \otimes !\Delta \otimes \Sigma \xrightarrow{\psi\otimes\mathrm{id}} (A \oplus B) \otimes !\Delta \otimes \Sigma \xrightarrow{\mathtt{distr}} (A \otimes !\Delta \otimes \Sigma) \oplus (B \otimes !\Delta \otimes \Sigma) \xrightarrow{\phi_A \oplus \phi_B} C$$
(l) $!\Delta, \Gamma, \Sigma \vdash \mathtt{match}\; M \;\mathtt{with}\; (x^A : N \mid y^B : L) : C$

$$!\Delta \otimes \Gamma \xrightarrow{\phi} 1 \oplus (A \otimes A^\ell) \xrightarrow{\mathrm{id}\oplus\mathtt{distr}} 1 \oplus (\textstyle\bigoplus_{n=1}^\infty A^{\otimes n}) = A^\ell$$
(m) $!\Delta, \Gamma \vdash M : A^\ell$

$$!\Delta \otimes \Gamma \xrightarrow{\mathtt{c}} !\Delta \otimes \Gamma \otimes !\Delta \xrightarrow{\mathrm{id}\otimes \mathrm{Y}(\mathtt{dig};\mathtt{m};!(\Lambda\phi))} !\Delta \otimes \Gamma \otimes !(A \multimap B) \xrightarrow{\psi} C$$
(n) $!\Delta, \Gamma \vdash \mathtt{letrec}\; f\, x = M \;\mathtt{in}\; N : C$

Table 5: Sketch of the interpretation of the typing judgements, using the Lafont structure of $\overline{\mathbf{CPMs}}^{\oplus}$ defined in Section 4.3. The morphisms $\phi$, $\psi$, $\phi_A$, $\phi_B$ refer to the denotation of the premises of the unique derivation concluding a typing judgement. In (c) and (n), the morphism $\mathtt{m}$ stands for $\mathtt{m}^{\mathbf{1}}$ or the suitable sequence of $\mathtt{m}^{\otimes}$, depending on the context $!!\Delta$.

$$[\![\mathtt{meas}]\!]^{!\Delta\vdash\mathbf{qubit}\multimap\mathbf{bit}}_{\vec{m},(*,b)} = \left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right) \mapsto \begin{cases}\alpha & \text{if } \vec{m} = \vec{[\,]} \text{ and } b = \mathtt{ff},\\ \delta & \text{if } \vec{m} = \vec{[\,]} \text{ and } b = \mathtt{tt},\\ 0 & \text{otherwise.}\end{cases}$$

$$[\![\mathtt{new}]\!]^{!\Delta\vdash\mathbf{bit}\multimap\mathbf{qubit}}_{\vec{m},(b,*)} = \alpha \mapsto \begin{cases}\left(\begin{smallmatrix}\alpha & 0\\ 0 & 0\end{smallmatrix}\right) & \text{if } \vec{m} = \vec{[\,]} \text{ and } b = \mathtt{ff},\\ \left(\begin{smallmatrix}0 & 0\\ 0 & \alpha\end{smallmatrix}\right) & \text{if } \vec{m} = \vec{[\,]} \text{ and } b = \mathtt{tt},\\ \mathbf{0} & \text{otherwise.}\end{cases}$$

$$[\![U]\!]^{!\Delta\vdash\mathbf{qubit}^{\otimes n}\multimap\mathbf{qubit}^{\otimes n}}_{\vec{m},(\vec{*},\vec{*})} = M \mapsto \begin{cases}UMU^{-1} & \text{if } \vec{m} = \vec{[\,]},\\ \mathbf{0} & \text{otherwise.}\end{cases}$$

Table 4: Interpretation of the quantum constants. The writing $\vec{m}$ stands for a sequence of multisets in $|[\![!\Delta]\!]|$, the equality $\vec{m} = \vec{[\,]}$ meaning that each of these multisets is empty. $U$ and $M$ have the same dimension $\mathbb{C}^{2^n\times 2^n}$, $U$ being unitary.

note in particular that the permutation groups play a role only when interpreting !-formulas.

Let $\Gamma = x_1{:}A_1, \ldots, x_n{:}A_n$. The denotation of a typing judgement $\Gamma \vdash M : A$ is a morphism $[\![M]\!]^{\Gamma\vdash A} : [\![A_1 \otimes \cdots \otimes A_n]\!] \to [\![A]\!]$. The definition is by structural induction on the unique type derivation $\pi$ of $\Gamma \vdash M : A$ (see Proposition 4). The denotations of the constants $\mathtt{meas}$, $\mathtt{new}$ and the unitary transformations are given in Table 4. Table 5 briefly recalls the denotation of the usual linear logic rules. Here, the morphisms $\phi, \psi, \phi_A, \phi_B$ refer to the denotation of the premises of the last rule of $\pi$, which are uniquely defined given $\Gamma \vdash M : A$.

In the interpretation of the $\mathtt{letrec}$ constructor, the fixed point operator $\mathrm{Y}$ is defined as follows. Let $\phi$ be a morphism in the set $\overline{\mathbf{CPMs}}^{\oplus}(!C \otimes !A, !A)$. By induction on $n$, we define the morphism $\phi^n \in \overline{\mathbf{CPMs}}^{\oplus}(!C, !A)$: $\phi^0 := !C \xrightarrow{\mathtt{w};!\mathbf{0}} !A$, $\phi^{n+1} := !C \xrightarrow{\mathtt{c}} !C \otimes !C \xrightarrow{\mathrm{id}\otimes\phi^n} !C \otimes !A \xrightarrow{\phi} !A$. Since $\phi$ can be regarded as a continuous module homomorphism (in particular it is monotone), the set $\{\phi^n\}$ is directed complete. We define $\mathrm{Y}(\phi)$ as its least upper bound.

#### 4.3.1 Biproduct ($\mathfrak{A} \oplus \mathfrak{B}$)

Let $I$ be a (possibly infinite) set of indexes. The biproduct $\bigoplus_{i\in I}\mathfrak{A}_i$ of a family $\{\mathfrak{A}_i\}_{i\in I}$ of objects in $\overline{\mathbf{CPMs}}^{\oplus}$ is defined by

$$|\bigoplus_{i\in I}\mathfrak{A}_i| := \bigcup_{i\in I}\{i\} \times |\mathfrak{A}_i|, \quad d^{\bigoplus_{i\in I}\mathfrak{A}_i}_{(j,a)} := d^{\mathfrak{A}_j}_a, \quad G^{\bigoplus_{i\in I}\mathfrak{A}_i}_{(j,a)} := G^{\mathfrak{A}_j}_a.$$

The corresponding projections and injections are denoted respectively by $\pi^j$ and $\iota^j$ and defined as:

$$\pi^j_{(i,a),a'} := \iota^j_{a',(i,a)} := \delta_{j,i}\delta_{a,a'}G^{\mathfrak{A}_i}_a.$$

The tupling $\langle\phi_i\rangle_{i\in I}$ (resp. (co)-tupling $[\psi_i]_{i\in I}$) of a family of morphisms $\phi_i$ elements of $\overline{\mathbf{CPMs}}^{\oplus}(\mathfrak{A}, \mathfrak{B}_i)$ (resp. $\psi_i$ elements of $\overline{\mathbf{CPMs}}^{\oplus}(\mathfrak{A}_i, \mathfrak{B})$) is defined by $(\langle\phi_i\rangle_{i\in I})_{a,(j,b)} := (\phi_j)_{a,b}$ (resp. $([\psi_i]_{i\in I})_{(j,a),b} := (\psi_j)_{a,b}$).

**Example 17.** Recall that in Notation 3, the type $\mathbf{bit}$ is interpreted as the biproduct $[\![1]\!] \oplus [\![1]\!]$, which is the two-element family $\{(1, \{\mathrm{id}\})_{\mathtt{tt}}, (1, \{\mathrm{id}\})_{\mathtt{ff}}\}$. The positive cones associated with 1 and $\mathbf{bit}$ are: $\mathrm{Pos}([\![1]\!]) = \overline{\mathbb{R}^+}$ and $\mathrm{Pos}([\![\mathbf{bit}]\!]) = \overline{\mathbb{R}^+}^2$.

The typing judgement $\vdash \mathtt{tt} : \mathbf{bit}$ is interpreted as the right injection, which can be seen both as a family of two completely positive maps from $\mathbb{C}$ to $\mathbb{C}$ (i.e., $[\![\mathtt{tt}]\!]^{\vdash\mathbf{bit}}_{\star,\mathtt{tt}} = p \mapsto p$ and $[\![\mathtt{tt}]\!]^{\vdash\mathbf{bit}}_{\star,\mathtt{ff}} = p \mapsto 0$) and as a quantum compatible and completely positive map sending $p \in \overline{\mathbb{R}^+}$ to $(0, p) \in \overline{\mathbb{R}^+}^2$. Symmetrically, $[\![\mathtt{ff}]\!]^{\vdash\mathbf{bit}}$ is the map $p \mapsto (p, 0)$.

As an example of a term with free variables, consider $\texttt{Neg}_x := \texttt{if}\ x\ \texttt{then}\ \texttt{ff}\ \texttt{else}\ \texttt{tt}$. The denotation of $x : \mathbf{bit} \vdash \texttt{Neg}_x : \mathbf{bit}$ can be seen both as a family of four constant maps $[\![\texttt{Neg}_x]\!]^{\mathbf{bit}\vdash\mathbf{bit}}_{b,b'}$ from $\mathbb{C}$ to $\mathbb{C}$ of value 1 if $b \neq b'$ and 0 otherwise, and as a single map from $\overline{\mathbb{R}^+}^2$ to $\overline{\mathbb{R}^+}^2$ sending $(p, p')$ to $(p', p)$.

### 4.3.2 Symmetric monoidal structure ($\mathfrak{A} \otimes \mathfrak{B}$, $\mathbf{1}$ and $\mathfrak{A}^\ell$)

The bifunctor $\otimes : \overline{\mathbf{CPMs}}^\oplus \times \overline{\mathbf{CPMs}}^\oplus \to \overline{\mathbf{CPMs}}^\oplus$ is defined on objects $\mathfrak{A}, \mathfrak{B}$ by:

$$|\mathfrak{A} \otimes \mathfrak{B}| := |\mathfrak{A}| \times |\mathfrak{B}|, \qquad d^{\mathfrak{A}\otimes\mathfrak{B}}_{(a,b)} := d^{\mathfrak{A}}_a \times d^{\mathfrak{B}}_b,$$

$$G^{\mathfrak{A}\otimes\mathfrak{B}}_{(a,b)} := \{(g, h)\ ;\ g \in G^{\mathfrak{A}}_a, h \in G^{\mathfrak{B}}_b\},$$

where $d^{\mathfrak{A}}_a \times d^{\mathfrak{B}}_b$ is the multiplication of the two numbers $d^{\mathfrak{A}}_a$ and $d^{\mathfrak{B}}_b$, which can be seen as the lexicographically ordered set of pairs $(i, j)$, for $i < d^{\mathfrak{A}}_a$, $j < d^{\mathfrak{B}}_b$. Hence, the action of a permutation $(g, h) \in G^{\mathfrak{A}\otimes\mathfrak{B}}_{(a,b)}$ on $d^{\mathfrak{A}}_a \times d^{\mathfrak{B}}_b$ can be described as $(i, j) \mapsto (g(i), h(j))$.

The bifunctor $\otimes$ on morphisms is defined componentwise, using the standard tensor of the category **CPM** extended to the infinite elements by the universal property of the D-completion (Section 4.1). The tensor unit is the object $[\![1]\!]$ interpreting the unit type.

The associativity, unit, and symmetry isomorphisms are defined componentwise from the corresponding isomorphisms in **CPM**, composed with the actions of the groups of the objects. E.g., the symmetry is $\sigma^{\mathfrak{A},\mathfrak{B}}_{(a,b),(b',a')} := \delta_{a,a'}\delta_{b,b'} G^{\mathfrak{A}\otimes\mathfrak{B}}_{(a,b)}\ ;\ \sigma^{d^{\mathfrak{A}}_a, d^{\mathfrak{B}}_b}$, where $\sigma^{d^{\mathfrak{A}}_a, d^{\mathfrak{B}}_b}$ is the symmetry in **CPM** between $\mathbb{C}^{d^{\mathfrak{A}}_a \times d^{\mathfrak{A}}_a} \otimes \mathbb{C}^{d^{\mathfrak{B}}_b \times d^{\mathfrak{B}}_b}$ and $\mathbb{C}^{d^{\mathfrak{B}}_b \times d^{\mathfrak{B}}_b} \otimes \mathbb{C}^{d^{\mathfrak{A}}_a \times d^{\mathfrak{A}}_a}$. Notice that it is sufficient to pre-compose $\sigma^{d^{\mathfrak{A}}_a, d^{\mathfrak{B}}_b}$ with $G^{\mathfrak{A}\otimes\mathfrak{B}}_{(a,b)}$ (or, symmetrically, post-compose with $G^{\mathfrak{B}\otimes\mathfrak{A}}_{(b,a)}$), in order to have a map invariant under both the permutation groups $G^{\mathfrak{A}\otimes\mathfrak{B}}_{(a,b)}$ and $G^{\mathfrak{B}\otimes\mathfrak{A}}_{(b,a)}$. This is because $G^{\mathfrak{A}\otimes\mathfrak{B}}_{(a,b)}\ ;\ \sigma^{d^{\mathfrak{A}}_a, d^{\mathfrak{B}}_b} = G^{\mathfrak{A}}_a \otimes G^{\mathfrak{B}}_b\ ;\ \sigma^{d^{\mathfrak{A}}_a, d^{\mathfrak{B}}_b} = \sigma^{d^{\mathfrak{A}}_a, d^{\mathfrak{B}}_b}\ ;\ G^{\mathfrak{B}}_b \otimes G^{\mathfrak{A}}_a = \sigma^{d^{\mathfrak{A}}_a, d^{\mathfrak{B}}_b}\ ;\ G^{\mathfrak{B}\otimes\mathfrak{A}}_{(b,a)}$. Similar simplifications will be done henceforth without explicitly mentioning it.

**Example 18.** The denotation of $\mathbf{qubit} \otimes \mathbf{qubit}$ is the singleton web family $\{(4, \{\mathrm{id}\})_\star\}$. This object is associated with the cone of positive matrices of dimension $4 \times 4$ plus the infinite elements needed to complete the Löwner order. The denotation of $\mathbf{bit} \otimes \mathbf{bit}$ instead has a web of cardinality 4, i.e., $\{(\texttt{ff}, \texttt{ff}), (\texttt{ff}, \texttt{tt}), (\texttt{tt}, \texttt{ff}), (\texttt{ff}, \texttt{ff})\}$, and, for each index $b \in |[\![\mathbf{bit} \otimes \mathbf{bit}]\!]|$, we have $d^{[\![\mathbf{bit}\otimes\mathbf{bit}]\!]}_b = 1$ and $G^{[\![\mathbf{bit}\otimes\mathbf{bit}]\!]}_b = \{\mathrm{id}\}$. This object is associated with the biproduct $\overline{\mathbb{R}^+} \oplus \overline{\mathbb{R}^+} \oplus \overline{\mathbb{R}^+} \oplus \overline{\mathbb{R}^+}$.

Notice that in the above example the tensor product distributes over the biproducts: $[\![\mathbf{bit} \otimes \mathbf{bit}]\!] = [\![(1 \oplus 1) \otimes (1 \oplus 1)]\!] = [\![1 \oplus 1 \oplus 1 \oplus 1]\!]$. This is true in general: the isomorphism between $\mathfrak{A} \otimes (\bigoplus_{i\in I} \mathfrak{B}_i)$ and $\bigoplus_{i\in I}(\mathfrak{A} \otimes \mathfrak{B}_i)$ is

$$\texttt{distr}_{(a,(i,b)),(i',(a',b'))} := \delta_{i,i'}\delta_{a,a'}\delta_{b,b'} G^{\mathfrak{A}\otimes\mathfrak{B}_i}_{(a,b)}.$$

This isomorphism allows us to define the list constructor as the infinite biproduct of tensor powers $\mathfrak{A}^\ell := \bigoplus_{n=0}^{\infty} \mathfrak{A}^{\otimes n}$. In fact, we have $\mathfrak{A}^\ell \simeq \mathbf{1} \oplus (\mathfrak{A} \otimes \mathfrak{A}^\ell)$.

**Example 19.** The denotation of the unit type list is: $|[\![\mathbf{1}^\ell]\!]| = \mathbb{N}$ and, for every $n \in \mathbb{N}$, $d^{[\![\mathbf{1}^\ell]\!]}_n = 1$, $G^{[\![\mathbf{1}^\ell]\!]}_n = \{\mathrm{id}\}$. This object can be associated with the module $\overline{\mathbb{R}^+}^{\mathbb{N}}$ and is suitable for denoting the numerals in unary notation. Indeed, writing $\underline{n}$ for the list $\texttt{skip} :: \ldots \texttt{skip} :: \texttt{nil}$ of length $n$, we have $[\![\underline{n}]\!]^{\vdash \mathbf{1}^\ell} = p \mapsto (\underbrace{0, \ldots, 0}_{n-1 \text{ times}}, p, 0, \ldots)$.

### 4.3.3 Compact closure ($\mathfrak{A}^\perp$, $\mathfrak{A} \multimap B$)

Dual objects coincide: we have $\mathfrak{A}^\perp := \mathfrak{A}$. The unit $\eta^{\mathfrak{A}} \in \overline{\mathbf{CPMs}}^\oplus(\mathbf{1}, \mathfrak{A}^\perp \otimes \mathfrak{A})$ and co-unit $\epsilon^{\mathfrak{A}} \in \overline{\mathbf{CPMs}}^\oplus(\mathfrak{A} \otimes \mathfrak{A}^\perp, \mathbf{1})$ are defined componentwise composing the unit and co-unit of **CPM** with the correspondent permutation group. Writing $E_{i,j}$ for the matrix that has 0 everywhere except 1 at $(i, j)$, we have:

$$\eta^{\mathfrak{A}}_{\star,(a,a')} := 1 \mapsto \sum_{i,j<d^{\mathfrak{A}}} G^{\mathfrak{A}}_a(E_{i,j}) \otimes G^{\mathfrak{A}}_a(E_{i,j})$$

$$\epsilon^{\mathfrak{A}}_{(a,a'),\star} := (E_{i,j} \otimes E_{i',j'}) \mapsto \sum_{g,g'\in G^{\mathfrak{A}}_a} \frac{1}{\# G^{\mathfrak{A}}_a} \delta_{g(i),g'(i')}\delta_{g(j),g'(j')}.$$

Compact closed categories are monoidal closed. Let us recall the monoidal closure structure, which is needed to model the abstraction and the application of the quantum lambda calculus. The internal hom object is defined as $\mathfrak{A} \multimap \mathfrak{B} := (\mathfrak{A}^\perp \otimes \mathfrak{B}) = \mathfrak{A} \otimes \mathfrak{B}$. The evaluation morphism $\mathrm{Eval}^{\mathfrak{A},\mathfrak{B}} : \overline{\mathbf{CPMs}}^\oplus((\mathfrak{A} \multimap \mathfrak{B}) \otimes \mathfrak{A}, \mathfrak{B})$ and the currying isomorphism $\Lambda(-)$ from $\overline{\mathbf{CPMs}}^\oplus(\mathfrak{C} \otimes \mathfrak{A}, \mathfrak{B})$ to $\overline{\mathbf{CPMs}}^\oplus(\mathfrak{C}, \mathfrak{A} \multimap \mathfrak{B})$ are,

$$\mathrm{Eval}^{\mathfrak{A},\mathfrak{B}} := \sigma;\alpha;(\epsilon\otimes\mathrm{id});\lambda,\ \ \Lambda(\phi) := \lambda^{-1};(\eta\otimes\mathrm{id});\alpha^{-1};(\mathrm{id}\otimes(\sigma;\phi)),$$

where $\alpha$, $\lambda$, and $\sigma$ are the associative, left unit and symmetric isomorphisms associated with $\otimes$.

**Example 20.** Let us consider the abstraction $\lambda x.\texttt{Neg}_x$ of the term $\texttt{Neg}_x$ discussed in Example 17. The denotation $[\![\lambda x.\texttt{Neg}_x]\!]^{\vdash\mathbf{bit}\multimap\mathbf{bit}}$ is obtained from $[\![\texttt{Neg}_x]\!]^{x:\mathbf{bit}\vdash\mathbf{bit}}$ just by shifting the matrix indexes: $[\![\lambda x.\texttt{Neg}_x]\!]^{\vdash\mathbf{bit}\multimap\mathbf{bit}}_{\star,(b,b')} = [\![\texttt{Neg}_x]\!]^{x:\mathbf{bit}\vdash\mathbf{bit}}_{b,b'}$. Looking at this matrix as a module homomorphism, the map $[\![\lambda x.\texttt{Neg}_x]\!]^{\vdash\mathbf{bit}\multimap\mathbf{bit}}$ is $p \mapsto (0, p, p, 0)$, which is a map from $\overline{\mathbb{R}^+}$ to $\overline{\mathbb{R}^+}_{(\texttt{ff},\texttt{ff})} \oplus \overline{\mathbb{R}^+}_{(\texttt{ff},\texttt{tt})} \oplus \overline{\mathbb{R}^+}_{(\texttt{tt},\texttt{ff})} \oplus \overline{\mathbb{R}^+}_{(\texttt{tt},\texttt{tt})}$, where we make explicit the correspondence between the web elements of $[\![\mathbf{bit} \multimap \mathbf{bit}]\!]$ and the components of the biproduct associated with.

Application corresponds basically to matrix multiplication. For example, $[\![(\lambda x.\texttt{Neg}_x)(\texttt{meas}\ y)]\!]^{y:\mathbf{qubit}\vdash\mathbf{bit}}_{\star,b}$ is the function defined as $\sum_{b'\in\{\texttt{tt},\texttt{ff}\}} [\![\lambda x.\texttt{Neg}_x]\!]^{\vdash\mathbf{bit}\multimap\mathbf{bit}}_{\star,(b',b)} [\![\texttt{meas}\ y]\!]^{y:\mathbf{qubit}\vdash\mathbf{bit}}_{\star,b'}$, which is sending $\left(\begin{smallmatrix}\alpha & \beta\\ \gamma & \delta\end{smallmatrix}\right)$ to $\delta$ if $b = \texttt{ff}$, $\alpha$ if $b = \texttt{tt}$, and 0 otherwise.

### 4.3.4 Free commutative comonoids ($\mathfrak{A}^{\odot k}$, $!\mathfrak{A}$)

Let us now focus on the crucial structure modeling the linear logic modality !. We first define the notion of $k$-th symmetric power of an object and then we show how the biproduct of all such symmetric powers yields an exponential structure.

**Notation 21.** Given a set $X$, a *multiset* $\mu$ over $X$ is a function $X \mapsto \mathbb{N}$. The *support* of $\mu$ is the set $|\mu| = \{a \mid \mu(a) \neq 0\} \subseteq X$, the *disjoint union* is $(\mu \uplus \nu)(a) = \mu(a) + \nu(a)$, and the *empty multiset* is the zero constant function. The *cardinality* of $\mu$ is $\sum_{a \in X} \mu(a) \in \mathbb{N} \cup \{\infty\}$. A multiset is finite if it has finite cardinality. $\mathcal{M}_k(X)$ (resp. $\mathcal{M}_f(X)$) is the set of the multisets over $X$ with cardinality $k$ (resp. finite). Finite multisets can be denoted by listing the occurrences of their elements between square brackets, i.e., $\mu = [a, a, b]$ is $\mu(a) = 2$, $\mu(b) = 1$ and zero on the other elements, and $[\,]$ is the empty multiset.

In a symmetric monoidal category, given a natural number $k$, the *$k$-th symmetric power* of an object $\mathfrak{A}$ is a pair $(\mathfrak{A}^{\odot k}, eq^{\mathfrak{A}^{\odot k}})$ of an object $\mathfrak{A}^{\odot k}$ and a morphism $eq^{\mathfrak{A}^{\odot k}}$ from $\mathfrak{A}^{\odot k}$ to $\mathfrak{A}^{\otimes k}$, which is an equalizer of the $k!$ symmetries of the $k$-ary tensor $\mathfrak{A}^{\otimes k}$. Such equalizers do not exist in general, but they do exist in $\overline{\mathbf{CPMs}}^{\oplus}$ and can be concretely represented using the multisets notation, as follows:

$$|\mathfrak{A}^{\odot k}| := \mathcal{M}_k(|\mathfrak{A}|), \quad d_\mu^{\mathfrak{A}^{\odot k}} := \prod_{a \in |\mu|} (d_a^{\mathfrak{A}})^{\mu(a)},$$

$$G_\mu^{\mathfrak{A}^{\odot k}} := \{(h_a, g_a^1, \ldots, g_a^{\mu(a)})_{a \in |\mu|} \ ; \ h_a \in S_{\mu(a)}, g_a^i \in G_a^{\mathfrak{A}}\},$$

where $(h_a, g_a^1, \ldots, g_a^{\mu(a)})_{a \in |\mu|}$ is a $|\mu|$-indexed family of sequences of permutations and $G_\mu^{\mathfrak{A}^{\odot k}}$ is a group (composition being defined componentwise) whose action on $\mathbb{C}^{d_\mu^{\mathfrak{A}^{\odot k}} \times d_\mu^{\mathfrak{A}^{\odot k}}}$ can be described by seeing $d_\mu^{\mathfrak{A}^{\odot k}}$ as the set of families of sequences of the form $(i_a^1, \ldots, i_a^{\mu(a)})_{a \in |\mu|}$, with $i_a^j < d_a^{\mathfrak{A}}$ for every $j \leqslant \mu(a)$. Then, the action of $(h_a, g_a^1, \ldots, g_a^{\mu(a)})_{a \in |\mu|}$ on such families is:

$$(i_a^1, \ldots, i_a^{\mu(a)})_{a \in |\mu|} \mapsto (g_a^1(i_a^{h_a(1)}), \ldots, g_a^{\mu(a)}(i_a^{h_a(\mu(a))}))_{a \in |\mu|}.$$

The morphism $eq^{\mathfrak{A}^{\odot k}}$ is given by

$$eq^{\mathfrak{A}^{\odot k}}{}_{\mu,(a_1,\ldots,a_k)} := \begin{cases} G_\mu^{\mathfrak{A}^{\odot k}} & \text{if } \mu = [a_1, \ldots, a_k], \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

**Remark 22.** The object $[\![A]\!]^{\odot k}$ describes $k$ *unordered* uses of an element of type $A$. The fact that our model uses the symmetric tensor power $\mathfrak{A}^{\odot k}$ instead of the $k$-fold tensor $\mathfrak{A}^{\otimes k}$ means operationally that the behavior of a program calling its input $k$ times does not depend on the order of the calls.

**Example 23.** In Example 18, we have seen that $[\![\mathbf{qubit}]\!]^{\otimes 2} = \{(4, \{\mathrm{id}\})_\star\}$. The symmetric 2-power $[\![\mathbf{qubit}]\!]^{\odot 2}$ is instead the singleton web family $\{(4, \{\mathrm{id}, \sigma\})_\star\}$, where $4$ is represented as the lexicographically ordered set $\{(0,0),(0,1),(1,0),(1,1)\}$ and the permutation $\sigma$ acts on it by $(b, b') \mapsto (b', b)$. The group of permutations $\{\mathrm{id}, \sigma\}$ shrinks the set of possible morphisms to or from $[\![\mathbf{qubit}]\!]^{\odot 2}$. For example, the matrix $N_c$ associated with the controlled-not gate (Equation (1)) defines a complete positive endo-map of $\mathbb{C}^{4\times 4}$, which is an endo-morphism of $[\![\mathbf{qubit}]\!]^{\otimes 2}$ but not of $[\![\mathbf{qubit}]\!]^{\odot 2}$, because $N_c$ is not invariant under the action of $\{\mathrm{id}, \sigma\}$:

$$\{\mathrm{id}, \sigma\}(N_c) = \frac{1}{2}(\mathrm{id}(N_c) + \sigma(N_c)) = \frac{1}{2}\begin{pmatrix} 2&0&0&0\\ 0&1&0&1\\ 0&0&1&1\\ 0&1&1&0 \end{pmatrix} \neq N_c.$$

Concerning the module associated with symmetric tensor powers, $\mathrm{Pos}([\![\mathbf{qubit}]\!]^{\odot 2})$ is the D-completion of

$$\left\{ \begin{pmatrix} \alpha_1&\alpha_2&\alpha_2&\alpha_3\\ \alpha_4&\alpha_5&\alpha_6&\alpha_7\\ \alpha_4&\alpha_6&\alpha_5&\alpha_7\\ \alpha_8&\alpha_9&\alpha_9&\alpha_{10} \end{pmatrix} \text{ positive } ; \ \forall i, \alpha_i \in \mathbb{C} \right\}$$

which is a subcone of the positive cone of $\mathbb{C}^{4\times 4}$ of dimension $10$.

Concerning biproducts, the denotation of $\mathbf{qubit} \oplus \mathbf{qubit}$ is given by $\{(2,\{\mathrm{id}\})_{\mathtt{tt}}, (2,\{\mathrm{id}\})_{\mathtt{ff}}\}$, while its symmetric tensor power $[\![\mathbf{qubit} \oplus \mathbf{qubit}]\!]^{\odot 2}$ is given by the three-element family $\{(4,\{\mathrm{id},\sigma\})_{[\mathtt{tt},\mathtt{tt}]}, (4,\{\mathrm{id}\})_{[\mathtt{tt},\mathtt{ff}]}, (4,\{\mathrm{id},\sigma\})_{[\mathtt{ff},\mathtt{ff}]}\}$. Notice the difference between the pair $(4,\{\mathrm{id}\})$ associated with $[\mathtt{tt},\mathtt{ff}]$ and the pair $(4,\{\mathrm{id},\sigma\})$ associated with the two multisets of singleton support.

The biproduct $!\mathfrak{A} := \bigoplus_{k=0}^{\infty} \mathfrak{A}^{\odot k}$ of all symmetric powers of $\mathfrak{A}$ can be defined as

$$|!\mathfrak{A}| = \mathcal{M}_f(|\mathfrak{A}|), \quad d_\mu^{!\mathfrak{A}} = d_\mu^{\mathfrak{A}^{\odot k}}, \quad G_\mu^{!\mathfrak{A}} = G_\mu^{\mathfrak{A}^{\odot k}} \ (\mu \in \mathcal{M}_k(|\mathfrak{A}|))$$

This object yields a concrete representation of the free commutative comonoid generated by $\mathfrak{A}$. The counit (also called *weakening*) $\mathtt{w} \in \overline{\mathbf{CPMs}}^{\oplus}(!\mathfrak{A}, \mathbf{1})$ and the comultiplication (or *contraction*) $\mathtt{c} \in \overline{\mathbf{CPMs}}^{\oplus}(!\mathfrak{A}, !\mathfrak{A} \otimes !\mathfrak{A})$ are:

$$\mathtt{w}_{\mu,\star} := \delta_{\mu,[]} G_{[]}^{!\mathfrak{A}}, \qquad \mathtt{c}_{\mu,(\mu',\mu'')} := \delta_{\mu,\mu'+\mu''} G_\mu^{!\mathfrak{A}}.$$

The freeness of the comonoid gives the structure of exponential comonad. The functorial promotion maps an object $\mathfrak{A}$ to $!\mathfrak{A}$ and a morphism $\phi \in \overline{\mathbf{CPMs}}^{\oplus}(\mathfrak{A}, \mathfrak{B})$ to $!\phi \in \overline{\mathbf{CPMs}}^{\oplus}(!\mathfrak{A}, !\mathfrak{B})$ defined by, for $\mu \in \mathcal{M}_f(|\mathfrak{A}|)$ and $\nu = [b_1, \ldots, b_k] \in \mathcal{M}_f(|\mathfrak{B}|)$,

$$!\phi_{\mu,\nu} := \sum_{\substack{(a_1,\ldots,a_k), \text{ st} \\ [a_1,\ldots,a_k]=\mu}} G_\mu^{!\mathfrak{A}} \ ; \bigotimes_{i=1}^{k} \phi_{a_i,b_i} \ ; G_\nu^{!\mathfrak{B}}.$$

The counit of the comonad (or *dereliction*) $\mathtt{d} \in \overline{\mathbf{CPMs}}^{\oplus}(!\mathfrak{A}, \mathfrak{A})$ and the comultiplication (or *digging*) $\mathtt{dig} \in \overline{\mathbf{CPMs}}^{\oplus}(!\mathfrak{A}, !!\mathfrak{A})$ are

$$\mathtt{d}_{\mu,a} := \delta_{\mu,[a]} G_a^{\mathfrak{A}}, \qquad \mathtt{dig}_{\mu,M} := \delta_{\mu,\sum M} G_\mu^{!\mathfrak{A}},$$

where $M \in |!!\mathfrak{A}|$ is a multiset of multisets $\nu$ over $|\mathfrak{A}|$ and $\sum M \in |!\mathfrak{A}|$ is the multiset union of such $\nu$'s, i.e., for every $a \in |\mathfrak{A}|$, $\sum M(a) = \sum_{\nu \in |M|} \nu(a)^{M(\nu)}$.

Finally, the last two morphisms that are essential to interpret our calculus are Bierman's $\mathtt{m}^{\otimes} \in \overline{\mathbf{CPMs}}^{\oplus}(!\mathfrak{A} \otimes !\mathfrak{B}, !(\mathfrak{A} \otimes \mathfrak{B}))$ and $\mathtt{m}^{\mathbf{1}} \in \overline{\mathbf{CPMs}}^{\oplus}(\mathbf{1}, !\mathbf{1})$, given by $\mathtt{m}^{\otimes}_{(\mu,\nu),\eta} := \delta_{\eta,\mu\times\nu} G_\eta^{!(\mathfrak{A}\otimes\mathfrak{B})}$ and $\mathtt{m}^{\mathbf{1}}_{\star,\mu} := \delta_{\mu,[\star]} G_\mu^{\mathbf{1}}$, where $\mu \times \nu$ is the multiset in $|!(\mathfrak{A} \otimes \mathfrak{B})|$ defined by, $\mu \times \nu(a, b) := \mu(a)\nu(b)$.

**Example 24.** Using the isomorphism between $\mathcal{M}_f(\{\star\})$ and the set $\mathbb{N}$, and between $\mathcal{M}_f(\{\mathtt{tt},\mathtt{ff}\})$ and $\mathbb{N} \times \mathbb{N}$, the free commutative comonoids associated with $[\![\mathbf{1}]\!]$ and $[\![\mathbf{bit}]\!]$ are $![\![\mathbf{1}]\!] = \{(1,\{\mathrm{id}\})_n\}_{n\in\mathbb{N}}$, and $![\![\mathbf{bit}]\!] = \{(1,\{\mathrm{id}\})_{(n,m)}\}_{n,m\in\mathbb{N}}$. In general, notice that all constructions of the Lafont category preserve the underlying pair $(1,\{\mathrm{id}\})$ and act only at the level of webs. For more involved examples, one should look for objects with larger dimension, like $[\![\mathbf{qubit}]\!]$. For example, $![\![\mathbf{qubit}]\!] = \{(2^n, S_n)_n\}_{n\in\mathbb{N}}$. Notice that $!\mathbf{1}$, $!\mathbf{bit}$ and $!\mathbf{qubit}$ are not allowed by our type grammar. In fact, $!\mathbf{qubit}$ is meaningless because of the no-cloning constraint on quantum bits. However, such spaces should exist in the model since they are isomorphic to the denotations of legal types, like $!(1 \multimap 1)$, $!(1 \multimap \mathbf{bit})$ and $!(1 \multimap \mathbf{qubit})$.

## 4.4 The soundness theorem

The soundness of $\overline{\mathbf{CPMs}}^{\oplus}$ with respect to the operational semantics given in Figure 3 is an easy consequence of the fact that the category gives a (dcpo-enriched) model of linear logic. In fact, the operational semantics is a trivial extension of a head-reduction strategy of linear logic cut-elimination.

**Proposition 25.** *The category $\overline{\mathbf{CPMs}}^{\oplus}$ is a dcpo-enriched compact closed Lafont category, hence $\overline{\mathbf{CPMs}}^{\oplus}$ is a model of linear logic.*

*Proof (Sketch).* This basically amounts to showing that $\overline{\mathbf{CPMs}}^{\oplus}$ is the result of a categorical construction applied to $\mathbf{CPM}_s$ which is known to give, under certain circumstances, a dcpo-enriched Lafont category and to preserve the compact closed structure of $\mathbf{CPM}_s$. This construction was sketched in [5] and detailed in [15, 11, 12]. It consists in moving: (i) from $\mathbf{CPM}_s$ to a category $\mathbf{CPMs}$ with symmetric tensors, which is actually a full sub-category of the Karoubi envelope of $\mathbf{CPM}_s$; (ii) to a dcpo-enriched category $\overline{\mathbf{CPMs}}$ using the D-completion defined in [23, 7]; and, finally, (iii) constructing the free biproduct completion $\overline{\mathbf{CPMs}}^{\oplus}$ of $\overline{\mathbf{CPMs}}$ and applying Equation (4). ☐

Given a linking $\ell = |y_1, \ldots, y_m\rangle$, we write $\ell \vdash M : A$ for the judgement $y_1 : \mathbf{qubit}, \ldots, y_m : \mathbf{qubit} \vdash M : A$.

**Proposition 26** (Invariance of the interpretation)**.** *Let $\ell$ be the linking $|y_1, \ldots, y_m\rangle$, and assume $\ell \vdash M : A$. If $M$ is not a value, then for all quantum states $q \in \mathbb{C}^{2^m}$,*

$$\llbracket M \rrbracket^{\ell \vdash A}(qq^*) = \sum_{[q,\ell,M] \xrightarrow{p} [q',\ell',N]} p \cdot \llbracket N \rrbracket^{\ell' \vdash A}(q'q'^*). \qquad (7)$$

*Proof.* By hypothesis, $[q, \ell, M]$ is a typable total closure, and so, by Proposition 11 and Lemma 13, all of its reducts $[q', \ell', N]$ are typable total closures, so that $\llbracket N \rrbracket^{\ell' \vdash A}(q'q'^*)$ is well-defined.

Equation 7 is proven by cases, depending on the rule applied to $[q, \ell, M]$. The cases of Table 3(a) follows from the fact that $\overline{\mathbf{CPMs}}^{\oplus}$ is a dcpo-enriched model of linear logic. The quantum rules (Table 3(b)) are trivial consequences of Table 4, and the congruence rules of Table 3(c) are done by induction on $M$, using the fact that the category $\overline{\mathbf{CPMs}}^{\oplus}$ is linear. ☐

**Corollary 27.** *We have $\llbracket M \rrbracket_*^{\vdash 1} \geqslant \mathrm{Halt}_{[|\,\rangle, |\,\rangle, M]}$.*

*Proof.* By induction on $n$ and using Proposition 7 we can show that $\llbracket M \rrbracket_*^{\ell \vdash 1}(qq^*)$ is greater or equal to $\sum_{[q',\ell',V]} \mathrm{Red}^n_{[\ell,q,M],[q',\ell',V]}$. Then $\llbracket M \rrbracket_*^{\ell \vdash 1}(qq^*) \geqslant \mathrm{Halt}_{[q,\ell,M]}$ follows by taking the limit as $n \to \infty$,and invoking the monotonicity of $\{\mathrm{Red}^n\}_n$. ☐

## 4.5 The denotations of qlist and teleport

**Example 28.** Recall the terms of Example 5. The web of $\llbracket \mathbf{qubit}^\ell \rrbracket$ is $\mathbb{N}$, while $\llbracket \mathbf{qubit}^\ell \rrbracket_n = (2^n, \{\mathrm{id}\})$. Note that $\mathrm{Pos}(\llbracket \mathbf{qubit}^\ell \rrbracket)$ is equivalent to the D-completion of $\bigoplus_n P(\mathbb{C}^{2^n \times 2^n})$ where the set $P(\mathbb{C}^{2^n \times 2^n})$ is the cone of $2^n \times 2^n$ positive matrices. The denotation of the term **qlist** is a morphism in $\overline{\mathbf{CPMs}}^{\oplus}(\mathbf{qubit}, \mathbf{qubit}^\ell)$, that is, a map sending a $2 \times 2$ positive matrix onto $\bigoplus_n P(\mathbb{C}^{2^n \times 2^n})$. The program **qlist** is defined using recursion: its semantics is the limit of the morphisms $f_n$ sending $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ to the infinite sequence $(\mathbf{0}, \frac{1}{2}e_1, \ldots, \frac{1}{2^n}e_n, \mathbf{0}, \mathbf{0}, \ldots)$ where $e_i$ is the $2^i \times 2^i$ positive matrix

$$\begin{pmatrix} a & 0 & \cdots & 0 & b \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ c & 0 & \cdots & 0 & d \end{pmatrix}.$$

This limit is the map sending $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ to the sequence of infinitely increasing matrices $(\mathbf{0}, \frac{1}{2}e_1, \ldots, \frac{1}{2^n}e_n, \ldots)$. Note that the first element of the sequence is $\mathbf{0}$, as the program **qlist** never return the empty list. Also note that all the positive matrices in the sequence represent *entangled states of arbitrary sizes*. Our semantics is the first one to be able to account for such a case: in [6], only fixed sizes were allowed for entangled states.

**Example 29.** We claim in the introduction that the model is expressive enough to describe entanglement at higher-order types. As we discuss in Example 6, the encoding of the quantum teleportation algorithm produces two entangled, mutually inverse functions: $f : \mathbf{qubit} \multimap \mathbf{bit} \otimes \mathbf{bit}$ and $g : \mathbf{bit} \otimes \mathbf{bit} \multimap \mathbf{qubit}$.

The term $(\mathbf{teleport}\,\mathtt{skip})$ of type $(\mathbf{qubit} \multimap \mathbf{bit} \otimes \mathbf{bit}) \otimes (\mathbf{bit} \otimes \mathbf{bit} \multimap \mathbf{qubit})$ is one instance of such a pair of functions. Its denotation is a finite sequence of 16 square matrices of size $4 \times 4$. Using a lexicographic convention, we can lay them out as in Fig. 6. Because of the convention, morally each row corresponds to an element of type $\mathbf{bit} \otimes \mathbf{bit} \multimap \mathbf{qubit}$ whereas each column corresponds to an element of type $\mathbf{qubit} \multimap \mathbf{bit} \otimes \mathbf{bit}$. Picking a row, i.e., a choice of two left-sided booleans, amounts to choosing the two booleans that will be passed to the function $g$. Picking a column, i.e., a choice of two right-sided booleans, amounts to deciding on the probabilistic result we get from the function $f$. The intersection of a column and a row is therefore the representation of a map $\mathbf{qubit} \multimap \mathbf{qubit}$. This map is a description of a possible path in the control flow of the algorithm.

The matrices on the diagonal correspond to a run of the algorithm as it was intended: applying $g$ to the result of $f$. Since they are supposed to be the identity on $\mathbf{qubit}$, we can therefore deduce that the matrices $A_{00,00}$, $A_{01,01}$, $A_{10,10}$ and $A_{11,11}$ are all equal to $\left(\begin{smallmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{smallmatrix}\right)$. Since this matrix cannot be written as the tensor of two $2 \times 2$ matrices, we conclude that the denotation $A$ of $(\mathbf{teleport}\,\mathtt{skip})$ is indeed entangled.

We can compute the other matrices $A_{xy,zt}$ using the same argument: in general, $A_{xy,zt}$ is a composition of $f$ and $g$, except that instead of applying $g$ to $(x, y)$, we apply it to $(z, t)$. We therefore get a function $\mathbf{qubit} \to \mathbf{qubit}$ constructed out of the $U_{\_\_}$ that might (if $xy = zt$) or might not be the identity. In general, the matrix $A_{xy,zt}$ is the denotation of the unitary $U_{zt}U^*_{xy}$. The denotation $A$ is given in full detail in Table 6.

**Remark 30.** Example 29 is a good illustration of what we claimed in the introduction: the model reflects the juxtaposition of quantum and classical structures, even at higher-order types. Here, the control-flow is handled by the biproduct structure, and the quantum part of the algorithm is split across the list of $4\times 4$ matrices.

# 5 Adequacy

In the following, we prove the adequacy of $\overline{\mathbf{CPMs}}^{\oplus}$ (Theorem 38). This amounts to achieving the converse inequality of Corollary 27. The proof uses a syntactic approach, following [6]. We introduce a bounded $\mathtt{letrec}^n$, which can be unfolded at most $n$ times. On the one hand, the language allowing only bounded `letrec` is strongly normalizing (Lemma 33), hence the adequacy for it can be easily achieved by induction on the longest reduction sequence of a term (Corollary 34). On the other hand, the unbounded `letrec` can be expressed as the supremum of its bounded approximants, both semantically (Lemma 36) and syntactically (Lemma 37). We then conclude the adequacy for the whole quantum lambda calculus by continuity.

**Definition 31.** Let us extend the grammar of terms (Table 1) by adding: (i) a new term $\Omega^A$; (ii) a family of new term constructs $\mathtt{letrec}^n\ f^{A\multimap B}\ x = M \ \mathtt{in}\ N$ indexed by natural numbers $n \geqslant 0$.

The typing rules for these new constructs are

$$\frac{}{!\Delta \vdash \Omega^A : A} \qquad \frac{!\Delta, f : !(A\multimap B), x : A \vdash M : B \quad !\Delta,\Gamma, f : !(A\multimap B) \vdash N : C}{!\Delta,\Gamma \vdash \mathtt{letrec}^n\ f^{A\multimap B}\ x = M\ \mathtt{in}\ N : C}$$

Their denotations are given, respectively, by the map $\mathbf{0}$ and the family of maps

$$!\Delta\otimes\Gamma \xrightarrow{\mathtt{c}} !\Delta\otimes\Gamma\otimes!\Delta \xrightarrow{\mathrm{id}\otimes(\mathtt{dig};\mathtt{m};!(\Lambda\phi))^n} !\Delta\otimes\Gamma\otimes!(A\multimap B) \xrightarrow{\psi} C,$$

where $\phi \in \overline{\mathbf{CPMs}}^{\oplus}(!\Delta\otimes !(A\multimap B)\otimes A, B)$ and $\psi \in \overline{\mathbf{CPMs}}^{\oplus}(!\Delta\otimes\Gamma\otimes !(A\multimap B), C)$ are the denotations of the premises and $(\mathtt{dig};\mathtt{m};!(\Lambda\phi))^n \in \overline{\mathbf{CPMs}}^{\oplus}(!\Delta, !(A\multimap B))$ is defined in a similar fashion as in Table 5.

The reduction rules are updated as follows.

$$[q,\ell,\mathtt{letrec}^0\ f^{A\multimap B}\ x = M\ \mathtt{in}\ N] \xrightarrow{1} [q,\ell,N\{(\lambda x^A.\Omega^B)/f\}]$$

$$[q,\ell,\mathtt{letrec}^{n+1}\ f^{A\multimap B}\ x = M\ \mathtt{in}\ N] \xrightarrow{1} [q,\ell,N\{(\lambda x^A.\mathtt{letrec}^n\ f^{A\multimap B}\ x = M\ \mathtt{in}\ M)/f\}].$$

The additions to the language do not modify the properties of the language: subject reduction (Proposition 11) and totality (Lemma 13) hold as they are stated, while type safety (Proposition 12) and soundness (Proposition 26) are satisfied, with the proviso of considering the set of normal forms to consist of the set of values *and* the set of terms containing $\Omega$ in evaluating position.

**Definition 32.** A term is called *finitary* when it does not contain any occurrence of the un-indexed `letrec` construct. It can however contain $\Omega$ and any of the indexed $\mathtt{letrec}^n$. We call a closure *finitary* when its term is finitary.

**Lemma 33** (Strong normalization)**.** *If $[q_1,\ell_1,M_1]$ is finitary and typable, then every reduction sequence of the form $[q_1,\ell_1,M_1] \xrightarrow{p_1} [q_2,\ell_2,M_2] \xrightarrow{p_2} [q_3,\ell_3,M_3] \xrightarrow{p_3} \cdots$ is finite.*

*Proof (Sketch).* We reduce the finitary quantum lambda calculus to a simply typed non-deterministic language without quantum states, for which a standard proof technique can be used. The terms of this language are the terms of the extended quantum lambda calculus, minus the `letrec` construct. The operational semantics is obtained from Table 3 and the rules for $\mathtt{letrec}^n$ by replacing closures with the respective terms and the rules of Table 3b by dummy reduction rules: like $U(\bullet\otimes\cdots\otimes\bullet) \to \bullet\otimes\cdots\otimes\bullet$, or $\mathtt{new}\ \mathtt{ff} \to \bullet$. The symbol $\bullet$ denotes a distinct term variable, which, by convention, it is never bound by an abstraction. Clearly, the strong normalization of this language implies that of the finitary quantum lambda calculus. □

**Corollary 34** (Finitary adequacy)**.** *Let $M$ be a closed finitary term of unit type. Then $[\![M]\!]_*^{\vdash 1} = \mathrm{Halt}_{[|\rangle,|\rangle,M]}$.*

*Proof (Sketch).* We prove that, for any total finitary quantum closure of unit type $[q,\ell,M]$ we have $[\![M]\!]^{\ell\vdash 1}(qq^*) = \mathrm{Halt}_{[q,\ell,M]}$. In fact, by Lemma 33, there exists $m\in\mathbb{N}$ such that $\mathrm{Halt}_{[q,\ell,M]} = \sum_{[q',\ell',V]} \mathrm{Red}^m_{[q,\ell,M],[q',\ell',V]}$. We conclude by induction on $m$. □

**Definition 35.** Let $\lhd$ be a relation between finitary terms and general terms defined as the smallest congruence relation on terms satisfying, for every $M \lhd M'$ and $N \lhd N'$:

$$N\{(\lambda x^A.\Omega^B)/f\} \lhd (\mathtt{letrec}\ f\,x = M'\ \mathtt{in}\ N'),$$
$$(\mathtt{letrec}^n\ f\,x = M\ \mathtt{in}\ N) \lhd (\mathtt{letrec}\ f\,x = M'\ \mathtt{in}\ N').$$

**Lemma 36.** *If $\Gamma\vdash M : A$, then $[\![M]\!]^{\Gamma\vdash A} = \bigvee_{\substack{M'\lhd M\\ M'\ \text{finitary}}} [\![M']\!]^{\Gamma\vdash A}$.* □

**Lemma 37.** *If $M\lhd M'$, then $\mathrm{Halt}_{[q,\ell,M]} \leqslant \mathrm{Halt}_{[q,\ell,M']}$.*

*Proof (Sketch).* By induction on $n$, one proves the inequality: $\sum_{[q',\ell',V]} \mathrm{Red}^n_{[q,\ell,M],[q',\ell',V]} \leqslant \sum_{[q',\ell',V]} \mathrm{Red}^n_{[q,\ell,M'],[q',\ell',V]}$, from which the statement follows trivially. □

**Theorem 38.** *Let $M$ be a program, i.e., a closed term of unit type. Then $[\![M]\!]_*^{\vdash 1} = \mathrm{Halt}_{[|\rangle,|\rangle,M]}$.*

*Proof.* By Corollary 27 we have $[\![M]\!]_*^{\vdash 1} \geqslant \mathrm{Halt}_{[|\rangle,|\rangle,M]}$. Conversely, by Lemma 36, $[\![M]\!]_*^{\vdash 1} = \bigvee_{M'\lhd M} [\![M']\!]_*^{\vdash 1}$, which is equal to $\bigvee_{M'\lhd M} \mathrm{Halt}_{[|\rangle,|\rangle,M']}$ by Corollary 34, which is less or equal to $\mathrm{Halt}_{[|\rangle,|\rangle,M]}$ by Lemma 37. □

# 6 Structure of the sets of representable elements

We conclude this paper with an analysis of some of the properties of the denotation of terms. Recall that a morphism in $\overline{\mathbf{CPMs}}^{\oplus}$ is an indexed family of either completely positive maps, or infinite elements added during D-completion. We show that (1) all types have a non-zero inhabitant; (2) provided that the term constant $U$ ranges over arbitrary unitary matrices, the representable elements

$$
A = \frac{1}{4}\left(
\begin{array}{llll}
A_{00,00} = \begin{pmatrix} 1&0&0&1\\ 0&0&0&0\\ 0&0&0&0\\ 1&0&0&1 \end{pmatrix}, &
A_{00,01} = \begin{pmatrix} 0&0&0&0\\ 0&1&1&0\\ 0&1&1&0\\ 0&0&0&0 \end{pmatrix}, &
A_{00,10} = \begin{pmatrix} 1&0&0&-1\\ 0&0&0&0\\ 0&0&0&0\\ -1&0&0&1 \end{pmatrix}, &
A_{00,11} = \begin{pmatrix} 0&0&0&0\\ 0&1&-1&0\\ 0&-1&1&0\\ 0&0&0&0 \end{pmatrix}, \\
A_{01,00} = \begin{pmatrix} 0&0&0&0\\ 0&1&1&0\\ 0&1&1&0\\ 0&0&0&0 \end{pmatrix}, &
A_{01,01} = \begin{pmatrix} 1&0&0&1\\ 0&0&0&0\\ 0&0&0&0\\ 1&0&0&1 \end{pmatrix}, &
A_{01,10} = \begin{pmatrix} 0&0&0&0\\ 0&1&-1&0\\ 0&-1&1&0\\ 0&0&0&0 \end{pmatrix}, &
A_{01,11} = \begin{pmatrix} 1&0&0&-1\\ 0&0&0&0\\ 0&0&0&0\\ -1&0&0&1 \end{pmatrix}, \\
A_{10,00} = \begin{pmatrix} 1&0&0&-1\\ 0&0&0&0\\ 0&0&0&0\\ -1&0&0&1 \end{pmatrix}, &
A_{10,01} = \begin{pmatrix} 0&0&0&0\\ 0&1&-1&0\\ 0&-1&1&0\\ 0&0&0&0 \end{pmatrix}, &
A_{10,10} = \begin{pmatrix} 1&0&0&1\\ 0&0&0&0\\ 0&0&0&0\\ 1&0&0&1 \end{pmatrix}, &
A_{10,11} = \begin{pmatrix} 0&0&0&0\\ 0&1&1&0\\ 0&1&1&0\\ 0&0&0&0 \end{pmatrix}, \\
A_{11,00} = \begin{pmatrix} 0&0&0&0\\ 0&1&-1&0\\ 0&-1&1&0\\ 0&0&0&0 \end{pmatrix}, &
A_{11,01} = \begin{pmatrix} 1&0&0&-1\\ 0&0&0&0\\ 0&0&0&0\\ -1&0&0&1 \end{pmatrix}, &
A_{11,10} = \begin{pmatrix} 0&0&0&0\\ 0&1&1&0\\ 0&1&1&0\\ 0&0&0&0 \end{pmatrix}, &
A_{11,11} = \begin{pmatrix} 1&0&0&1\\ 0&0&0&0\\ 0&0&0&0\\ 1&0&0&1 \end{pmatrix}
\end{array}
\right).
$$

Table 6: The denotation of the quantum teleportation algorithm.

$$
\begin{array}{ll}
\omega_{\mathbf{qubit}} & = \lambda\mathtt{skip}.\mathtt{new}\,\mathtt{ff} \\
\omega_{A\multimap B} & = \lambda\mathtt{skip}.\lambda x^{A}.(\overline{\omega}_A\, x);(\omega_B\,\mathtt{skip}) \\
\omega_{!(A\multimap B)} & = \lambda\mathtt{skip}.\lambda x^{A}.(\omega_{A\multimap B}\,\mathtt{skip})\, x \\
\omega_{1} & = \lambda\mathtt{skip}.\mathtt{skip} \\
\omega_{A\otimes B} & = \lambda\mathtt{skip}.(\omega_A\,\mathtt{skip}) \otimes (\omega_B\,\mathtt{skip}) \\
\omega_{A\oplus B} & = \lambda\mathtt{skip}.\mathtt{if}\ \mathbf{c}\ \mathtt{then}\ (\omega_A\,\mathtt{skip})\ \mathtt{else}\ (\omega_B\,\mathtt{skip}) \\
\omega_{A^\ell} & = \mu f\mathtt{skip}.\mathtt{if}\ \mathbf{c}\ \mathtt{then}\ (\mathtt{skip})\ \mathtt{else}\ (\omega_A\,\mathtt{skip}) :: (f\,\mathtt{skip})
\end{array}
$$

$$
\begin{array}{ll}
\overline{\omega}_{\mathbf{qubit}} & = \lambda x^{\mathbf{qubit}}.\mathtt{if}\ \mathtt{meas}\, x\ \mathtt{then}\ \mathtt{skip}\ \mathtt{else}\ \mathtt{skip} \\
\overline{\omega}_{A\multimap B} & = \lambda f^{A\multimap B}.\overline{\omega}_B\,(f\,(\omega_A\,\mathtt{skip})) \\
\overline{\omega}_{!(A\multimap B)} & = \mu g f^{!(A\multimap B)}.\mathtt{if}\ \mathbf{c}\ \mathtt{then}\ \mathtt{skip}\ \mathtt{else}\ (\overline{\omega}_{A\multimap B}\, f);(g\, f) \\
\overline{\omega}_{1} & = \lambda\mathtt{skip}.\mathtt{skip} \\
\overline{\omega}_{A\otimes B} & = \lambda x^{A\otimes B}.\mathtt{let}\ z_1 \otimes z_2 \ = \ x\ \mathtt{in}\ (\overline{\omega}_A\, z_1);(\overline{\omega}_B\, z_2) \\
\overline{\omega}_{A\oplus B} & = \lambda x^{A\oplus B}.\mathtt{match}\, x\ \mathtt{with}\ (z_1^A : \overline{\omega}_A\, z_1 \mid z_2^B : \overline{\omega}_B\, z_2) \\
\overline{\omega}_{A^\ell} & = \mu f x^{A^\ell}.\mathtt{match}\,\mathtt{split}\, x\ \mathtt{with} \\
& \quad \left( z_1^1 \ : \ z_1 \ \middle| \ z_2^{A\otimes A^\ell} \ : \ \mathtt{let}\ y_1 \otimes y_2 = z_2\ \mathtt{in}\ (\overline{\omega}_A\, y_1);(f\, y_2)\right)
\end{array}
$$

Table 7: Two mutually recursive families of terms

of a given homset form a convex set including $\mathbf{0}$; and (3) infinite elements are not part of any representable map.

We first need two auxiliary definitions.

**Definition 39.** We define two type-indexed families of terms $\overline{\omega}_A$ and $\omega_A$ by mutual induction in Table 7. The term $\mathbf{c}$ represents the fair coin toss $\mathtt{meas}\,(H\,(\mathtt{new}\,\mathtt{ff}))$ (recall Example 1) and the notation $\mu f x.M$ stands for $\mathtt{letrec}\ f\, x = M\ \mathtt{in}\ f$.

**Lemma 40.** *For all types $A$, we have $\vdash \omega_A : 1 \multimap A$ and $\vdash \overline{\omega}_A : A \multimap 1$. Moreover, the morphisms $[\![\omega_A]\!]^{\vdash 1\multimap A}$ and $[\![\overline{\omega}_A]\!]^{\vdash A\multimap 1}$, seen as indexed families, do not contain the zero map.* ☐

**Corollary 41.** *All types are inhabited by at least one closed value of non-null denotation.*

*Proof.* Immediate with Lemma 40: for a given type $A$, choose the term $(\omega_A\,\mathtt{skip})$. ☐

**Proposition 42.** *Given a type $A$ and a context $\Gamma$, the denotations $[\![M]\!]^{\Gamma\vdash A}$ of valid typing judgements $\Gamma \vdash M : A$ form a convex set including $\mathbf{0}$.*

*Proof.* Suppose that $\Gamma$ is $x_1 : A_1, \ldots, x_n : A_n$. A term $M$ mapping to $\mathbf{0}$ is $(\overline{\omega}_{A_1} x_1; \ldots; \overline{\omega}_{A_n} x_n; \mathbf{\Omega})$ where the term $\mathbf{\Omega}$ is a shortcut for $\mathtt{letrec}\ f\, x = f\, x\ \mathtt{in}\ f\ \mathtt{skip}$, of denotation $\mathbf{0}$.

Now, suppose that $f = [\![M_1]\!]^{\Gamma\vdash A}$ and $g = [\![M_2]\!]^{\Gamma\vdash A}$, and choose two non-negative real numbers $\rho_1$, $\rho_2$ such that $\rho_1 + \rho_2 = 1$. There exists an angle $\phi$ such that $(\cos\phi)^2 = \rho_1$ and that $(\sin\phi)^2 = \rho_2$. As the term constants $U$ range over arbitrary unitaries, the unitary matrix $V_\phi = \left(\begin{smallmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{smallmatrix}\right)$ is representable in the quantum lambda calculus. The term $\mathbf{c}' = \mathtt{meas}\,(V_\phi\,(\mathtt{new}\,\mathtt{ff}))$ has denotation $(\rho_1, \rho_2)$. We then conclude that the term $\mathtt{if}\ \mathbf{c}'\ \mathtt{then}\ M_1\ \mathtt{else}\ M_2$ has denotation $\rho_1 f + \rho_2 g$. ☐

**Proposition 43.** *If $\Gamma \vdash M : A$ is valid, then no infinite element is part of the denotation $[\![M]\!]^{\Gamma\vdash A}$ of $M$.*

*Proof.* Suppose that one of the infinite elements of the D-completion were to be found in the interpretation of $x_1 : A_1, \ldots, x_n : A_n \vdash M : A$. Then the closed term

$$(\lambda x_1 \ldots x_n.\overline{\omega}_A\, M)(\omega_{A_1}\mathtt{skip})\ldots(\omega_{A_n}\mathtt{skip})$$

of type $1$ has infinite denotation, contradicting Theorem 38. ☐

This last proposition indicates that infinite elements introduced during the D-completion are really an artifact only needed for the categorical construction. The representable elements in the model are only built out of families of completely positive maps.

# 7 Conclusion

We presented a higher-order lambda calculus for quantum computation featuring classical and quantum data, duplication, recursion, and an infinite parametric type for lists. We then answered a long-standing open question: the description of a model for the full quantum lambda calculus. The model we propose is a free construction based on the known model of completely positive maps, but nevertheless has a concrete presentation.

One thing that this model explains and illustrates is the distinction between the quantum and classical parts of the language. The quantum part is described by completely positive maps (finite dimension), whereas the classical control is given by the Lafont category (i.e., linear logic). The model demonstrates that the two "universes" work well together, but also – surprisingly – that they do not mix too much, even at higher order types (we always have an *infinite* list of *finite* dimensional CPMs). The control flow is completely handled by the biproduct completion, and not by the CPM structure. The adequacy result, moreover, validates that the model is a "good" representation of the language.

One should also note that the product and the coproduct coincide in our model. For example, the model has morphisms that correspond to a program returning true with probability $1$ and false with probability $1$. We would like to point out that our interpretation is not *surjective*. For example, there are also morphisms in the model corresponding to "probability 2". (Incidentally, adding terms with such behavior makes it possible to build a term whose denotation is $\infty$ – so the fact that this provably does not happen somehow captures the sanity of the model). Interpretations in denotational models are often not surjective. In fact, it is an open problem to give a non-syntactic characterization of the image of our interpretation. Similarly, the problem of full-abstraction is still open.

## References


[1] V. Danos and T. Ehrhard. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Inform. Comput.*, 2011.

[2] T. Ehrhard. Finiteness spaces. *MSCS*, 15(4):615–646, 2005.

[3] J.-Y. Girard. Linear logic. *Th. Comp. Sc.*, 50:1–102, 1987.

[4] J.-Y. Girard. Normal functors, power series and lambda-calculus. *Ann. Pure Appl. Logic*, 37(2):129–177, 1988.

[5] J.-Y. Girard. Coherent Banach spaces: a continuous denotational semantics. *Theoretical Computer Science*, 227:297, 1999.

[6] I. Hasuo and N. Hoshino. Semantics of higher-order quantum computation via geometry of interaction. In *Proceedings of LICS*, pages 237–246, 2011.

[7] K. Keimel and J. D. Lawson. D-completions and the *d*-topology. *Annals of Pure and Applied Logic*, 159(3):292 – 306, 2009.

[8] E. H. Knill. Conventions for quantum pseudocode. Technical Report LAUR-96-2724, Los Alamos National Laboratory, 1996.

[9] Y. Lafont. *Logiques, catégories et machines*. PhD thesis, Université Paris 7, 1988.

[10] U. D. Lago, A. Masini, and M. Zorzi. Confluence results for a quantum lambda calculus with measurements. *Electr. Notes Theor. Comput. Sci.*, 270(2):251–261, 2011.

[11] J. Laird, G. Manzonetto, and G. McCusker. Constructing differential categories and deconstructing categories of games. *Information and Computation*, 222:247–264, 2013.

[12] J. Laird, G. McCusker, G. Manzonetto, and M. Pagani. Weighted relational models of typed lambda-calculi. In *LICS'13*, 2013.

[13] O. Malherbe. *Categorical models of computation: partially traced categories and presheaf models of quantum computation*. PhD thesis, University of Ottawa, 2010.

[14] P.-A. Melliès. Categorical semantics of linear logic. *Panoramas et Synthèses*, 12, 2009.

[15] P.-A. Melliès, N. Tabareau, and C. Tasson. An explicit formula for the free exponential modality of linear logic. In *ICALP'09 (2)*, pages 247–260, 2009.

[16] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2002.

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

[18] P. Selinger. Towards a semantics for higher-order quantum computation. In *QPL'04*, TUCS General Publication No 33, pages 127–143, 2004.

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

[20] P. Selinger and B. Valiron. On a fully abstract model for a quantum linear functional language. In *QPL'06*, 2008.

[21] P. Selinger and B. Valiron. Quantum lambda calculus. In S. Gay and I. Mackie, editors, *Semantic Techniques in Quantum Computation*, chapter 9, pages 135–172. Cambridge University Press, 2009.

[22] B. Valiron. *Semantics for a higher-order functional programming language for quantum computation*. PhD thesis, University of Ottawa, 2008.

[23] D. Zhao and T. Fan. Dcpo-completion of posets. *Th. Comp. Sc.*, 411(22–24):2167–2173, 2010.