Quantum CPOs

Andre Kornell
Tulane University
Louisiana, USA
akornell@tulane.edu

Bert Lindenhovius
Tulane University
Louisiana, USA
alindenh@tulane.edu

Michael Mislove
Tulane University
Louisiana, USA
mislove@tulane.edu

We introduce the monoidal closed category **qCPO** of *quantum cpos*, whose objects are 'quantized' analogs of ω-complete partial orders (cpos). The category **qCPO** is enriched over the category **CPO** of cpos, and contains both **CPO**, and the opposite of the category **FdAlg** of finite-dimensional von Neumann algebras as monoidal subcategories. We use **qCPO** to construct a sound model for the quantum programming language Proto-Quipper-M (PQM) extended with term recursion, as well as a sound and computationally adequate model for the Linear/Non-Linear Fixpoint Calculus (LNL-FPC), which is both an extension of the Fixpoint Calculus (FPC) with linear types, and an extension of a circuit-free fragment of PQM that includes recursive types.

1 Introduction

This is a progress report on an ongoing project to develop semantic models for quantum programming languages, with a particular focus supporting term recursion and type recursion. We introduce the category **qCPO** of 'quantum cpos' and Scott continuous maps, which can be regarded as a 'quantized' version of **CPO**, the category of ω-complete partial orders (cpos) and Scott continuous maps. **CPO** is a crucial building block for denotational models of classical programming languages with recursive types [11]; similarly, we show denotational models for quantum programming languages with recursive types can be built using **qCPO**: we prove **qCPO** supports a sound and computationally adequate model of Proto-Quipper-M (PQM) [22] extended with term recursion, as well as sound model of the Linear/Non-Linear Fixpoint Calculus (LNL-FPC) [14] that is computationally adequate at non-linear types.

The models of high-level functional quantum programming languages such as PQM are based on *linear / nonlinear models*, which arise from the work of Benton [1, 3] on models of the linear lambda calculus. More precisely, a linear/non-linear model consists of a non-linear Cartesian closed category **C**, a symmetric monoidal closed linear category **L**, and a strong monoidal functor $\mathbf{C} \to \mathbf{L}$ that has a right adjoint:

$$\mathbf{C} \rightleftarrows \mathbf{L} \tag{1}$$

A result in [12] shows that the semantic models of PQM all are suitable linear/non-linear models, and results in [12] and [14] show how to add term recursion and type recursion, respectively, to such models.

Abstract LNL models are suitable for a wide range of circuit description languages beyond those aimed at quantum devices. Examples include nonlinear languages that also support linear types, such as recent work on languages supporting session types for concurrency [5]. There also have been efforts to construct more concrete models for quantum programming languages based on a linear category closer to quantum computation. The category **FdAlg** of finite-dimensional von Neumann algebras and unital $*$-homomorphisms is one such linear category: any quantum circuit is a sequence of unitary operators followed by a measurement, so **FdHilb** is an obvious example of the linear category **M** which is the basis for the model of quantum programming languages like PQM in [22]. But there is no adjunction between

Submitted to:
QPL 2020

FdHilb and **Set**, so one has to embed it into an appropriate linear category to form an LNL model for PQM. In fact, **FdHilb** embeds contravariantly in **qCPO**, which also is a **CPO** enriched category; this opens the door to using **qCPO** as a concrete model for quantum programming languages like PQM that also supports recursion.

The categories **qCPO** and $\mathbf{qCPO}_{\perp!}$ (the subcategory of **qCPO** of pointed objects and strict maps) are completely analogous to **CPO** and $\mathbf{CPO}_{\perp!}$ in the nonlinear setting. In fact, the diagram below consists of the left adjoints of a commuting square of symmetric monoidal adjunctions:

$$\begin{tikzcd}
\mathbf{CPO}_{\perp!} \arrow[r, "`(-)"] & \mathbf{qCPO}_{\perp!} \\
\mathbf{CPO} \arrow[u, "(-)_\perp"] \arrow[r, "`(-)"'] & \mathbf{qCPO} \arrow[u, "(-)_\perp"']
\end{tikzcd} \tag{2}$$

Moreover, **qCPO** is symmetric monoidal closed, and the horizontal left adjoints in the above diagram are both inclusion functors. What's more, the 'lifting' functor $(-)_\perp : \mathbf{qCPO} \to \mathbf{qCPO}_{\perp!}$ in the diagram restricts to the ordinary lifting functor $(-)_\perp : \mathbf{CPO} \to \mathbf{CPO}_{\perp!}$. Combining this lift with the inclusion $\mathbf{CPO} \to \mathbf{qCPO}$ gives us a linear/non-linear model:

$$\begin{tikzcd}
\mathbf{CPO} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{qCPO}_{\perp!} \arrow[l, bend left, "G"]
\end{tikzcd} \tag{3}$$

We now can state our main results, Theorems 6.2, 6.3 and 6.4 below. Following [7], we view a *type constructor* as a bifunctor on the category of types, and then Theorem 6.2 states that all type constructors on $\mathbf{qCPO}_{\perp!}$ are parametrically algebraically compact. It follows that $\mathbf{qCPO}_{\perp,!}$ supports recursive types defined by its type constructors. Theorem 6.3 then states that (3) is a sound model for LNL-FPC (and hence supports recursive types) that is computationally adequate at nonlinear types, while Theorem 6.4 says (3) also is a sound model for Proto-Quipper-M that contains $\mathbf{FdAlg}^{\mathrm{op}}$ as a monoidal subcategory of the linear category $\mathbf{qCPO}_{\perp!}$. Importantly, to our knowledge, (3) is the only known LNL model to satisfy all these properties.

1.1 Related work

Typical quantum programming languages such as Proto-Quipper-M and its relatives [22], the quantum lambda calculus [24], and QWire [18], all have models consisting of a linear/non-linear adjunction, i.e., a monoidal adjunction as given in (1), where the symmetric monoidal category **L** contains a suitable monoidal subcategory representing quantum circuits, such as the category **FdAlg** of finite-dimensional von Neumann algebras. Models for PQM were given first in [22] and then in [12]. A model of the quantum lambda calculus in terms of Lafont categories (which also are linear/non-linear models [15]) was given in [17]. Another model was given in [6]. Models for QWire are given in [20], and of its extension EWire [21], including descriptions of their relations to linear/non-linear models.

We discuss some of these models in more detail. The category **CPO** of ω-complete partial orders (cpos) and Scott continuous maps, and its subcategory $\mathbf{CPO}_{\perp!}$ of pointed cpos and strict maps play a fundamental role in the semantics of programming languages supporting recursion [8]. Using this approach, PQM was extended with term recursion in [12], and soundness was established for the linear/nonlinear presheaf model

$$\mathbf{CPO} \rightleftarrows [\mathbf{FdAlg}, \mathbf{CPO}_{\perp!}]. \tag{4}$$

The article [14] focused on extending PQM with recursive types, but considered only the circuit-free fragment of the language, dubbed LNL-FPC, because it extends FPC with linear types. That paper also includes a computational adequacy result at non-linear types for abstract models of LNL-FPC.

Because the tensor product of $[\mathbf{FdAlg}, \mathbf{CPO}_{\perp !}]$ fails to satisfy a crucial hypothesis, the computational adequacy proof in [14] does not apply to this presheaf model. In fact, the only concrete model that satisfies all the required conditions for adequacy in [14] is

$$\mathbf{CPO} \rightleftarrows \mathbf{CPO}_{\perp !}, \tag{5}$$

but this model does not support quantum circuits since **FdAlg** does not embed in $\mathbf{CPO}_{\perp !}$.

The model of the quantum lambda calculus given in [17] is sound, computationally adequate at the unit type, and supports term recursion. However, its construction is intricate, so an alternative model was proposed in [6]:

$$\mathbf{Set} \rightleftarrows \mathbf{WStar}_{\mathrm{NMIU}}^{\mathrm{op}} \rightleftarrows \mathbf{WStar}_{\mathrm{NCPSU}}^{\mathrm{op}}, \tag{6}$$

where $\mathbf{WStar}_{\mathrm{NMIU}}$ denotes the category of von Neumann algebras and normal $*$-homomorphisms, and $\mathbf{WStar}_{\mathrm{NCPSU}}$ denotes the category of von Neumann algebras and normal completely positive subunital maps. This model (6) is natural because it utilizes von Neumann algebras, which are commonly used in physics to model quantum systems. However, this model is enriched over **Set**, not **CPO**, and it can be shown that the comonad induced by the adjunction between **Set** and $\mathbf{WStar}_{\mathrm{NMIU}}^{\mathrm{op}}$ cannot be algebraically compact, which implies the standard proof relying on algebraically compact type constructors cannot be used to prove this model supports recursive types.

The only proof we know that an LNL model supports recursion requires the Cartesian closed category of the model to be **CPO**, and the linear category to be **CPO**-enriched. To apply this approach to model (6) requires replacing **Set** with **CPO**, but the functors in the model don't admit obvious extensions. Instead, our solution is to restrict the class of von Neumann algebras in the model, and to equip them with a generalized cpo structure, which we call a quantum cpo.

1.2 Overview of the Rest of the Paper

The remainder of the paper lays out the theory of quantum cpos, their relation to classical cpos, and the basic categorical results that are needed to use quantum cpos as semantic models. We begin with quantum sets [9], which describe combined classical/quantum systems that are discrete in the sense that every complete Boolean algebra of propositions is isomorphic to a power set. The results in [9] draw heavily on the work in [10] and [27].

We next consider quantum posets and the morphisms between them. Quantum sets can be endowed with the quantum analog of a *partial order* – a reflexive, antisymmetric transitive relation. This notion is derived from the notion of a binary relation between quantum sets, essentially the quantum relations of Weaver [27]. By formulating an appropriate quantum generalization of families of monotonically increasing sequences, we then are able to define quantum cpos as a subclass of quantum posets, and Scott continuous functions as a subclass of monotone functions between them.

The presentation then describes the monoidal closed categories **qCPO** and $\mathbf{qCPO}_{\perp !}$. We state all the properties that make these categories useful as models for Proto-Quipper-M, and likely for the semantics of arbitrary quantum programming languages as well. Finally, we conclude with a short discussion of intended future work.

Because of space limitations, most proofs are omitted.

2 Quantum Sets and Relations

We briefly review the essential notions of [9]. A quantum set is a collection of finite-dimensional Hilbert spaces, e.g., $\mathcal{X} = \{\mathbb{C}^2, \mathbb{C}^5\}$. We call these Hilbert spaces the atoms of $\mathcal{X}$. *Intuitively*, a quantum set $\mathcal{X}$ is not an ordinary set, so we prefer to write $X \in \mathrm{At}(\mathcal{X})$, rather than $X \in \mathcal{X}$, when X is an atom of $\mathcal{X}$. Thus, $\mathcal{X}$ is a quantum set, whereas $\mathrm{At}(\mathcal{X})$ is an ordinary set, that is, a set in the ordinary sense, although $\mathcal{X}$ and $\mathrm{At}(\mathcal{X})$ are formally equal. This choice of notation affects the meaning of our terminology and expressions; for example, we will see that $\ell^\infty(\mathcal{X})$ means something other than $\ell^\infty(\mathrm{At}(\mathcal{X}))$. This convention is convenient for navigating the dictionary of quantum generalizations.

Two canonical quantum sets are $\mathbf{0} = \emptyset$, and $\mathbf{1} = \{\mathbb{C}\}$. The Cartesian product of two quantum sets $\mathcal{X}$ and $\mathcal{Y}$ consists of tensor product Hilbert spaces: $\mathcal{X} \times \mathcal{Y} = \{X \otimes Y \,|\, X \in \mathrm{At}(\mathcal{X}), Y \in \mathrm{At}(\mathcal{Y})\}$. Each quantum set $\mathcal{X}$ has a dual $\mathcal{X}^* = \{X^* \,|\, X \in \mathrm{At}(\mathcal{X})\}$, which consists of dual Hilbert spaces.

A quantum set with a single atom is called *atomic*. A quantum binary relation from an quantum set $\{H\}$ to an atomic quantum set $\{K\}$ is just a space of operators from H to K, i.e., a subspace of $L(H,K)$. More generally, a quantum binary relation R from a quantum set $\mathcal{X}$ to a quantum set $\mathcal{Y}$ is a matrix of such subspaces; formally, R is a choice of subspaces $R(X,Y) \subseteq L(X,Y)$, for each $X \in \mathrm{At}(\mathcal{X})$, and each $Y \in \mathrm{At}(\mathcal{Y})$.

Quantum sets and binary relations form a dagger-compact category **qRel**. The composition of binary relations then amounts to matrix multiplication, and as a matrix, the identity binary relation I has subspaces $\mathbb{C} \cdot 1$ down the diagonal, and zero subspaces off the diagonal. The adjoint binary relation $R^\dagger$ from $\mathcal{Y}$ to $\mathcal{X}$ is defined by $R^\dagger(Y,X) = \{F^\dagger \mid F \in R(X,Y)\}$, for $X \in \mathrm{At}(\mathcal{X})$ and $Y \in \mathrm{At}(\mathcal{Y})$.

The operator spaces $L(X,Y)$ are Hilbert spaces, so the binary relations from $\mathcal{X}$ to $\mathcal{Y}$ form an orthomodular lattice [23]. The bottom element of this orthomodular lattice is $\perp_{\mathcal{X},\mathcal{Y}}$, defined by $\perp_{\mathcal{X},\mathcal{Y}}(X,Y) = 0$. The top element of this orthomodular lattice is $\top_{\mathcal{X},\mathcal{Y}}$, defined by $\top_{\mathcal{X},\mathcal{Y}}(X,Y) = L(X,Y)$. Two binary relations R and S are said to be *orthogonal*, synonymously *disjoint*, just in case they are orthogonal in each entry, i.e., just in case $\mathrm{Tr}_X(S(X,Y)^\dagger \cdot R(X,Y)) = 0$ for all $X \in \mathrm{At}(\mathcal{X})$ and $Y \in \mathrm{At}(\mathcal{Y})$.

Each ordinary set S is identified with a quantum set ‘S, obtained by replacing each element of S with a one-dimensional Hilbert space. The space of linear operators from one one-dimensional Hilbert space to another is itself one-dimensional, so a quantum binary relation from ‘S to ‘T is corresponds to an ordinary binary relation from S to T. Thus, we have an inclusion functor from the category of ordinary sets and ordinary binary relations, to the category of quantum sets and quantum binary relations. It preserves the obvious dagger-compact structure on the former category, and it is essentially surjective onto those quantum sets whose atoms are all one-dimensional, which we term *classical*.

A *function* $F \colon \mathcal{X} \to \mathcal{Y}$ is simply a binary relation F from $\mathcal{X}$ to $\mathcal{Y}$ satisfying $F \circ F^\dagger \leq I$ and $F^\dagger \circ F \geq I$; these are the same conditions that characterize which ordinary relations f are functions, if one interprets $F^\dagger$ as the converse f^{-1}. Together with the Cartesian product of quantum sets defined above, the category **qSet** of quantum sets and functions is monoidal closed. Then the inclusion functor from the category **Set** of ordinary sets and ordinary functions into **qSet** is strong monoidal, and thus **Set** and **qSet** form a linear/non-linear model [2,3] with **qSet** as the linear category.

The category of quantum sets and functions is also dual to a category **OpAlg** of operator algebras. Up to isomorphism, the objects of **OpAlg** are von Neumann algebras of the form $\bigoplus_{\alpha \in I} M_{n_\alpha}(\mathbb{C})$, and the morphisms of **OpAlg** are unital normal $*$-homomorphisms. The contravariant equivalence takes each quantum set $\mathcal{X}$ to the von Neumann algebra $\ell^\infty(\mathcal{X}) := \bigoplus_{X \in \mathrm{At}(\mathcal{X})} L(X)$.Each function $F \colon \mathcal{X} \to \mathcal{Y}$ corresponds to a unital normal $*$-homomorphism $F^\star \colon \ell^\infty(\mathcal{Y}) \to \ell^\infty(\mathcal{X})$.

Normal states on $\ell^\infty(\mathcal{X})$ are intuitively probability distributions on $\mathcal{X}$. A normal state μ on $\ell^\infty(\mathcal{X})$ can be pushed forward along a function F from $\mathcal{X}$ to $\mathcal{Y}$, using the contravariant equivalence of the previous paragraph. Explicitly, the pushforward of μ is the normal state $\mu \circ F^\star$ on $\ell^\infty(\mathcal{Y})$. If $\mathcal{X} = `S$ and $\mathcal{Y} = `T$, for sets S and T, then this corresponds to the usual pushforward of a probability distribution.

A normal state μ on $\ell^\infty(\mathcal{X})$ can also be described by an element $m \in \ell^1(\mathcal{X}) := \bigoplus_{X \in \mathrm{At}(\mathcal{X})} L(X)$, using the expected formula $\mu(a) = \mathrm{Tr}(a \cdot m)$, for $a \in \ell^\infty(\mathcal{X})$. This trace is defined by $\mathrm{Tr}(a) = \sum_{X \in \mathrm{At}(\mathcal{X})} \mathrm{Tr}(a(x))$, for $a \in \ell^\infty(\mathcal{X})$. The operator m is a choice of density matrices, but normalized so that their traces sum to 1. The pushforward of μ can be calculated from m using Kraus operators. The contribution of the density matrix of μ at the atom $X \in \mathrm{At}(\mathcal{X})$, to the density matrix of $\mu \circ F^\star$ at the atom $Y \in \mathrm{At}(\mathcal{Y})$ is given by the expression $\dim(Y) \cdot \sum_{v \in B} v \cdot m(X) \cdot v^\dagger$, where B is any basis for $F(X,Y)$. Indeed, $F(X,Y)$ is canonically a Hilbert space for the inner product $\langle f_1 | f_2 \rangle = \mathrm{Tr}(f_1 \cdot f_2^\dagger)$.

3 Modeling physical systems

Finitary physical types are naturally modelled by quantum sets. For example, the qubit is modelled by the quantum set $\mathcal{H}_2$, whose only atom is a two-dimensional Hilbert space. By contrast, the classical bit is modelled by the quantum set $`\{0,1\}$, which has two one-dimensional atoms. The 'memory' of an idealized quantum computer might consist of finitely many qubits, and finitely many bits, so we can model such a quantum computer as a composite physical system.

The Cartesian product of ordinary sets generalizes to a symmetric monoidal structure × on the categories **qRel** and **qSet**. We recall that the monoidal product of two quantum sets is defined by forming binary tensor products of their atoms. Composite systems consisting of two fully quantum systems, that is, of quantum systems each modelled by a single Hilbert space, are obtained by forming the tensor product of the two Hilbert spaces. Similarly, composite systems consisting of two fully classical systems, that is, of quantum systems modelled by ordinary sets, are obtained by forming the ordinary Cartesian product of the two sets. Thus, our generalized product models composite systems consisting entirely of fully quantum systems, or entirely of fully classical systems. In fact, it is appropriate for modelling mixed quantum/classical systems as well. In particular, an idealized quantum computer possessing n qubits and m bits can be modelled by the quantum set $\underbrace{\mathcal{H}_2 \times \cdots \times \mathcal{H}_2}_{n} \times \underbrace{`\{0,1\} \times \cdots \times `\{0,1\}}_{m}$. This quantum set has 2^m atoms, each of dimension 2^n.

Other natural type constructors are also easily modelled in quantum sets. Sum types are modelled by disjoint unions of quantum sets, portraying a kind of classical disjunction of potentially quantum systems. The resulting physical system may be in a pure state of the first type or in a pure state of the second type, but no superpositions may occur. For example, an idealized quantum computer possessing a single qubit and a single bit is modelled by a quantum set with two atoms, because the computer can be in a configuration where the bit has value 0, and it can be in a configuration where the bit has value 1, but it cannot be in a superposition between two such states.

Higher types are modelled by quantum function sets [9, Definition 9.2], the inner hom objects of the closed monoidal category **qSet**. These quantum sets are more challenging to describe, mainly due to the large automorphism group of the physical qubit. The one-dimensional atoms of the quantum function set $\mathcal{H}_2^{\mathcal{H}_2}$ are in canonical bijective correspondence with the automorphisms of $\mathcal{H}_2$ in **qSet**, i.e., with the $*$-automorphisms of $M_2(\mathbb{C})$. The quantum function set $\mathcal{H}_2^{\mathcal{H}_2}$ has higher-dimensional atoms as well. In general, the quantum function set from a quantum set $\mathcal{X}$ to a quantum set $\mathcal{Y}$ has an atom of dimension d for each unital normal $*$-homomorphism $\rho\colon \ell^\infty(\mathcal{Y}) \to \ell^\infty(\mathcal{X}) \overline{\otimes} L(H_d)$[1] that is irreducible in the sense that 0 and 1 are the only projections of the form $1 \otimes p$ that commute with the image of ρ.

[1] Here $\overline{\otimes}$ denotes the spatial tensor product of von Neumann algebras [25, Definition IV.1.3].

The bang operator ! applied to a quantum set extracts the quantum subset consisting of its one-dimensional atoms; this is essentially an ordinary set. This also is the largest subset of the original quantum set that admits a duplication map. Indeed, classical sets all admit duplication maps, but the no-cloning theorem forbids the duplication of higher dimensional atoms. Returning to the example of the previous paragraph, every atom of $!(\mathcal{H}_2^{\mathcal{H}_2})$ is one-dimensional, and furthermore $!(\mathcal{H}_2^{\mathcal{H}_2}) \cong$ ‘$\mathbf{qSet}(\mathcal{H}_2,\mathcal{H}_2)$.

Each quantum gate or measurement may be modelled by a function between quantum sets. Such a function is analogous to a function between the configuration spaces of two classical systems. A function between configuration spaces induces an map from the states on the domain system to the states on the codomain system. The function pushes probability measures on the domain system forward onto the codomain system. The standard formalizations of quantum gates and measurements as maps on state spaces are analogous to these pushforward maps. Examples 3.1 and 3.2 below exhibit the formalizations of the Hadamard gate and of qubit measurement respectively by functions between quantum sets. Each function induces a map on states, and in Example 3.3 below, we examine this map in the case of qubit measurement, recovering the expected probability distributions on experimental outcomes.

Example 3.1. Quantum gates are automorphisms of fully quantum systems, which consist of qubits. Such an automorphism is typically formalized by a unitary operator. In **qSet**, such an automorphism is formalized by a closely-related function. Indeed, each function $F_1\colon \mathcal{H}_d \to \mathcal{H}_d$, for any positive integer d, is defined by $F_1(H_d,H_d) = \mathbb{C}\cdot u$, for some unitary operator u. For example, the Hadamard gate is formalized by the function $F_1\colon \mathcal{H}_2 \to \mathcal{H}_2$ defined by

$$F_1(H_2,H_2) = \mathbb{C}\cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

This function F_1 is an automorphism of $\mathcal{H}_2$ in **qSet**. The unital normal $*$-homomorphism $M_2(\mathbb{C}) \cong \ell^\infty(\mathcal{H}_2) \to \ell^\infty(\mathcal{H}_2) \cong M_2(\mathbb{C})$ that corresponds to this function, in the sense of the duality between quantum sets and hereditarily atomic von Neumann algebras [9], is simply conjugation by the Hadamard matrix above, appropriately normalized.

Example 3.2. Measurement is a channel from a fully quantum system to a fully classical system; it may be formalized by a function from an atomic quantum set to a classical quantum set. The effect of this function on states yields probability distributions on experimental outcomes. In this example we exhibit the function, and in the next example, we examine its effect on states.

The definition of a function between quantum sets implies functions from an atomic quantum set $\mathcal{H}_d$ to a classical quantum set ‘S are in bijective correspondence with projection-valued measurements on $\mathcal{H}_d$. Explicitly, $F_2(H_d,\mathbb{C}_s) = L(H_d,\mathbb{C}_s)\cdot p_s$ for each $s \in S$, where $(p_s \,|\, s\in S)$ is a projective POVM.[2] In particular, the standard measurement of a qubit is formalized by a function $F_2\colon \mathcal{H}_2 \to$ ‘$\{1,-1\}$ defined by

$$F_2(H_2,\mathbb{C}_s) = \begin{cases} \mathbb{C}\cdot\begin{bmatrix}1 & 0\end{bmatrix} & s=1; \\ \mathbb{C}\cdot\begin{bmatrix}0 & 1\end{bmatrix} & s=-1. \end{cases}$$

This function F_2 is an epimorphism in **qSet**. Equivalently, it is surjective in the sense that $F_2\circ F_2^\dagger \geq I$. The unital normal $*$-homomorphism $\mathbb{C}^2 \cong \ell^\infty(\{1,-1\}) \to \ell^\infty(\mathcal{H}_2) \cong M_2(\mathbb{C})$ corresponding to this function, in the sense of the duality between quantum sets and hereditarily atomic von Neumann algebras [9], includes $\mathbb{C}^2$ into $M_2(\mathbb{C})$ diagonally.

[2] POVM stands for *Positive Operator-Valued Measurement.*

Example 3.3. A function between quantum sets is also a quantum channel in the familiar mathematical sense: normal states are pushed forward to normal states, as described in section 2, and the function is completely determined by this mapping. In the case of the measurement channel, $\{[1 \quad 0]\}$ is an orthonormal basis for $F_2(H_2,\mathbb{C}_1)$, and $\{[0 \quad 1]\}$ is an orthonormal basis for $F_2(H_2,\mathbb{C}_{-1})$. A state on $\ell^\infty(\mathcal{H}_2) = M_2(\mathbb{C})$ is a density matrix $m = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$, while a state on $\ell^\infty(`\{1,-1\}) \cong \mathbb{C}\oplus\mathbb{C}$ is a pair of positive numbers p_1 and p_{-1} that sum to 1. Pushing forward the density matrix m in the prescribed way, we find that

$$p_1 = \dim(\mathbb{C}_1)\cdot[1 \quad 0]\cdot\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}\cdot\begin{bmatrix}1\\0\end{bmatrix} = m_{11}, \quad \text{and} \quad p_2 = \dim(\mathbb{C}_{-1})\cdot[0 \quad 1]\cdot\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}\cdot\begin{bmatrix}0\\1\end{bmatrix} = m_{22}.$$

In this way we recover the probabilities of our two experimental outcomes from F_2.

4 Quantum posets

Definition 4.1. *A* quantum poset *is a pair $(\mathcal{X},R)$ consisting of a quantum set $\mathcal{X}$ and a relation $R \in$* $\mathbf{qRel}(\mathcal{X},\mathcal{X})$ *satisfying: (1) $I_\mathcal{X} \leq R$ (reflexivity); (2) $R\circ R \leq R$ (transitivity); and (3) $R\wedge R^\dagger \leq I_\mathcal{X}$ (anti-symmetry). (This definition of a quantum poset is essentially that of Weaver* [27, Definition 2.6].*)*

A monotone *map $F : (\mathcal{X},R) \to (\mathcal{Y},S)$ is simply a function $F : \mathcal{X} \to \mathcal{Y}$ satisfying $F\circ R \leq S\circ F$.*[3] *We denote the category of quantum posets with monotone maps by* **qPOS**.

Example 4.2. Let $\mathcal{X}$ be a quantum set. Then $I_\mathcal{X}$ is a quantum order on $\mathcal{X}$, which we call the *trivial* order.

If $(S,\sqsubseteq)$ is an ordinary poset, then $(`S, `\sqsubseteq)$ is a quantum poset, and vice versa; for example, the trivial order $\sqsubseteq$ on S corresponds to $`(\sqsubseteq) = I_{`S}$, the trivial order on $`S$. And, a monotone map $f$ between ordinary posets gives rise to a monotone function $`f$ between the associated quantum posets, and vice versa. It follows that $`(-)$ extends to a fully faithful functor **POS** → **qPOS**.

Example 4.3. A 'non-classical' quantum poset is given by the relation R on $\mathcal{H}_2$ specified by

$$R(H_2,H_2) = \mathbb{C}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} + \mathbb{C}\begin{pmatrix}0 & 1\\ 0 & 0\end{pmatrix}.$$

Since $\mathcal{H}_2$ has only one atom H_2, R is determined by $R(H_2,H_2)$. Then $(\mathcal{H}_2,R)$ is a quantum poset.

The partial orders that appear in recursion theory are often viewed as *information orderings*. For example, a partial function on the natural numbers can be viewed as one step in the construction of a total function, i.e., as a partially specified total function. In this sense, an extension of a given partial function carries more information about the total function that it purports to describe.

A preorder structure on the phase space of a classical system can be viewed as formalizing some entropic process undergone by the system. More precisely, whereas a stochastic matrix records the transition probabilities of the process, the preorder structure simply records which transitions are possible. A lower configuration in the order is the result of more transitions, and thus carries less information about the initial configuration of the system.

The same intuition is available in the quantum setting. A preorder structure on a quantum set $\mathcal{X}$ may be understood as an information order on that quantum set not only by analogy with the classical case,

[3] This is analogous to the condition $(f\times f)\circ(\leq_P) \subseteq (\leq_Q)\circ(f\times f)$ which characterizes when a map $f\colon (P,(\leq_P)) \to (Q,(\leq_Q))$ between ordinary posets is monotone.

but also directly. A stochastic channel on $\mathcal{H}_d$, formally a completely positive map of the appropriate kind, determines a preorder on $\mathcal{H}_d$, essentially the unital subalgebra of $L(H_d)$ generated by the adjoints of its Kraus operators (c.f. [28]). We work with the algebra generated by the adjoints because the information order is opposite to the transition order; later states carry less information.

Any partial order on a finite set can be encoded as the possible transitions of some stochastic matrix, but the same is not true in the quantum case. For instance, it is easy to show that the partial order on $\mathcal{H}_2$ defined in Example 4.3 does not arise in this way. This phenomenon is intuitively related to the fact that some partial functions between quantum sets cannot be extended to total functions. If we allow partial channels, which are permitted to consume their input without producing an output, then the partial order in 4.3 arises from the partial channel Φ:

$$\Phi(\rho) = \frac{1}{2}\rho + \frac{1}{2}v\rho v^* \qquad \text{where} \qquad v = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

In both the classical and quantum cases, it is appropriate to identify configurations that are equivalent in the information they carry when modelling recursion. In the classical case, we turn the preordered set into a partially ordered set by taking a quotient. This quotient construction has a quantum analog, which is most easily viewed using Weaver's characterization of binary relations between quantum sets [27].

Thus, quantum posets, and therefore also quantum cpos, can be regarded as quantum systems equipped with a stochastic channel that models a kind of decay, and therefore a kind of information order.

More elaborate examples of quantum posets can be described using the coproduct and the monoidal product on **qPOS**. The monoidal product $(\mathcal{X},R)\times(\mathcal{Y},S)$ of $(\mathcal{X},R)$ and $(\mathcal{Y},S)$ in **qPOS** is the quantum poset $(\mathcal{X}\times\mathcal{Y},R\times S)$, where $\mathcal{X}\times\mathcal{Y}$ is the monoidal product of objects in **qRel**, and $R\times S$ is the monoidal product of morphisms in **qRel**. The embedding $`(-):\mathbf{POS}\to\mathbf{qPOS}$ is strong monoidal, and has a right adjoint.

Similarly, the coproduct $(\mathcal{X},R)\uplus(\mathcal{Y},S)$ of $(\mathcal{X},R)$ and $(\mathcal{Y},S)$ is the quantum poset $(\mathcal{X}\uplus\mathcal{Y},R\uplus S)$, where $\mathcal{X}\uplus\mathcal{Y}$ is the coproduct of objects in **qRel**, as well as in **qSet**, and $R\uplus S$ is the coproduct of morphisms in **qRel**. The generalization to coproducts of arbitrary families is immediate.

The limit of a diagram $D:A\to\mathbf{qPOS}$ can be formed as follows: For each $\alpha\in A$, let $D(\alpha)=(\mathcal{X}_\alpha,R_\alpha)$. Let $\mathcal{X}$ be the limit of the $\mathcal{X}_\alpha$ in **qSet**, and let $J_\alpha:\mathcal{X}\to\mathcal{X}_\alpha$ be the limiting maps. Then $R=\bigwedge_{\alpha\in A}J_\alpha^\dagger\circ R_\alpha\circ J_\alpha$ is a partial order on $\mathcal{X}$, and $(\mathcal{X},R)$ is the limit of D in **qPOS**.

A partial order R on a quantum set $\mathcal{X}$ also imposes a partial order on the set of functions into $\mathcal{X}$ from any fixed quantum set $\mathcal{W}$: if F and G are functions $\mathcal{W}\to\mathcal{X}$, we define $F\sqsubseteq G$ iff $G\leq R\circ F$; this is completely analogous to the classical setting where $g\subseteq\leq\circ f$ expresses $g\leq f$.

Lemma 4.4. *Let $(\mathcal{X},R)$ be a quantum poset, and let $\mathcal{W}$ be any quantum set. Then set* $\mathbf{qSet}(\mathcal{W},\mathcal{X})$ *of functions from $\mathcal{W}$ to $\mathcal{X}$ is partially ordered by $F\sqsubseteq G$ iff $G\leq R\circ F$.*

It follows that $\mathbf{qPOS}(\mathcal{X},\mathcal{Y})$ inherits the order from its superset $\mathbf{qSet}(\mathcal{X},\mathcal{Y})$ for any quantum posets $\mathcal{X}$ and $\mathcal{Y}$. This order does not depend on the order on $\mathcal{X}$, which is also what happens classically.

Lemma 4.5. **qPOS** *is order enriched.*

The category **qPOS** has all the categorical properties of the category **POS**:

Theorem 4.6. *The category* **qPOS** *is complete, has all coproducts, and is monoidal closed. The functor* $`(-):\mathbf{POS}\to\mathbf{qPOS}$ *is strong monoidal and has a right adjoint.*

Since **qPOS** is monoidal closed, it is enriched over itself, so $\times$ is a **qPOS**-bifunctor. By Theorem 4.6, $`(-):\mathbf{POS}\to\mathbf{qPOS}$ is the left adjoint of an LNL model, and it follows that $\times$ is also a **POS**-functor. One also can show that the monoidal product $\times$ reflects the order on homsets, hence:

Proposition 4.7. *Let $(\mathcal{X}_1,R_1)$, $(\mathcal{X}_2,R_2)$, $(\mathcal{Y}_1,S_1)$ and $(\mathcal{Y}_2,S_2)$ be quantum posets, and let $F_1,G_1:\mathcal{X}_1\to\mathcal{Y}_1$ and $F_2,G_2:\mathcal{X}_2\to\mathcal{Y}_2$ be functions. Then both $F_1\sqsubseteq G_1$ and $F_2\sqsubseteq G_2$ if and only if $F_1\times F_2\sqsubseteq G_1\times G_2$.*

5 Quantum CPOs

An ω-CPO is an ordinary poset P satisfying the property that any increasing chain $x_1 \leq x_2 \leq \cdots \in P$ has a *least upper bound* $x_\infty \in P$ satisfying $x_\infty = \sup_n x_n$. It is well known that this is equivalent to the condition that, for any ordinary set W and any increasing sequence of functions $f_1 \leq f_2 \leq \cdots : W \to P$, there is a function $f_\infty \colon W \to P$ satisfying $\sup_n f_n(w) = f_\infty(w)$, $\forall w \in W$. This motivates the following definitions:

Definition 5.1. *Let $(\mathcal{X}, R)$ be a quantum poset, and let $\mathcal{W}$ be any quantum set. Let $K_1 \sqsubseteq K_2 \sqsubseteq \cdots : \mathcal{W} \to \mathcal{X}$ be a monotonically increasing sequence for the order on* $\mathbf{qSet}(\mathcal{W}, \mathcal{X})$ *defined above. If there exists a function $K_\infty : \mathcal{W} \to \mathcal{X}$ such that*

$$R \circ K_\infty = \bigwedge_{n \in \mathbb{N}} R \circ K_n,$$

then we say that K_∞ is the limit *of the sequence, and we write $K_n \nearrow K_\infty$.*

Thus, it would be natural to define a quantum cpo to be a quantum poset $\mathcal{X}$ that has a limit K_∞ for any increasing sequence $K_1 \sqsubseteq K_2 \sqsubseteq \cdots : \mathcal{W} \to \mathcal{X}$, from any quantum set $\mathcal{W}$. To tame this characterization, we recall $\mathcal{H}_n$ denotes the quantum set whose single atom is H_n, the n-dimensional Hilbert space. Together the quantum sets $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \ldots$ form a generating family [4, Definition 4.5.1] for **qSet**, leading to:

Definition 5.2. *(a) A quantum poset $(\mathcal{X}, R)$ is a* quantum cpo *iff for each $d \in \mathbb{N}$, and each increasing sequence $K_1 \sqsubseteq K_2 \sqsubseteq \cdots : \mathcal{H}_d \to \mathcal{X}$, there is a function $K_\infty \colon \mathcal{H}_d \to \mathcal{X}$ such that $K_n \nearrow K_\infty$.*

(b) A function $F \colon \mathcal{X} \to \mathcal{Y}$ between quantum cpos $(\mathcal{X}, R)$ and $(\mathcal{Y}, S)$ is Scott continuous *iff for each $d \in \mathbb{N}$, and all functions $K_i \colon \mathcal{H}_d \to \mathcal{X}$, with $i \in \mathbb{N} \cup \{\infty\}$, if $K_n \nearrow K_\infty$ then $F \circ K_n \nearrow F \circ K_\infty$.*

We write **qCPO** *for the category of quantum cpos and Scott continuous functions.*

As implied by the terminology, every ordinary cpo is also a quantum cpo, and conversely.

Theorem 5.3. *Let $(S, \sqsubseteq)$ be an ordinary poset. Then $(`S, `\sqsubseteq)$ is a quantum cpo if and only if $(S, \sqsubseteq)$ is a cpo. Moreover, the functor $`(-) :$* **CPO** $\to$ **qCPO** *is fully faithful.*

As in the classical case, all finite quantum posets are quantum cpos.

Proposition 5.4. *Let $(\mathcal{X}, R)$ be a quantum poset, where $\mathcal{X}$ has a finite number of atoms. Then $(\mathcal{X}, R)$ is a quantum cpo.*

As a consequence, the quantum poset in Example 4.3 is a quantum cpo. We can obtain a more complex example of a quantum cpo as a quantum analog of the powerset under the inclusion order:

Example 5.5. Let n be a positive integer, and equip the atomic quantum set $\mathcal{H}_n$ with the trivial order $I_{\mathcal{H}_n}$, and the ordinary set $\Omega = \{0, 1\}$ the standard order $0 < 1$, which we denote $\sqsubseteq$. Then the quantum poset $[\mathcal{H}_n, `\Omega]_\sqsubseteq$ is a quantum cpo.

Just as **POS** and **qPOS** form an LNL model, so do **CPO** and **qCPO**:

Theorem 5.6. *The category* **qCPO** *is monoidal closed, complete, has all coproducts, and is enriched over* **CPO** *as a monoidal closed category, and the functor $`(-) :$* **CPO** $\to$ **qCPO** *is strong monoidal and has a right adjoint.*

Classical cpos and finite quantum posets are the basic examples of quantum cpos. Theorem 5.6 allows us to combine these basic examples to form new quantum cpos.

5.1 Pointed quantum cpos

Any model of a programming language supporting recursion needs denotations for nonterminating terms. In ordinary domain theory, this is done by passing to the category $\mathbf{CPO}_{\perp!}$ of pointed cpos with strict Scott continuous maps. In this section we formulate the quantum generalization of this category.

We say that a quantum cpo $(\mathcal{X},R)$ is *pointed* if there exists a unique one-dimensional atom $X^{\perp} \in \mathrm{At}(\mathcal{X})$ such that $R(X^{\perp},X) = L(X^{\perp},X)$ for each $X \in \mathrm{At}(\mathcal{X})$. It follows that any ordinary cpo $(P,\sqsubseteq)$ is pointed if and only if $(`P,`\sqsubseteq)$ is pointed. Given pointed quantum cpos $(\mathcal{X},R)$ and $(\mathcal{Y},S)$, a Scott continuous map $F : \mathcal{X} \to \mathcal{Y}$ is said to be *strict* iff $F(X^{\perp},Y^{\perp}) = L(X^{\perp},Y^{\perp})$. We denote the category of pointed quantum cpos with strict Scott continuous maps by $\mathbf{qCPO}_{\perp!}$.

There also is a lift functor $(-)_{\perp} : \mathbf{qCPO} \to \mathbf{qCPO}_{\perp!}$, generalizing the ordinary lift functor in the sense that Diagram 2 of Section 1 commutes. Just as the ordinary lift functor is left adjoint to the inclusion of $\mathbf{CPO}_{\perp!}$ into $\mathbf{CPO}$, we have

Lemma 5.7. *The lift functor* $(-)_{\perp} : \mathbf{qCPO} \to \mathbf{qCPO}_{\perp!}$ *is left adjoint to the inclusion functor* $\mathbf{qCPO}_{\perp!} \to \mathbf{qCPO}$.

There is a symmetric monoidal product $\otimes$ on $\mathbf{qCPO}_{\perp!}$, reminiscent of the smash product on $\mathbf{CPO}_{\perp!}$. The monoidal unit is obtained by lifting the monoidal unit $\mathbf{1}$ of $\mathbf{qCPO}$.

Proposition 5.8. *The category* $\mathbf{qCPO}_{\perp!}$ *is monoidal closed, and the lifting functor* $(-)_{\perp} : \mathbf{qCPO} \to \mathbf{qCPO}_{\perp!}$ *is strong monoidal.*

Proposition 4.7 implies the monoidal product on $\mathbf{qCPO}_{\perp!}$ reflects the order on non-zero morphisms.

Proposition 5.9. *Let $(\mathcal{X}_1,R_1)$, $(\mathcal{X}_2,R_2)$, $(\mathcal{Y}_1,S_1)$ and $(\mathcal{Y}_2,S_2)$ be point quantum cpos, and let $F_1,G_1 : \mathcal{X}_1 \to \mathcal{Y}_1$ and $F_2,G_2 : \mathcal{X}_2 \to \mathcal{Y}_2$ be strict Scott continuous functions such that $F_i \neq 0_{\mathcal{X}_i\mathcal{Y}_i}$ for $i = 1,2$. Then both $F_1 \sqsubseteq G_1$ and $F_2 \sqsubseteq G_2$ if and only if $F_1 \otimes F_2 \sqsubseteq G_1 \otimes G_2$.*

Thus the monoidal product on $\mathbf{qCPO}_{\perp!}$ behaves similarly to the smash product on $\mathbf{CPO}_{\perp!}$. Our main theorem provides the LNL model defined by $\mathbf{CPO}$ and $\mathbf{qCPO}_{\perp!}$:

Theorem 5.10. *The category* $\mathbf{qCPO}_{\perp!}$ *has all coproducts, is complete, symmetric monoidal closed, is enriched over* $\mathbf{CPO}_{\perp!}$*, and has a zero object that is e-initial. Furthermore both compositions in Diagram (2) of Section 1 form the left adjoint component of the model in Diagram (3).*

6 Models of quantum programming languages

Proto-Quipper-M (PQM) is a circuit description language designed to support writing quantum circuits in a high-level functional language and then displaying the evolution of the circuit as a diagram. The "computer" for which PQM is intended is Knill's QRAM model – a classical computer with a quantum coprocessor. Here's an example of the sort of circuit PQM is meant to define. We present the circuit in $\mathbf{qSet}$ for simplicity. It can then be imported in tact into $\mathbf{qCPO}$ by the functor $\mathcal{X} \mapsto (\mathcal{X},I_{\mathcal{X}})$, where $I_{\mathcal{X}}$ is the trivial order, and then into $\mathbf{qCPO}_{\perp!}$ using the lift functor.

Example 6.1. The quantum Fourier transform is a channel from an n-qubit system to itself, and it is given by a sequence of basic quantum gates: the Hadamard gate and rotation gates [16, section 5.1]. For fixed n, the quantum Fourier transform is thus modelled by a function on the quantum set $\mathcal{H}_2^n = \mathcal{H}_2 \times \cdots \times \mathcal{H}_2$ that can be expressed as a composition of these basic gates.

We model a quantum system consisting of an unknown number of qubits by the quantum set $\mathcal{X} = \mathbf{1} + \mathcal{H}_2 + (\mathcal{H}_2 \times \mathcal{H}_2) + \cdots$. Indeed, a pure state on $\ell^{\infty}(\mathcal{X})$ is just a pure state on $\ell^{\infty}(\mathcal{H}_2^n)$ for some

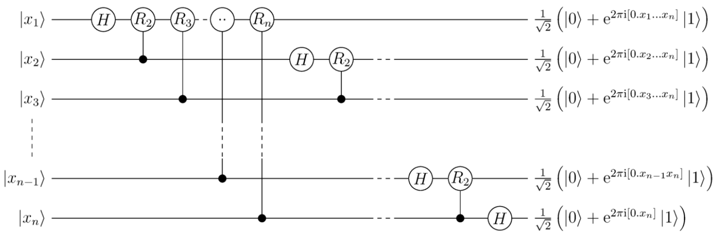
$|x_1\rangle$
$|x_2\rangle$
$|x_3\rangle$
$|x_{n-1}\rangle$
$|x_n\rangle$
H
R_2
R_3
R_n
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_1 \ldots x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_2 \ldots x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_3 \ldots x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_{n-1}x_n]}|1\rangle\right)$
$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_n]}|1\rangle\right)$

Figure 1: The Quantum Fourier Transform [26]

natural number n. A mixed state may be supported on more than one summand, expressing uncertainty about just how many qubits there are! Note that $\mathcal{X}$ is a recursive type over $\mathcal{H}_2$; it is the list type. We have a function $N\colon \mathcal{X} \to$ ‘$\mathbb{N}$ that maps each atom to its dimension; in other words, N is defined by $N(H_2^{\otimes n}, \mathbb{C}_n) = L(H_2^{\otimes n}, \mathbb{C}_n)$ with the other components vanishing. It does not play a direct role in the following description of the quantum Fourier transform.

Using the standard characterization of $\mathcal{X}$ as a fixed point, i.e., $\mathcal{X} \cong \mathbf{1} + \mathcal{X} \times \mathcal{H}_2$, we may define the term $x : \mathcal{X} \vdash \mathrm{Swap}(x) : \mathcal{X}$, which reverses the order of the list, in the usual way. Suppressing this isomorphism, we "unfold" $x : X$, writing it as either $x = * : \mathbf{1}$ or $x = (x_1, y) : \mathcal{X} \times \mathcal{H}_2$, and then we define $\mathrm{Swap}(*) = *$ and $\mathrm{Swap}(x_1, y_1) = (y_1, \mathrm{Swap}(x_1))$. We curry to obtain a function $\mathrm{Swap} : \mathcal{X} \multimap \mathcal{X}$.

Using the same approach, we can define a function $\mathrm{Step} : \mathcal{X} \multimap \mathcal{X}$ modelling each computational step of the algorithm. To formalize this definition, we assume a controlled phase gate $\mathrm{R} : \mathcal{H}_2 \times \mathcal{H}_2 \times$ ‘$\mathbb{N} \multimap \mathcal{H}_2 \times \mathcal{H}_2$, which rotates the phase of the up state of the second qubit by $2\pi/2^n$, whenever the first qubit is in the up state, or equivalently, vice versa. As before, we unfold $x : \mathcal{X}$. If $x = *$, we define $\mathrm{Step}(x) = *$. If $x = (x_1, y)$, then we unfold x_1. If $x_1 = *$, then we define $\mathrm{Step}(x) = Hy$, where $H\colon \mathcal{H}_2 \multimap \mathcal{H}_2$ is the Hadamard gate, and if $x_1 = (x_2, y_2)$, then we define $\mathrm{Step}(x)$ to be the result of applying a controlled phase gate rotating by $2\pi/2^n$ to $(\mathrm{Step}(x_2), y_2)$. The dimension n is classical data, so it can be duplicated and used any number of times at each recursion step, but $\mathcal{X}$ is a quantum type, so *a priori*, we cannot evaluate the function N without consuming the quantum data in any given step in the recursion. For this reason, this recursion should occur over $\mathcal{X} \times$ ‘$\mathbb{N}$, so we can increment the dimension separately.

The swapped quantum Fourier transform $\mathrm{QFT}_0\colon \mathcal{X} \multimap \mathcal{X}$, depicted in the diagram, can now be similarly defined for $x : \mathcal{X}$ distinct from $*$, by $\mathrm{QFT}_0(x) = (\tilde{y}_1, \mathrm{QFT}_0(\mathrm{Swap}(\tilde{x}_1))$, where $(\tilde{x}_1, \tilde{y}_1) = \mathrm{Swap}(\mathrm{Step}(x))$. Of course, the quantum Fourier transform itself is just $\mathrm{QFT} = \mathrm{Swap} \circ \mathrm{QFT}_0$.

There are some important features that this example illustrates. First, lists of qubits are needed to model the memory of the quantum device, so any model for this circuit must have support for (at least) inductive types. Second, we don't know the number of qubits in the state on which the program will run when the circuit is written, so it must be able to respond to any input length. Since the QFT itself is inherently (primitive) recursive, the model must support primitive recursion. While these are the simplest forms of recursion, they make clear the need for these features. Finally, our model also includes **FdHilb**, which allows us to reason concretely about the effect of QFT on a given list of qubits.

Returning to our results, we first consider type recursion. Models built over **CPO** support type recursion if all the type constructors are *algebraically compact* [7]. A direct consequence of Theorem 5.10 is that the **CPO/qCPO** model (3) in Section 1 satisfies all conditions for a **CPO**-LNL model (cf. [14, Definition 5.3.1]), hence we can apply [12, Theorem 3.2] and [14, Theorem 5.3.3] to conclude:

Theorem 6.2. *All* **CPO***-endofunctors on* $\mathbf{qCPO}_{\perp!}$ *are algebraically compact. In particular,* $\otimes$, $+$, $\multimap$ *and* $!$ *are* **CPO***-functors, hence endofunctors on* $\mathbf{qCPO}_{\perp!}$ *that are compositions of these four functors* *are algebraically compact.*

Moreover, by [14, Theorem 6.3.9] any **CPO**-LNL model is a sound model for LNL-FPC, i.e., the language introduced in [14] that can both be seen as an FPC with linear types and as an extension of the circuit-free fragment of PQM with recursive types. By [14, Theorem 7.0.10] any **CPO**-LNL model is computationally adequate at non-linear types if $I \ncong 0$ (which is the case for $\mathbf{qCPO}_{\perp!}$ the zero object is $\mathbf{1}$, whereas the monoidal unit I is $\mathbf{1}_\perp$) and if the monoidal product reflects the order as in Proposition 5.9. So we conclude:

Theorem 6.3. *The linear/non-linear adjunction (3) is a sound model for LNL-FPC that is computationally adequate at non-linear types.*

We now consider term recursion. PQM (including circuits) is extended with term recursion in [12]. We can add the following result:

Theorem 6.4. *The linear/non-linear adjunction (3) is a sound model for PQM with term recursion.*

Proof. By Theorem 3.2 and Definition 3.4 of [12] a model for PQM consists of a symmetric monoidal category $\mathbf{M}$ of circuits (for quantum computing one typically chooses this category to be $\mathbf{FdAlg}^{\mathrm{op}}$), and an LNL-adjunction of the form (1) such that $\mathbf{C}$ and $\mathbf{L}$ have finite coproducts, and such that there exists a strong monoidal embedding $E : \mathbf{M} \to \mathbf{L}$. By [12, Theorem 4.3] such a model is a sound model for PQM with term recursion if the functor $!(-)$ is parametrically algebraically compact, i.e., for fixed $A \in \mathbf{L}$, the endofunctor $A \otimes !(-) : \mathbf{L} \to \mathbf{L}$ is algebraically compact. For $\mathbf{qCPO}_{\perp!}$, the latter condition follows from Theorem 6.2. Finally, we also know $\mathbf{FdAlg}^{\mathrm{op}}$ embeds into $\mathbf{qCPO}_{\perp!}$. □

To our knowledge, the only other model for which the conclusions of Theorem 6.4 hold is (4), whereas the only other model for which the conclusions of Theorem 6.3 hold is (5). Hence our model (3) is the only one we know for which the conclusions of both theorems hold.

7 Future work

Our presentation of the quantum Fourier transform in $\mathbf{qCPO}_{\perp!}$ raises the issue of computational adequacy of our model. We do not have a proof, but we believe a proof strategy similar to that in [14] will prove this for circuits in our model, but some details still need checking.

Our model (3) is the only one known that is both sound for Proto-Quipper-M extended with term recursion, and also sound and computationally adequate at non-linear types for LNL-FPC. The latter can be regarded as the circuit-free fragment of Proto-Quipper-M extended with recursive types. Our next goal is to show that (3) is a sound model for Proto-Quipper-M extended with recursive types that is computationally adequate at non-linear types. Since our model also supports affine types, we expect that an adequacy result at all types may follow from results in [19].

We are also working on quantizing the probabilistic power domain monad, whose existence would imply (3) supports state preparation. With this in hand, we anticipate extending Proto-Quipper-M with dynamic lifting, i.e., the execution of quantum circuits. We also expect that quantizing the probabilistic power domain will yield another model for the quantum lambda calculus with term recursion [17].

Acknowledgements

Thanks to Vladimir Zamdzhiev for insightful discussions, and because this work was partly inspired by prior work with him. We also thank Xiaodong Jia for his comments. This work is supported by AFOSR under MURI grant FA9550-16-1-0082.

References

[1] P. N. Benton and P. Wadler, *Linear logic, monads and the lambda calculus*, LICS 1996, 1996.

[2] P.N. Benton, *A mixed linear and non-linear logic: Proofs, terms and models*, 1994. Technical Report.

[3] ———, *A mixed linear and non-linear logic: Proofs, terms and models*, Computer science logic: 8th workshop, CSL '94, selected papers, 1995, pp. 121–135.

[4] F. Borceux, *Handbook of categorical algebra 1: Basic category theory*, Cambridge University Press, 1994.

[5] L. Caires and F.. Pfenning, *Session types as intuitionistic linear propositions*, Proceedings of CONCUR 2010, ACM Press, 2010, pp. 222-236., 2010.

[6] K. Cho and A. Westerbaan, *Von Neumann algebras form a model for the quantum lambda calculus* (2016), available at `1603:02113`.

[7] M. P. Fiore, *Axiomatic domain theory in categories of partial maps*, Ph.D. Thesis, 1994.

[8] Marcelo Fiore and Gordon Plotkin, *An axiomatization of computationally adequate domain theoretic models of FPC*, LICS, 1994, pp. 92–102.

[9] A. Kornell, *Quantum sets*, to appear in J. Math. Phys. (2020), available at `arXiv:1804.00581`.

[10] G. Kuperberg and N. Weaver, *A Von Neumann Algebra Approach to Quantum Metrics: Quantum Relations*, Memoirs of the American Mathematical Society, American Mathematical Society, 2012.

[11] Daniel J Lehmann and Michael B Smyth, *Algebraic specification of data types: A synthetic approach*, Mathematical Systems Theory (1981).

[12] Bert Lindenhovius, Michael Mislove, and Vladimir Zamdzhiev, *Enriching a linear/non-linear lambda calculus: A programming language for string diagrams*, Proceedings of the 33rd annual acm/ieee symposium on logic in computer science, 2018, pp. 659–668.

[13] ———, *Mixed linear and non-linear recursive types*, Proc. ACM Program. Lang. **3** (July 2019), no. ICFP, 111:1–111:29.

[14] ———, *LNL-FPC: The Linear/Non-Linear Fixpoint Calculus, journal version of* [13], to appear in LMCS (2020).

[15] P.-A. Melliès, *Categorical models of linear logic revisited*, hal-00154229 (2003).

[16] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, 10th Anniversary edition, Cambridge University Press, 2010.

[17] Michele Pagani, Peter Selinger, and Benoît Valiron, *Applying quantitative semantics to higher-order quantum computing*, The 41st annual ACM SIGPLAN-SIGACT symposium on principles of programming languages, POPL '14, San Diego, CA, USA, January 20-21, 2014, 2014, pp. 647–658.

[18] J. Paykin, R. Rand, and S. Zdancewic, *QWIRE: a core language for quantum circuits*, POPL, 2017, pp. 846–858.

[19] R. Péchoux, S. Perdrix, M. Rennela, and V. Zamdzhiev, *Quantum programming with inductive datatypes: Causality and affine type theory* (2019), available at `1910.09633`.

[20] M. Rennela and S. Staton, *Classical control and quantum circuits in enriched category theory*, 2017. MFPS XXXIII.

[21] ———, *Classical control, quantum circuits and linear logic in enriched category theory* (2017), available at `1711.05159`.

[22] Francisco Rios and Peter Selinger, *A categorical model for a quantum circuit description language*, Proceedings 14th international conference on quantum physics and logic, QPL 2017, Nijmegen, the Netherlands, 3-7 july 2017., 2017, pp. 164–178.

[23] U. Sasaki, *Orthocomplemented lattices satisfying the exchange axiom*, J. Sci. Hiroshima Univ. Ser. A **17** (1954), no. 3, 293–302.

[24] P. Selinger and B. Valiron, *A lambda calculus for quantum computation with classical control*, Mathematical Structures in Computer Science **16** (2006), no. 3, 527–552.

[25] M. Takesaki, *Theory of Operator Algebra I*, Springer, 2000.

[26] Trenar3, *Quantum fourier transform*, Wikimedia, available at https://commons.wikimedia.org/wiki/File:Q_fourier_-nqubits.png.

[27] N. Weaver, *Quantum relations*, Mem. Amer. Math. Soc. **215** (2012).

[28] ———, *Hereditarily antisymmetric operator algebras* (2019), available at `1902.09989`.