Complete Positivity and Natural Representation of Quantum Computations

Mathys Rennela[1]

Institute for Computing and Information Sciences
Radboud University
Nijmegen, The Netherlands

Sam Staton[2]

Department of Computer Science
Oxford University
Oxford, United Kingdom

Abstract

We propose a new 'quantum domain theory' in which Scott-continuous functions are replaced by Scott-continuous natural transformations.

Completely positive maps are widely accepted as a model of first-order quantum computation. We begin by establishing a categorical characterization of completely positive maps as natural families of positive maps. We explore this categorical characterization by building various representations of quantum computation based on different structures: affine maps between cones of positive elements, morphisms of algebras of effects, and affine maps of convex sets of states. By focusing on convex dcpos, we develop a quantum domain theory and show that it supports some important constructions such as tensor products by quantum data, and lifting.

Keywords: Operator algebra, complete positivity, quantum computation, domain theory, convex set

Introduction

This paper is about semantic models of quantum computation. In common with other approaches to programming language semantics, the general idea is to interpret a type A as a space $[\![A]\!]$ of observations about A. One interprets a computation $x : A \vdash t : B$, that produces something t of type B but depends on something x of type A, as a predicate transformer $[\![B]\!] \to [\![A]\!]$, which maps a predicate on B to its weakest precondition. (See e.g. [4,18,3].)

---

[1] Email: mathysr@cs.ru.nl
[2] Email: sam.staton@cs.ox.ac.uk

In more detail, one interprets a type A as a C*-algebra of operators $[\![A]\!]$, and the computations describe maps that are in particular *positive*: it is actually only the positive elements of the algebra that describe the observables, and these must be preserved. Moreover the maps should be *completely positive*. Informally this means that it makes sense to run the computation on a subsystem of a bigger system; for example, we could adjoin an extra qubit to the system and still run the computation. More formally it means that not only does the map $[\![t]\!] : [\![B]\!] \to [\![A]\!]$ preserve positive elements, but also $\mathrm{id}_{[\![\mathrm{qubit}]\!]} \otimes [\![t]\!] : [\![\mathrm{qubit}]\!] \otimes [\![B]\!] \to [\![\mathrm{qubit}]\!] \otimes [\![A]\!]$ preserves positive elements.

The first contribution of this paper (Section 2) is a technique for building representations of quantum computation in terms of completely positive maps. In the second half of the paper (Section 3) we demonstrate our technique by making some first steps in the development of a 'quantum domain theory'.

A technique for building representations

Here, a representation is a full and faithful functor $F : \mathbf{C} \to \mathbf{R}$, that is, a functor for which each function $F_{A,B} : \mathbf{C}(A, B) \to \mathbf{R}(F(A), F(B))$ is a bijection.

From a programming language perspective, where objects interpret types and morphisms interpret programs, a representation result gives two things. Firstly, it gives a way of interpreting types as different mathematical structures, which can be illuminating or convenient, while retaining essentially the same range of interpretable programs. Secondly, since $\mathbf{R}$ may be bigger than $\mathbf{C}$, it gives the chance to interpret more types without altering the interpretation of programs at existing types.

There are several existing representation results which allow us to understand and analyze quantum computations in terms of different structures, such as convex sets (e.g. [11]), domains (e.g. [18]), partial monoids and effect algebras (e.g. [10]). However, many of these representation results are only valid for positive maps, and so they do not fully capture quantum computation. Our contribution is a general method for extending these results to completely positive maps. Roughly, the method allows us to convert a full and faithful functor

$$(\text{positive maps}) \longrightarrow \mathbf{R}$$

(where $\mathbf{R}$ is an arbitrary category) into a full and faithful functor

$$(\text{completely positive maps}) \longrightarrow [\mathbf{N}, \mathbf{R}]$$

into a functor category, where $\mathbf{N}$ is a category whose objects are natural numbers.

Towards a quantum domain theory

In the second part of the paper we demonstrate our technique by making some first steps in the development of a 'quantum domain theory'. The ultimate goal in this line of work is to analyze all kinds of quantum programming by solving

domain equations involving qubits. For example, one should expect a solution to the equation

$$A = (\text{qubit} \otimes A)_\perp$$

which would be a type of infinite streams of qubits. In this paper we exhibit (for the first time) a domain theory that supports qubits and lifting.

In brief, we begin from the observation that taking states of a W*-algebra yields a representation of positive maps in terms of affine maps between convex sets. We use this to build a representation

(W*-algebras and completely positive maps) $\longrightarrow$ [**N**, (convex sets and affine maps)]

We can now extend the representation with domain theoretic structure, by replacing convex sets with directed complete convex sets. Thus 'quantum domains' are defined to be functors

$$\mathbf{N} \to (\text{convex dcpos and affine continuous maps})$$

and quantum computations are interpreted as affine Scott-continuous natural transformations between quantum domains. We show that this class of quantum domains supports various constructions, including tensor with quantum data and lifting.

1 Preliminaries

1.1 Linear maps of C*-algebras

The basic idea of matrix mechanics is that the observables for a quantum system are elements of a C*-algebra. Recall that a (unital) C*-algebra is a vector space over the field of complex numbers that also has multiplication, a unit and an involution, satisfying associativity laws for multiplication, involution laws (e.g. $x^{**} = x$, $(xy)^* = y^*x^*$, $(\alpha x)^* = \bar{\alpha}(x^*)$) and such that the spectral radius provides a norm making it a Banach space.

A key source of examples of C*-algebras are the algebras M_k of $k \times k$ complex matrices, with matrix addition and multiplication, and where involution is conjugate transpose. In particular the set $M_1 = \mathbb{C}$ of complex numbers has a C*-algebra structure, and the 2×2 matrices, M_2, contain the observables of qubits.

If A is a C*-algebra then the $k \times k$ matrices valued in A also form a C*-algebra, $M_k(A)$. For instance $M_k(\mathbb{C}) = M_k$, and $M_k(M_l) \cong M_{k \times l}$. Informally, we can think of the C*-algebra $M_k(A)$ as representing k entangled copies of A. This can be thought of as a kind of tensor product: as a vector space $M_k(A)$ is a tensor product $M_k(\mathbb{C}) \otimes A$. There are various ways to extend this to define a tensor product on arbitrary C*-algebras, but we will not need tensor products other than $M_k(A)$ in this paper.

The 'direct sum' $X \oplus Y$ of C*-algebras is given by the cartesian product of the underlying sets. It has the universal property of the categorical product. The C*-algebra $\mathbb{C} \oplus \mathbb{C}$ represents classical bits.

An element $x \in A$ is *positive* if it can be written in the form $x = y^*y$ for $y \in A$. We denote by A^+ the set of positive elements of a C*-algebra A and define the following partial order on the elements of A: $x \leq y$ if and only if $(y - x) \in A^+$.

We consider the following kinds of map of C*-algebras. Let $f : A \to B$ be a linear map between the underlying vector spaces.

P The map f is *positive* if it preserves positive elements and therefore restricts to a function $A^+ \to B^+$. A positive map $A \to \mathbb{C}$ will be called a state on A.

U The map f is *unital* if it preserves the unit, i.e. $f(1_A) = 1_B$;

SU The map f is *sub-unital* if $f(1_A) \leq 1_B$;

CP The map f is *completely positive* if it is n-positive for every $n \in \mathbb{N}$, i.e. the map $M_n(f) : M_n(A) \to M_n(B)$ defined for every matrix $[x_{i,j}]_{i,j \leq n} \in M_n(A)$ by $M_n(f)([x_{i,j}]_{i,j \leq n}) = [f(x_{i,j})]_{i,j \leq n}$ is positive for every $n \in \mathbb{N}$.

As a matter of convenience, we will denote through this paper different classes of maps by the first letters of the names of the properties they follow. In particular, the term (C)P((S)U)-map will refer respectively to a (completely) positive (sub-)unital map. We write $\mathbf{C^*}\text{-}\mathbf{Alg}_{\mathrm{CPSU}}$ for the category of C*-algebras and completely positive sub-unital maps, and so on.

We refer the interested reader to [20,24] for a complete introduction to C*-algebras.

1.2 Representation of quantum computations

For the reader familiar with semantics of programming languages, we recall basic ideas for the semantics of quantum programming languages in C*-algebras. A type A is interpreted as a C*-algebra $[\![A]\!]$. A terminating computation-in-context $x_1 : A_1, \ldots, x_n : A_n \vdash t : B$ is interpreted as a CPU-map $B \to \bigotimes_i A_i$, transforming observations about the result type to requisite observations about the free variables. We let $[\![\mathsf{qubit}]\!] = M_2$, and the empty tensor is $\mathbb{C}$, and so a computation $\vdash t : \mathsf{qubit}$ that generates a qubit with no free variables is interpreted as a CPU-map $M_2 \to \mathbb{C}$. In the theory of operator algebras, CPU-maps into $\mathbb{C}$ are called *states*.

1.3 Isometries and pure states

An important class of CP-maps comes from multiplication by matrices. Let A be a C*-algebra. Any $m \times n$ complex matrix F induces a CP-map $F^*\_F : M_m(A) \to M_n(A)$ given by $(F^*\_F)(x) = F^*xF$, where F^* is the conjugate transpose of F. This is a CPU-map if F is an isometry, i.e. $F^*F = I$. In particular, putting $n = 1$ and $A = \mathbb{C}$, any vector $v \in \mathbb{C}^2$ with $v^*v = 1$ induces a state $v^*\_v : M_2 \to \mathbb{C}$, called a 'pure state'.

In what follows we will consider the vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad |+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad |\rho\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix}$$

which all induce CPU-maps $M_2 \to \mathbb{C}$. One often writes $\langle v|$ for the conjugate

transpose $|v\rangle^*$, so the induced CPU-map can be written $\langle v| \_ |v\rangle : M_2(A) \to A$. In particular:

$$\langle 0| \begin{smallmatrix} a & b \\ c & d \end{smallmatrix} |0\rangle = a \qquad\qquad \langle +| \begin{smallmatrix} a & b \\ c & d \end{smallmatrix} |+\rangle = \tfrac{1}{2}(a + b + c + d)$$

$$\langle 1| \begin{smallmatrix} a & b \\ c & d \end{smallmatrix} |1\rangle = d \qquad\qquad \langle \rho| \begin{smallmatrix} a & b \\ c & d \end{smallmatrix} |\rho\rangle = \tfrac{1}{2}(a - ib + ic + d)$$

2　Naturality and representations of complete positivity

In this section, we will provide a new categorical characterization of completely positive maps as natural families of positive maps (§2.1–§2.3). This gives a technique for building representations of completely positive maps (§2.3), which we demonstrate with several examples: positive cones (§2.4–2.5), effects (§2.6), and states (§2.7).

2.1　Complete positivity as naturality

In Section 1.3 we considered how a matrix $F \in \mathbb{C}^{m \times n}$ induces a completely positive map $F^*\_F : M_m \to M_n$. This construction is functorial. To make this precise, we introduce the category $\mathbb{N}_{\mathrm{Mat}}$ of complex matrices: the objects are non-zero natural numbers seen as dimensions, and the morphisms $m \to n$ are $m \times n$ complex matrices. Composition is matrix multiplication. (We remark that the category $\mathbb{N}_{\mathrm{Mat}}$ is equivalent to the category of finite-dimensional complex vector spaces and linear maps, since every finite-dimensional vector space is isomorphic to $\mathbb{C}^n$. It is also equivalent to the category of finite-dimensional Hilbert spaces and linear maps, since every such space has a canonical inner product.)

The construction of matrices of elements of a C*-algebra can be made into a functor $\mathbf{C}^*\text{-}\mathbf{Alg}_{\mathrm{CP}} \times \mathbb{N}_{\mathrm{Mat}} \to \mathbf{C}^*\text{-}\mathbf{Alg}_{\mathrm{P}}$. It takes a pair (A, m) to $M_m(A)$ and a pair of morphisms $(f, F) : (A, m) \to (B, n)$ to the positive map $F^*(f\_)F : M_m(A) \to M_n(B)$.

We will consider this functor in curried form, $M : \mathbf{C}^*\text{-}\mathbf{Alg}_{\mathrm{CP}} \to [\mathbb{N}_{\mathrm{Mat}}, \mathbf{C}^*\text{-}\mathbf{Alg}_{\mathrm{P}}]$. It takes a C*-algebra A to a functor, i.e. an indexed family of C*-algebras, $M(A) = \{M_n(A)\}_n$. A completely positive map $f : A \to B$ is taken to the corresponding family of positive maps $M(f) = \{M_n(f) : M_n(A) \to M_n(B)\}_n$. This gives our main result: the completely positive maps are in natural bijection with families of positive maps.

Theorem 2.1 *The functor $M : \mathbf{C}^*\text{-}\mathbf{Alg}_{\mathrm{CP}} \to [\mathbb{N}_{\mathrm{Mat}}, \mathbf{C}^*\text{-}\mathbf{Alg}_{\mathrm{P}}]$ is full and faithful.*

The rest of this section is dedicated to the proof of Theorem 2.1. Faithfulness is obvious, since for any CP-map $f : A \to B$ we have $M(f)_1 = f$. To show fullness we begin with the following lemma.

Lemma 2.2 *Consider two positive maps $f_2 : M_2(B) \to M_2(A)$ and $f_1 : B \to A$ of C*-algebras. The following conditions are equivalent:*

(i) $\forall y \in M_2(B), v \in \mathbb{C}^2. \quad v^*(f_2(y))v = f_1(v^*yv)$

(ii) $f_2 = M_2(f_1)$.

Proof. We can show that (ii) implies (i) with the following argument:

For every 2-by-2 matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \in M_2(B)$:

$$v^*(M_2(f_1) \begin{bmatrix} a & b \\ c & d \end{bmatrix})v = v^* \begin{bmatrix} f_1(a) & f_1(b) \\ f_1(c) & f_1(d) \end{bmatrix} v = f_1(v^* \begin{bmatrix} a & b \\ c & d \end{bmatrix} v)$$

since v^*_v maps a 2-by-2 matrix to a linear combination of its entries, which will be preserved by the linear map f_1.

We will now focus on the proof that $(i) \implies (ii)$.

Consider $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \in M_2(B)$, let $\begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} = f_2 \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and suppose that (i) holds.

We use the assumption (i) with the vectors $|0\rangle$, $|1\rangle$, $|+\rangle$ and $|\rho\rangle$, to obtain $a' = f_1(a)$, $d' = f_1(d)$, $a' + b' + c' + d' = f_1(a) + f_1(b) + f_1(c) + f_1(d)$ and $a' - ib' + ic' + d = f_1(a) - if_1(b) + if_1(c) + f_1(d)$. We can combine these four facts to also deduce that $b' = f_1(b)$ and $c' = f_1(c)$.

And thus finally, we observe that $f_2 \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} f_1(a) & f_1(b) \\ f_1(c) & f_1(d) \end{bmatrix} = M_2(f_1) \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

This concludes our proof of Lemma 2.2.   □

We use Lemma 2.2 to establish fullness in our proof of Theorem 2.1. Consider a natural transformation $f : M(A) \Rightarrow M(B)$, described by the following commuting diagram:

$$
\begin{array}{ccc}
n & M_n(A) \xrightarrow{f_n} M_n(B) \\
\Big\downarrow g & g\otimes A \Big\downarrow \qquad\quad \Big\downarrow g\otimes B \\
m & M_m(A) \xrightarrow{f_m} M_m(B)
\end{array}
$$

where g is a n-by-m complex matrix in $\mathbb{N}_{\text{Mat}}$.

As the map g can correspond to any pure state $v^*\_v : M_2 \to \mathbb{C}$, the condition (i) of Lemma 2.2 holds and therefore $f_2 = M_2(f_1)$. By induction, if $f_{2^{k+1}} = M_2(f_{2^k}) : M_{2^{k+1}}(A) \to M_{2^{k+1}}(B)$ for some natural number k, then $f_{2^{k+2}} = M_2(f_{2^{k+1}}) : M_{2^{k+2}}(A) = M_2(M_{2^{k+1}}(A)) \to M_2(M_{2^{k+1}}(B)) = M_{2^{k+2}}(B)$ since the condition (i) holds when one consider $M_{2^{k+1}}(A)$ as A and $M_{2^{k+1}}(B)$ as B. Then $M_{2^k}(f_1) = f_{2^k}$ for every natural number k. It follows that the map f_1 is 2^k-positive for every $k \in \mathbb{N}$. Finally, since n-positive maps are $(n-1)$-positive (see $M_{n-1}(B)$ as the left upper block of $M_n(B)$ for $n \geq 1$), one can conclude that f_1 is a completely positive map, with $M_n(f_1) = f_n$ for every natural number n.

So $M : \mathbf{C}^*\text{-}\mathbf{Alg}_{\text{CP}} \to [\mathbb{N}_{\text{Mat}}, \mathbf{C}^*\text{-}\mathbf{Alg}_{\text{P}}]$ is a full functor. This concludes our proof of Theorem 2.1.

2.2   *Variations on the characterization theorem*

The proof of Theorem 2.1 is quite flexible and can accommodate some variation in the index category and the base category.

For a first variation, we change the index category so that we can focus on unit-preserving completely positive maps. We consider the subcategory $\mathbb{N}_{\mathrm{Isom}}$ of $\mathbb{N}_{\mathrm{Mat}}$ with the same objects but where the morphisms are isometries ($F^*F = I$).

We will be quite general about the base category. Consider a subcategory $\mathbf{V}$ of $\mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P}$ that is closed under matrix algebras, i.e.

$$\mathbb{C} \in \mathbf{V} \quad \text{and} \quad A \in \mathbf{V} \implies M_n(A) \in \mathbf{V}. \tag{1}$$

Then define $\mathbf{V}_\mathrm{C}$ to be the closure of $\mathbf{V}$ under matrices of morphisms: the objects of $\mathbf{V}_\mathrm{C}$ are the same as the objects of $\mathbf{V}$, and a function $f : A \to B$ is in $\mathbf{V}_\mathrm{C}$ if $M_n(f) : M_n(A) \to M_n(B)$ is in $\mathbf{V}$ for all n. For instance, $(\mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P})_\mathrm{C} = \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{CP}$.

Theorem 2.3 *Consider a subcategory $\mathbf{V}$ of $\mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P}$ satisfying (1) and such that the matrices functor*

$$\mathbf{V}_\mathrm{C} \times \mathbb{N}_{\mathrm{Isom}} \to \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P}$$

factors through $\mathbf{V}$. It induces a full and faithful functor $\mathbf{V}_\mathrm{C} \to [\mathbb{N}_{\mathrm{Isom}}, \mathbf{V}]$.

There are other variations on the result, by changing the index category to a different subcategory of $\mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{CP}$. We focus on two examples which are particularly relevant in the enriched setting (see §2.4):

- Let $\mathbb{N}_\mathrm{CP}$ be the category whose objects are natural numbers and where a morphism $m \to n$ is a completely positive map $M_m \to M_n$. In the literature, this category is often called **CPMs** [15,3], **W** [21] or **CPM**[**FdHilb**] [22].

- Let $\mathbb{N}_\mathrm{CPU}$ be the category whose objects are natural numbers and where a morphism $m \to n$ is a completely positive unital map $M_m \to M_n$. The dual of this category can be thought of as comprising the trace-preserving completely positive maps between density matrices (e.g. $\mathbf{Q}'_s$ in [15, Def. 2.9]).

The matrices functors

$$\mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{CP} \times \mathbb{N}_{\mathrm{Mat}} \to \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P} \quad \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{CPU} \times \mathbb{N}_{\mathrm{Isom}} \to \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{PU}$$

extend to functors

$$\mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{CP} \times \mathbb{N}_\mathrm{CP} \to \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P} \quad \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{CPU} \times \mathbb{N}_\mathrm{CPU} \to \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{PU}$$

using the idea that $M_n(A) = M_n \otimes A$, and if $f : M_m \to M_n$ is completely positive then so too is $f \otimes A : M_m(A) \to M_n(A)$.

Theorem 2.4 *Consider a subcategory $\mathbf{V}$ of $\mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P}$ that is closed under matrix algebras (1) and such that the matrices functor*

$$\mathbf{V}_\mathrm{C} \times \mathbb{N}_{\mathrm{CP(U)}} \to \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P}$$

factors through $\mathbf{V}$. It induces a full and faithful functor $\mathbf{V}_\mathrm{C} \to [\mathbb{N}_{\mathrm{CP(U)}}, \mathbf{V}]$.

2.3   Representations of quantum computation

Our intention is to use Theorem 2.1 to build representation results for completely positive maps out of representation results for positive maps. For instance, the following corollary is immediate:

Corollary 2.5 *Every full and faithful functor $F : \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P} \to \mathbf{R}$ induces a full and faithful functor $\mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{CP} \to [\mathbb{N}_\mathrm{Mat}, \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P}] \to [\mathbb{N}_\mathrm{Mat}, \mathbf{R}]$.*

For the remainder of this section we illustrate our technique by building representation theorems for CP-maps.

2.4   Example: positive cones of C*-algebras

We show how to build a representation for CP-maps out of affine maps between cones. We begin by recalling basic definitions. For any $i \leq m$ let $\delta^{i,m}$ be the Kronecker vector with $\delta^{i,m}_i = 1$ and $\delta^{i,m}_j = 0$ for $i \neq j$.

Definition 2.6 A *cone* is a set X together with an m-ary function $(\vec{r})_X : X^m \to X$ for each vector $\vec{r} = (r_1 \dots r_m)$ of non-negative real numbers, often written infix as $\sum_i r_i.x_i$, such that for each i, $\delta^{i,m}_i(x_1, \dots x_m) = x_i$, and for each $m \times n$ matrix $(s_{i,j})_{i,j}$ of non-negative real numbers, $\sum_i r_i.(\sum_j(s_{i,j}.x_j)) = \sum_j((\sum_i(r_i.s_{i,j})).x_j)$.

A *homomorphism* of cones is a function that preserves the algebraic structure. Homomorphisms are often called *affine maps*. The category **Cone** is the category of cones together with affine maps.

There are different ways to formulate this definition. A subset of a real vector space forms a cone if it is closed under addition and multiplication by positive real scalars, and conversely every cone arises in this way. This motivates the terminology 'affine map'.

Alternatively, the abstract definition of cones can be reformulated in terms of scalar multiplication and binary addition, and all the m-ary operations can be built from these operations.

Representations

For any C*-algebra the positive elements form a cone.

Lemma 2.7 *Taking the cone of positive elements yields a full and faithful functor $(-)^+ : \mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{P} \to \mathbf{Cone}$.*

Proof. [notes] Any positive map $f : X \to Y$ is completely defined by its action on X^+: an arbitrary element $x \in X$ can be written as a linear sum of four positive elements $x = x_1 + ix_2 - x_3 - ix_4$, for x_i all positive [7, Lemma 2.2], determining the value $f(x)$, which does not depend on a particular decomposition of x.   □

Example 2.8 The functor $\mathbf{C}^*\text{-}\mathbf{Alg}_\mathrm{CP} \to [\mathbb{N}_\mathrm{Mat}, \mathbf{Cone}]$ taking A to $n \mapsto (M_n(A))^+$ is full and faithful.

This appears to be a new categorical way to formulate the theory of matrix ordered spaces (e.g. [6], [17, Ch. 13]).

Enrichment

Example 2.9 The functor $\mathbf{C}^*\text{-}\mathbf{Alg}_{\mathrm{CP}} \to [\mathbb{N}_{\mathrm{CP}}, \mathbf{Cone}]$ taking A to $n \mapsto (M_n(A))^+$ is full and faithful.

The category $\mathbb{N}_{\mathrm{CP}}$ is enriched in **Cone**: one can scale completely positive maps and add them too. This leads us to focus on locally affine functors $F : \mathbb{N}_{\mathrm{CP}} \to \mathbf{Cone}$, i.e., functors that preserve the cone structure of the hom-sets, i.e., enriched presheaves [5].

The category of locally affine functors $\mathbb{N}_{\mathrm{CP}} \to \mathbf{Cone}$ is the free colimit completion of $\mathbb{N}_{\mathrm{CP}}$ as a **Cone**-enriched category. This draws a comparison with other models of quantum computation, which partly inspired the current work. Firstly there are models based around biproduct completions of $\mathbb{N}_{\mathrm{CP}}$ (e.g. [21] and [16]); this is relevant since a biproduct completion is a free coproduct completion of $\mathbb{N}_{\mathrm{CP}}$ as a **Cone**-enriched category. Secondly there are models based around (non-enriched) colimit completions of categories such as $\mathbb{N}_{\mathrm{CPU}}$ [15].

2.5   Example: Directed complete cones and W*-algebras

Directed-completeness

Recall that a directed complete partial order is a partial order in which every directed set has a least upper bound. A bounded dcpo (bdcpo) is a partial order in which every directed set that has an upper bound has a least upper bound.

Definition 2.10 A *conic bdcpo* (or d-cone) is a cone X (Def. 2.6) equipped with a bdcpo structure such that the operations $(\vec{r})_X : X^m \to X$ are all Scott-continuous functions from the product bdcpo. This yields a category **dCone** of conic bdcpos and affine Scott-continuous maps between them.

Definition 2.11 A C*-algebra A is called monotone complete if the cone A^+ of positive elements is a conic bdcpo. A positive map between monotone complete C*-algebras $A \to B$ is called *normal* if its restriction to the positive cone preserves joins of bounded directed sets.

We will focus on W*-algebras, which are monotone complete C*-algebras such that for every non-zero positive element $x \in A^+$ there is a normal positive map $f : A \to \mathbb{C}$ such that $f(x) \neq 0$ (e.g. [24, III.3.16]). W*-algebras encompass all finite dimensional C*-algebras, and also the algebras of bounded operators on any Hilbert spaces, the function space $L^\infty(X)$ for some standard measure space X, and the space $\ell^\infty(\mathbb{N})$ of bounded sequences.

We write $\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{P}}$ for the category of W*-algebras and normal positive maps, and $\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CP}}$ for the category of W*-algebras and normal completely positive maps, and so on. Essentially by definition we have a full and faithful functor $(-)^+ : \mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{P}} \to \mathbf{dCone}$. In consequence:

Example 2.12 The functor $\mathbf{W^*}\text{-}\mathbf{Alg}_{\mathrm{CP}} \to [\mathbb{N}_{\mathrm{Mat}}, \mathbf{dCone}]$ taking A to $n \mapsto (M_n(A))^+$ is full and faithful.

2.6 Examples: Effects

We briefly discuss examples based on the theory of effects of C*-algebras, although we will not elaborate on this any further in this article.

An *effect* of a C*-algebra is a positive element that is less than 1. Informally, an effect is a kind of 'unsharp' predicate. The effects $[0,1]_A$ of a C*-algebra A form an algebraic structure called an 'effect module': they have a partial monoid structure given by addition, a top element, and they admit multiplication by scalars in the unit interval $[0,1]$.

Taking effects actually yields a full and faithful functor $\mathbf{C^*}\text{-}\mathbf{Alg}_{\mathrm{PU}} \to \mathbf{EMod}$ (see e.g. [7]), giving us another illustration of our framework:

Example 2.13 The functor $\mathbf{C^*}\text{-}\mathbf{Alg}_{\mathrm{CPU}} \to [\mathbb{N}_{\mathrm{Isom}}, \mathbf{EMod}]$, taking A to $n \mapsto [0,1]_{M_n(A)}$, is full and faithful.

There are some interesting variations on this example.

Example 2.14 • A 'generalized effect module' is an effect module without a top element. By ignoring the top effects we obtain a full and faithful functor $\mathbf{C^*}\text{-}\mathbf{Alg}_{\mathrm{PSU}} \to \mathbf{GEMod}$ [7] and hence a full and faithful functor $\mathbf{C^*}\text{-}\mathbf{Alg}_{\mathrm{CPSU}} \to [\mathbb{N}_{\mathrm{Isom}}, \mathbf{GEMod}]$.

- In a W*-algebra, the effects form a directed complete effect module; this gives a full and faithful functor $\mathbf{W^*}\text{-}\mathbf{Alg}_{\mathrm{PU}} \to \mathbf{dEMod}$ [18] and hence we obtain a new full and faithful functor $\mathbf{W^*}\text{-}\mathbf{Alg}_{\mathrm{CPU}} \to [\mathbb{N}_{\mathrm{Isom}}, \mathbf{dEMod}]$.

- Similarly, from a full and faithful functor $\mathbf{W^*}\text{-}\mathbf{Alg}_{\mathrm{PSU}} \to \mathbf{dGEMod}$ [18] we obtain a full and faithful functor $\mathbf{W^*}\text{-}\mathbf{Alg}_{\mathrm{CPSU}} \to [\mathbb{N}_{\mathrm{Isom}}, \mathbf{dGEMod}]$.

2.7 Examples: States

Convex sets

Definition 2.15 A *convex set* is a set X together with an m-ary function $(\vec{r})_X : X^m \to X$ for each vector $\vec{r} = (r_1 \dots r_m)$ of non-negative real numbers with $\sum_i r_i = 1$, such that for each i, $\delta_X^{i,m}(x_1, \dots x_m) = x_i$, and for each $m \times n$ matrix $(s_{i,j})_{i,j}$ of non-negative real numbers such that $\sum_j s_{i,j} = 1$, we have $\sum_i r_i.(\sum_j (s_{i,j}.x_j)) = \sum_j ((\sum_i (r_i.s_{i,j})).x_j)$.

A *homomorphism* of convex sets is a function that preserves the algebraic structure. Homomorphisms are often called *affine maps*.

The definition of convex sets can be reformulated in terms of a weighted binary addition (e.g. [14]). For example, a subset of a real vector space is convex if it is closed under convex sums.

For a W*-algebra A, consider the normal state space $\mathcal{NS}(A) = \mathbf{W^*}\text{-}\mathbf{Alg}_{\mathrm{PU}}(A, \mathbb{C})$. The hom-sets of the category $\mathbf{W^*}\text{-}\mathbf{Alg}_{\mathrm{PU}}$ can be given a con-

vex structure, considered as a subset of the vector space of all linear maps. The mapping $\mathcal{NS}(-)$ can thus be turned into a contravariant functor to the category of convex sets, which acts as follows on positive unital maps: $\mathcal{NS}(\,A \xrightarrow{f} B\,) = (-) \circ f : \mathcal{NS}(B) \to \mathcal{NS}(A)$.

Theorem 2.16 ([20],[2],[8]) *The functor* $\mathcal{NS}(-) : \mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{PU}}^{\mathrm{op}} \to \mathbf{Conv}$ *is full and faithful.*

(The normal states functor is not faithful when restricted to completely positive maps ($\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}^{\mathrm{op}}$): the transpose map is positive but not completely positive, and it yields an isomorphism of convex sets.)

Example 2.17 The functor $\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}^{\mathrm{op}} \to [\mathbb{N}_{\mathrm{Isom}}^{\mathrm{op}}, \mathbf{Conv}]$, taking A to $n \mapsto \mathcal{NS}(M_n(A))$, is full and faithful.

Example 2.18 The functor $\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}^{\mathrm{op}} \to [\mathbb{N}_{\mathrm{CPU}}^{\mathrm{op}}, \mathbf{Conv}]$, taking A to $n \mapsto \mathcal{NS}(M_n(A))$, is full and faithful.

3 Quantum domain theory

In this section, we will use the techniques in Section 2 to begin to build a 'quantum domain theory': a new categorical model for quantum computations based on order-valued functors.

We proceed by analogy with classical domain theory. Recall that in classical domain theory there are two categories that play important roles: firstly a category **Predom** of dcpos and Scott-continuous functions, and secondly a category **Dom**₁ of pointed dcpos (dcpos with a bottom element) and strict Scott-continuous functions (functions that preserve the bottom element). Lifting (freely adding a bottom element) is left adjoint to the evident forgetful functor (e.g. [1]).

3.1 Preliminaries on convex dcpos

Definition 3.1 A *convex dcpo* is a convex set (Def. 2.15) equipped with a dcpo structure such that the functions that constitute its convex structure are Scott-continuous. This yields a category **dConv** of convex dcpos and affine Scott-continuous maps between them.

A simple example of a convex dcpo is the unit interval of the reals.

3.1.1 Sums of convex dcpos

Recall that the sum $A + B$ of two convex sets, A and B, can be described as the set $A \uplus B \uplus (A \times B \times (0, 1))$, where $(0, 1)$ is the open unit interval. Its elements either come directly from A, or from B, or are a non-trivial formal convex combination of elements from A and B. With a slightly informal notation, we write $(a, -, 0)$

instead of a, and $(-, b, 1)$ instead of b. Then define the convex structure as follows

$$\sum_i r_i.(a_i, b_i, \lambda_i) \stackrel{\text{def}}{=} \left(\sum_i \frac{r_i(1 - \lambda_i)}{1 - \sum_i r_i \lambda_i}.a_i, \sum_i \frac{r_i \lambda_i}{\sum_i r_i \lambda_i}.b_i, \left(\sum_i r_i \lambda_i\right)\right)$$

taking the obvious convention where $\left(\sum_i r_i \lambda_i\right)$ is 0 or 1. This has the universal property of the coproduct in the category of convex sets.

3.1.2 Skew sums

There is a variation on the sum that will be useful in what follows. To motivate, observe that if A and B are partial orders then we can form a new partial order $A \mathbin{\rotatebox[origin=c]{180}{$\Subset$}} B$ whose carrier is $A + B$ but with the partial order generated by $a \leq_{A \mathbin{+} B} a'$ whenever $a \leq_A a'$, and $b \leq_{A \mathbin{+} B} b'$ whenever $b \leq_B b'$, and $a \leq_{A \mathbin{+} B} b$ whenever $a \in A$ and $b \in B$. We call this the *skew sum*. It gives a universal square
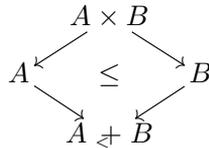
If A and B are convex dcpos then we define a *skew sum* $A \mathbin{+} B$ as the coproduct of convex sets, but with the partial order $(a, b, \lambda) \leq (a', b', \mu)$ if $a \leq a'$ and $b \leq b'$ and $\lambda \leq \mu$. This has a universal property like a coproduct except with an additional requirement that $a \leq b$ for $a \in A$, $b \in B$.

For example, we can freely add a bottom element to a convex dcpo A by taking the skew sum $(1 \mathbin{+} A)$.

3.2 Abstract definitions of quantum domains

Our definition of quantum domain is inspired by Example 2.18. Recall that $\mathbb{N}_{\text{CPU}}$ is the category of natural numbers and where a morphism $m \to n$ is a CPU-map (§ 2.3). A functor $D : \mathbb{N}_{\text{CPU}}^{\text{op}} \to \mathbf{dConv}$ is 'locally affine' if D preserves the convex structure of morphisms, i.e. $D(r.f + s.g) = r.D(f) + s.D(g)$ whenever $r + s = 1$.

Definition 3.2 A *quantum predomain* is a locally affine functor $D : \mathbb{N}_{\text{CPU}}^{\text{op}} \to \mathbf{dConv}$. A *quantum domain* is a locally affine functor $D : \mathbb{N}_{\text{CPU}}^{\text{op}} \to \mathbf{dConv}$ such that the convex dcpo $D(1)$ has a least element.

A morphism of quantum (pre)domains, which will be called a *QD-map*, is a natural transformation $\phi : D \Rightarrow E$ between quantum (pre)domains, i.e. a family of continuous affine maps $\{\phi_n : D(n) \to E(n)\}_{n \in \mathbb{N}}$ such that, for every map $f : n \to m$

in $\mathbb{N}_{\mathrm{CPU}}$, the following diagram commutes:

$$
\begin{array}{ccc}
n & D(n) \xrightarrow{\phi_n} E(n) \\
\downarrow f & D(f)\Big\uparrow \qquad \Big\uparrow E(f) \\
m & D(m) \xrightarrow[\phi_m]{} E(m)
\end{array}
$$

If D is a quantum domain, i.e. $D(1)$ has a least element, then we say that a QD-map $\phi : D \to E$ is *strict* if $\phi_1(\bot_{D(1)})$ is a least element in $E(1)$.

We define **QDom** to be the full subcategory of **QPredom** comprising quantum domains, and **QDom**$_!$ to be the subcategory of **QPredom** comprising quantum domains and strict QD-maps. Those three categories are enriched over the category **Dcpo** of dcpos together with Scott-continuous maps.

For a motivating example, recall (Ex. 2.18) that every W*-algebra A induces a functor $\mathcal{NS}(A) : \mathbb{N}_{\mathrm{CPU}}^{\mathrm{op}} \to \mathbf{Conv}$ with $\mathcal{NS}(A)(n) = \mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}(M_n(A), \mathbb{C})$. This can be understood as a quantum predomain, where each $\mathcal{NS}(A)(n)$ is considered with a discrete order. In particular $\mathcal{NS}(\mathbb{C}) \cong \mathbb{N}_{\mathrm{CPU}}(-, 1)$. This gives an embedding $\mathcal{NS} : \mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}^{\mathrm{op}} \to \mathbf{QPredom}$.

3.3 Construction of quantum domains

3.3.1 Sums of quantum predomains

The coproduct of quantum predomains is defined pointwise. We define the sum of two quantum predomains D and E pointwise: let $(D + E)(n) = D(n) + E(n)$ for $n \in \mathbb{N}$. This has the universal property of the coproduct in **QPredom**.

The embedding $\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}^{\mathrm{op}} \to \mathbf{QPredom}$ preserves sums. This follows from two facts: first, $\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}^{\mathrm{op}}(A \oplus B, \mathbb{C}) \cong \mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}^{\mathrm{op}}(A, \mathbb{C}) + \mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}^{\mathrm{op}}(B, \mathbb{C})$ (e.g. [11, Prop. 16]) and second, $M_n(A \oplus B) \cong M_n(A) \oplus M_n(B)$, for all W*-algebras A and B.

3.3.2 Tensor with quantum data, aka copower

For every quantum predomain D, one can define a quantum predomain $(n \odot D)$ by $(n \odot D)(m) = D(nm)$ for every natural number $n \in \mathbb{N}$ and $(n \odot D)(f) = D(M_n(f))$ for $f : m \to p$ in $\mathbb{N}_{\mathrm{CPU}}^{\mathrm{op}}$. In particular $(2 \odot D)$ can be thought of informally as a predomain of entangled pairs (x, d) where x is a qubit and d is from D.

We can build a functor

$$
\odot : \mathbb{N}_{\mathrm{CPU}}^{\mathrm{op}} \times \mathbf{QPredom} \to \mathbf{QPredom}.
$$

This has the universal property of 'copower by representables' (see e.g. [12]).

3.3.3 Lifting via skew sums

We can also define a pointwise skew-sum of quantum predomains. For quantum predomains D and E, as a quantum predomain $D + E$ with $(D \mathbin{\nleftarrow} E)(n) = D(n) \mathbin{\nleftarrow}$

$E(n)$ for $n \in \mathbb{N}$. This has the universal property of the skew coproduct (§3.1.2) in **QPredom**.

We use the skew coproduct to define a way of lifting quantum predomains to quantum domains. Let $D_\perp = \mathcal{NS}(\mathbb{C}) \mathbin{\skew} D$. In more detail, $D_\perp(n) = \mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}(M_n, \mathbb{C}) \mathbin{\skew} D(n)$. Since $(\mathcal{NS}(\mathbb{C}))(1) = 1$ we know that $D_\perp(1)$ has a least element for any quantum predomain D.

Proposition 3.3 *The construction* $(-)_\perp : \mathbf{QPredom} \to \mathbf{QDom}_!$ *is left adjoint to the forgetful functor* $\mathbf{QDom}_! \to \mathbf{QPredom}$.

$$\mathbf{QDom}_! \xrightleftharpoons[\ (-)_\perp\ ]{\ U\ } \mathbf{QPredom}$$

Moreover the adjunction is **Dcpo**-*enriched.*

Proof. Consider $\delta : D \Rightarrow E$ and $\eta : D_\perp \Rightarrow E$.

Firstly, one can define a strict QD-map $\delta^* : D_\perp \Rightarrow E$ as a natural family of maps $\delta^*(n) : D_\perp(n) = \mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}(M_n, \mathbb{C}) + D(n) \to E(n)$, where $\delta^*(n) : (\varphi, x, \lambda) \mapsto (\varphi, \delta(n)(x), \lambda)$ for every $n \in \mathbb{N}$.

Secondly, one can define a QD-map $\eta_* : D \Rightarrow E$ as a natural family of maps $\eta_*(n) : D(n) \to E(n)$, where $\eta_*(n) : x \mapsto \eta(-, x, 1)$ for every $n \in \mathbb{N}$.

The Scott-continuous maps of hom-sets, $(-)^* : \mathbf{QPredom}(D, U(E)) \to \mathbf{QDom}_!(D_\perp, E)$ and $(-)_* : \mathbf{QDom}_!(D_\perp, E) \to \mathbf{QPredom}(D, U(E))$, are inverse of each other. □

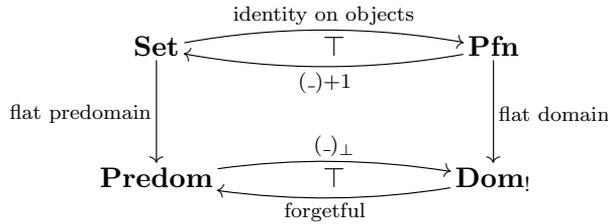## 3.4   *Relationship with the Löwner order and earlier work*

We conclude by relating these steps in quantum domain theory with earlier work on using operator algebra to model quantum computation.

To make an analogy, we recall the basic adjunction between the category **Set** of sets and functions and the category **Pfn** of sets and partial functions.

$$\mathbf{Set} \xrightleftharpoons[\ (\_)+1\ ]{\ \text{identity on objects}\ } \mathbf{Pfn}$$
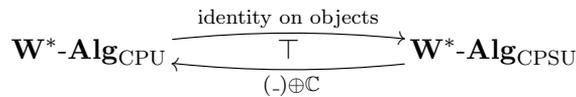
Partial functions can be thought of as first-order computations, and indeed each hom-set forms a dcpo. However, the adjunction is not enriched in dcpos. Thus, although there is a notion of lifting it does not properly capture partiality. The set $A + 1$ captures the notion of 'programs that either return something of type A or diverge', but the order structure associated with partiality is not captured in this

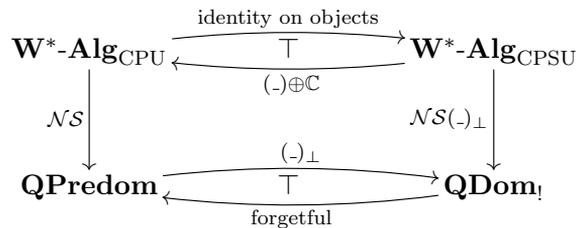set. To remedy this, one embeds this adjunction into one involving domains

$$
\begin{array}{ccc}
\textbf{Set} & \xrightarrow[\quad\text{identity on objects}\quad]{\top} & \textbf{Pfn} \\
& \xleftarrow{(\text{-})+1} & \\
\text{flat predomain}\Big\downarrow & & \Big\downarrow\text{flat domain} \\
& \xrightarrow{(\text{-})_\perp} & \\
\textbf{Predom} & \xleftarrow[\quad\text{forgetful}\quad]{\top} & \textbf{Dom}_!
\end{array}
$$

where the lower adjunction and the right-hand embedding are dcpo-enriched.

Now, we also have an adjunction between $\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPSU}}$ and $\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}}$:

$$
\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}} \;\;
\begin{array}{c}
\xrightarrow{\text{identity on objects}} \\
\quad\top\quad \\
\xleftarrow{(\text{-})\oplus\mathbb{C}}
\end{array}
\;\; \mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPSU}}
$$

which has been proposed for studying quantum computation. The hom-sets of $\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPSU}}$ are dcpos, under the Löwner order: $f \le g$ if $g - f$ is completely positive. Again, however, the adjunction is not dcpo-enriched. This time, the W*-algebra $A \oplus \mathbb{C}$ captures the notion of 'programs that either return something of type A or diverge', but again the order structure associated with partiality is not captured in this algebra. Our proposal for quantum domains resolves this since there is an embedding

$$
\begin{array}{ccc}
\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPU}} & \xrightarrow[\quad\text{identity on objects}\quad]{\top} & \mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPSU}} \\
& \xleftarrow{(\text{-})\oplus\mathbb{C}} & \\
\mathcal{NS}\Big\downarrow & & \Big\downarrow\mathcal{NS}(\text{-})_\perp \\
& \xrightarrow{(\text{-})_\perp} & \\
\textbf{QPredom} & \xleftarrow[\quad\text{forgetful}\quad]{\top} & \textbf{QDom}_!
\end{array}
$$

where the right-hand embedding is dcpo-enriched and so is the lower adjunction.

3.5 Next steps for quantum domain theory

We have demonstrated that our representation techniques can be used to build a quantum domain theory that supports lifting and tensor products by quantum data. We conclude by mentioning some next steps in this direction.

Each quantum predomain of the form $\mathcal{NS}(A)$ has some extra structure. For example, the block diagonal function $A \oplus A \to M_2(A)$ between W*-algebras gives a QD-map $2 \odot \mathcal{NS}(A) \to \mathcal{NS}(A) + \mathcal{NS}(A)$. This can be thought of as measuring a qubit in the standard basis, returning a classical bit. This algebraic structure has been axiomatized in [23]. It seems likely that it would be helpful to require this structure on *all* quantum domains.

Some authors impose additional conditions on d-cones and convex dcpos (see e.g. [13,14]). In this article we are focusing on morphisms, rather than objects, but

we suppose that extra conditions will be relevant in future work on semantics of quantum programs.

In earlier work [19] the first author has shown that the category $\mathbf{W}^*\text{-}\mathbf{Alg}_{\mathrm{CPSU}}$ is algebraically complete, and so supports the solution of recursive domain equations. An important next step is to investigate whether this result extends to quantum domains.

It would also be interesting to investigate connections to other models of higher-order quantum computation [9].

Summary

In the first part of the paper, we have characterized the notion of complete positivity in a natural way. This abstract setting can be used as a way of extending the logical and semantical properties of positive maps to completely positive maps.

In the other half of the paper, we have shown that W*-algebras are order-valued presheaves on a category of qubits, and normal completely positive maps are natural transformations between W*-algebras, seen as order-valued presheaves. We have exposed some of the categorical properties of those presheaves. We argue that our presheaves are a suitable generalization of W*-algebras when it comes to denotational semantics of quantum programs.

Acknowledgment

References

[1] S. Abramsky, A. Jung, "Domain Theory", Handbook of logic in computer science 3, pp. 1-168, 1994.

[2] E.M. Alfsen, F.W. Shultz, "State spaces of operator algebras: basic theory, orientations and C*-products", Birkhäuser, 2001.

[3] K. Cho, "Semantics for a Quantum Programming Language by Operator Algebras", in B. Coecke, I. Hasuo, P. Panangaden, Proceedings of the 11th workshop on Quantum Physics and Logic, EPTCS 172, pp. 165-190, 2014.

[4] E. D'Hondt, Prakash Panangaden. "Quantum weakest preconditions". Mathematical Structures in Computer Science 16(3): 429-451, 2006.

[5] B. J. Day. "Construction of biclosed categories", PhD thesis, University of New South Wales, 1970.

[6] E. G. Effros & Z.-J. Ruan, "Operator spaces", London Mathematical Society monographs 23, Clarendon Press, Oxford, 2000.

[7] R. Furber, B. Jacobs, "From Kleisli categories to commutative C*-algebras: Probabilistic Gelfand Duality", in R. Heckel, S. Milius, Algebra and Coalgebra in Computer Science, pp. 141-157, Springer Berlin Heidelberg.

[8] R. Furber, PhD thesis, forthcoming.

[9] I. Hasuo & N. Hoshino, "Semantics of Higher-Order Quantum Computation via Geometry of Interaction", in 26th Annual IEEE Symposium on Logic in Computer Science (LICS) (pp. 237-246), p. 237-246, IEEE Computer Society, 2011.

[10] B. Jacobs, "New Directions in Categorical Logic, for Classical, Probabilistic and Quantum Logic", arXiv preprint arXiv:1205.3940, 2012.

[11] B. Jacobs, B. Westerbaan, B. Westerbaan, "States of convex sets", in A. Pitts, Foundations of Software Science and Computation Structures, Lecture Notes in Computer Science 9034, pp. 87-101, Springer Berlin Heidelberg, 2015.

[12] G. Janelidze and G.M. Kelly. "A Note on Actions of a Monoidal Category", Theory and Applications of Categories 9(4), pp.61-91, 2001.

[13] C. Jones, G. D. Plotkin, "A probabilistic powerdomain of evaluations", Proceedings of the Fourth Annual Symposium on Logic in computer science, pp.186-195, 1989.

[14] K. Keimel, G. D. Plotkin, "Mixed powerdomains for probability and non-determinism", to appear in Logical Methods in Computer Science, 2015.

[15] O. Malherbe, P. Scott, P. Selinger, "Presheaf models of quantum computation: an outline", in B. Coecke, L. Ong, P. Panangaden, "Computation, Logic, Games, and Quantum Foundations. The Many Facets of Samson Abramsky", Lecture Notes in Computer Science Volume 7860, pp. 178-194, 2013.

[16] M. Pagani, B. Valiron, P. Selinger, "Applying quantitative semantics to higher-order quantum computing", In Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2014), San Diego, ACM SIGPLAN Notices 49(1), pp. 647-658, 2014.

[17] V. Paulsen, "Completely bounded maps and operator algebras", Cambridge Studies in Advanced Mathematics 78, 2003.

[18] M. Rennela, "Operator Algebras in Quantum Computation", Master thesis, 2013.

[19] M. Rennela, "Towards a Quantum Domain Theory: Order-enrichment and Fixpoints in W*-algebras", in B. Jacobs, A. Silva & S. Staton, Proceedings of the 30th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXX), Electronic Notes in Theoretical Computer Science 308, pp. 289-307, 2014.

[20] S. Sakai, "C*-algebras and W*-algebras", Springer-Verlag, 1971.

[21] P. Selinger, "Towards a quantum programming language", Mathematical Structures in Computer Science 14, pp. 527-586, 2004.

[22] P. Selinger, "Dagger compact closed categories and completely positive maps", In P. Selinger, Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005), Chicago, Electronic Notes in Theoretical Computer Science 170, pp.139-163, Elsevier, 2007.

[23] S. Staton. "Algebraic effects, linearity, and quantum programming languages", in Proceedings of 42nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2015), pp. 395-406, 2015.

[24] M. Takesaki, "Theory of operator algebras Vol. 1", Springer, 2002.