

Dagger compact closed categories and completely positive maps (extended abstract)

Peter Selinger¹

*Department of Mathematics and Statistics
Dalhousie University, Halifax, Nova Scotia, Canada*

Abstract

Dagger compact closed categories were recently introduced by Abramsky and Coecke, under the name “strongly compact closed categories”, as an axiomatic framework for quantum mechanics. We present a graphical language for dagger compact closed categories, and sketch a proof of its completeness for equational reasoning. We give a general construction, the CPM construction, which associates to each dagger compact closed category its “category of completely positive maps”, and we show that the resulting category is again dagger compact closed. We apply these ideas to Abramsky and Coecke’s interpretation of quantum protocols, and to D’Hondt and Panangaden’s predicate transformer semantics.

Key words: Categorical model, quantum computing, dagger categories, CPM construction.

1 Introduction

In the last few years, there have been several proposals for axiomatic semantics of quantum programming languages and/or protocols. In [9], I proposed the notion of an “elementary quantum flow chart category” as an abstraction of the category of superoperators. In such a category, one can give a denotational interpretation of first-order quantum programming languages with classical control and finite classical and quantum data types.

At about the same time, Abramsky and Coecke introduced their notion of a “strongly compact closed category” (which will be called “dagger compact closed category” in this paper). This axiomatic framework captures enough of the properties of the category of finite-dimensional Hilbert spaces to be able to express basic quantum-mechanical concepts such as unitary maps, scalars, projectors, inner products etc.

¹ Email: selinger@mathstat.dal.ca. Research supported by NSERC.

The respective axiomatics of [9] and [1] are at the outset quite different: the former attempts to capture von Neumann’s formulation of quantum mechanics in terms of mixed states, density matrices, and completely positive maps, while the latter formalizes the Hilbert space point of view centered on pure states, unitary maps, and projectors.

The aim of this paper is to demonstrate that, despite these obvious differences, the two approaches actually have a lot in common. We show that the category of completely positive maps considered in [9] is actually a biproduct dagger compact closed category. We further introduce a general construction, called the CPM construction, that associates to any dagger compact closed category \mathbf{C} its “category of completely positive maps” $\mathbf{CPM}(\mathbf{C})$. In this way, we obtain a canonical and very general way of passing from a “Hilbert-style” to a “von Neumann-style” setting.

As an application of these ideas, we discuss Abramsky and Coecke’s interpretation of quantum protocols. Abramsky and Coecke have argued that biproducts can be used to model classical information flow in their categorical setting. We show that the proper place for such an interpretation is not the category of Hilbert spaces, but its derived category of completely positive maps.

As another application of the dagger structure on the categories of completely positive maps, we also we also briefly discuss D’Hondt and Panangaden’s predicate transformer semantics [4].

An important tool in reasoning about many variants of monoidal categories is the use of graphical languages [5]. Here, we describe a graphical language for dagger compact closed categories. A similar language has already been used, in a more informal manner, in the papers of Abramsky and Coecke. We go into some detail about this language, and sketch a proof of its completeness for equational reasoning in dagger compact closed categories.

2 Dagger compact closed categories

Dagger compact closed categories were introduced, under the name “strongly compact closed categories”, by Abramsky and Coecke in [1]. They extend compact closed categories [6] with a notion of *adjoint* of a morphism (here the word “adjoint” is used in the sense of linear algebra, not of category theory). Dagger compact closed categories axiomatize many of the structural properties of the category of finite dimensional Hilbert spaces, and therefore they appear to form a suitable framework for axiomatizing quantum mechanics.

2.1 Compact closed categories

Recall that a *symmetric monoidal category* is a category \mathbf{C} together with a bifunctor \otimes , a distinguished object I , and natural isomorphisms $\alpha_{A,B,C} : (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)$, $\lambda_A : A \rightarrow I \otimes A$, and $\sigma_{A,B} : A \otimes B \rightarrow B \otimes A$,

subject to standard coherence conditions [7]. As usual, we also write $\rho_A = \sigma_{I,A} \circ \lambda_A : A \rightarrow A \otimes I$.

A *compact closed category* is a symmetric monoidal category where each object A is assigned a dual object A^* , together with a unit map $\eta_A : I \rightarrow A^* \otimes A$ and a counit map $\epsilon_A : A \otimes A^* \rightarrow I$, such that $\lambda_A^{-1} \circ (\epsilon_A \otimes A) \circ \alpha_{A,A^*,A}^{-1} \circ (A \otimes \eta_A) \circ \rho_A = \text{id}_A$ and $\rho_{A^*}^{-1} \circ (A^* \otimes \epsilon_A) \circ \alpha_{A^*,A,A^*} \circ (\eta_A \otimes A^*) \circ \lambda_A = \text{id}_{A^*}$.

Remark 2.1 In any compact closed category, the operation $(-)^*$ extends to a contravariant functor by mapping $f : A \rightarrow B$ to $f^* : B^* \rightarrow A^*$, defined by $\rho_{A^*}^{-1} \circ (A^* \otimes \epsilon_B) \circ (A^* \otimes f \otimes B^*) \circ (\eta_A \otimes B^*) \circ \lambda_{B^*}$. This functor preserves the symmetric monoidal structure, and possesses a natural isomorphism $A^{**} \cong A$.

2.2 Dagger categories

Definition 2.2 (Dagger category) A *dagger category* is a category \mathbf{C} together with an involutive, identity-on-objects, contravariant functor $\dagger : \mathbf{C} \rightarrow \mathbf{C}$.

Concretely, this means that to every morphism $f : A \rightarrow B$ one associates a morphism $f^\dagger : B \rightarrow A$, called the *adjoint* of f , such that for all $f : A \rightarrow B$ and $g : B \rightarrow C$:

$$\begin{aligned} \text{id}_A^\dagger &= \text{id}_A & : A &\rightarrow A, \\ (g \circ f)^\dagger &= f^\dagger \circ g^\dagger & : C &\rightarrow A, \\ f^{\dagger\dagger} &= f & : A &\rightarrow B, \end{aligned} \tag{1}$$

Definition 2.3 (Unitary map, self-adjoint map) In a dagger category, a morphism $f : A \rightarrow B$ is called *unitary* if it is an isomorphism and $f^{-1} = f^\dagger$. A morphism $f : A \rightarrow A$ is called *self-adjoint* or *hermitian* if $f = f^\dagger$.

2.3 Dagger symmetric monoidal categories

Definition 2.4 (Dagger symmetric monoidal category) A *dagger symmetric monoidal category* is a symmetric monoidal category \mathbf{C} with a dagger structure, such that the contravariant functor $\dagger : \mathbf{C} \rightarrow \mathbf{C}$ coherently preserves the symmetric monoidal structure.

Concretely, the requirement that \dagger ‘‘coherently preserves the symmetric monoidal structure’’ means the following, for all $f : A \rightarrow B$ and $g : C \rightarrow D$:

$$\begin{aligned} (f \otimes g)^\dagger &= f^\dagger \otimes g^\dagger & : B \otimes D &\rightarrow A \otimes C, \\ \alpha_{A,B,C}^\dagger &= \alpha_{A,B,C}^{-1} & : A \otimes (B \otimes C) &\rightarrow (A \otimes B) \otimes C, \\ \lambda_A^\dagger &= \lambda_A^{-1} & : I \otimes A &\rightarrow A, \\ \sigma_{A,B}^\dagger &= \sigma_{A,B}^{-1} & : B \otimes A &\rightarrow A \otimes B. \end{aligned} \tag{2}$$

Remark 2.5 Equivalently, the requirement that \dagger preserves the symmetric monoidal structure “coherently” means that the two natural isomorphisms $A^\dagger \otimes B^\dagger \cong (A \otimes B)^\dagger$ coincide: namely, the one coming from the monoidal functor, and the other one coming from the fact that \dagger is an identity-on-objects functor:

$$\begin{array}{ccc} A^\dagger \otimes B^\dagger & & \\ \text{id} \downarrow & \searrow \cong & \\ A \otimes B & \xrightarrow{\text{id}} & (A \otimes B)^\dagger \end{array}$$

Since the vertical and horizontal morphisms are in fact identities, coherence in this case amounts to the requirement that \dagger preserves the symmetric monoidal structure “on the nose”, rather than up to isomorphism.

2.4 Dagger compact closed categories

Definition 2.6 (Dagger compact closed category [1]) A dagger compact closed category is a dagger symmetric monoidal category that is also compact closed, and such that the following diagram commutes for all A :

$$\begin{array}{ccc} I & \xrightarrow{\epsilon_A^\dagger} & A \otimes A^* \\ & \searrow \eta_A & \downarrow \sigma_{A,A^*} \\ & & A^* \otimes A \end{array} \quad (3)$$

Remark 2.7 Abramsky and Coecke coined the term “strongly compact closed category” for a dagger compact closed category. However, a dagger structure can be added to almost any type of category, and so it is useful to have a terminology that can be extended to these cases. Since it would sound silly, for instance, to speak of a “strong category”, we propose the adjective “dagger” as a more distinctive alternative.

Remark 2.8 In any dagger compact closed category, we have $f^{*\dagger} = f^{\dagger*}$. This is a consequence of equation (3). The converse is not true, i.e., a compact closed dagger category in which $f^{*\dagger} = f^{\dagger*}$ is not necessarily dagger compact closed. To construct a counterexample, take a dagger compact closed category \mathbf{C} in which there is some invertible, but not unitary, scalar $\phi : I \rightarrow I$ (for example, the category of finite dimensional Hilbert spaces, with $\phi(x) = 2x$). Define a new compact closed structure on \mathbf{C} by letting $\eta'_A = \eta_A \circ \phi$ and $\epsilon'_A = \phi^{-1} \circ \epsilon_A$. In the modified compact closed structure, the definition of f^* is unchanged and thus $f^{*\dagger} = f^{\dagger*}$ still holds. However, equation (3) is no longer satisfied.

Definition 2.9 (Lower-star operation) Given $f : A \rightarrow B$ in a dagger compact closed category, we define $f_* : A^* \rightarrow B^*$ by $f_* = f^{*\dagger} = f^{\dagger*}$. This assignment defines a covariant functor.

2.5 Biproducts

Recall the definitions of a zero object and biproducts. An object is called a *zero object* if it is initial and terminal. If $\mathbf{0}$ is a zero object, then there is a distinguished map $A \rightarrow \mathbf{0} \rightarrow B$ between any two objects, denoted $0_{A,B}$. If A_1 and A_2 are objects, their *biproduct* is an object $A_1 \oplus A_2$, together with morphisms $q_i : A_i \rightarrow A_1 \oplus A_2$ and $p_i : A_1 \oplus A_2 \rightarrow A_i$, for $i = 1, 2$, such that the pair p_1, p_2 forms a product cone, the pair q_1, q_2 forms a coproduct cone, and $p_i \circ q_j = \delta_{ij}$. Here $\delta_{ii} = \text{id}_{A_i}$ and $\delta_{ij} = 0_{A_j, A_i}$ when $i \neq j$. We say that \mathbf{C} has *finite biproducts* if it has a zero object $\mathbf{0}$ and a biproduct for any pair of objects.

Recall that a category \mathbf{C} is said to be *enriched in commutative monoids* if each hom-set is equipped with an associative, commutative addition operation with unit 0 , such that composition is linear: $(f + g) \circ h = f \circ h + g \circ h$, $h \circ (f + g) = h \circ f + h \circ g$, $0 \circ h = 0$, and $h \circ 0 = 0$.

If a category \mathbf{C} has finite biproducts, then it carries a canonical such enrichment: Given $f, g : A \rightarrow B$, we can define $f + g : A \rightarrow B$ as $f + g = [\text{id}_B, \text{id}_B] \circ (f \oplus g) \circ \langle \text{id}_A, \text{id}_A \rangle$. The unit for this addition is $0_{A,B}$,

Remark 2.10 We have $q_1 \circ p_1 + q_2 \circ p_2 = \text{id}_{A \oplus B}$.

Lemma 2.11 *Let \mathbf{C}, \mathbf{D} be categories with finite biproducts, and let $F : \mathbf{C} \rightarrow \mathbf{D}$ be a functor. If F has a left or right adjoint (and in particular, if F is an equivalence of categories), then F preserves the biproducts up to isomorphism. Indeed, in this case, the canonical maps*

$$\begin{aligned} [F(q_1), F(q_2)] : F(A) \oplus F(B) &\rightarrow F(A \oplus B) \\ \langle F(p_1), F(p_2) \rangle : F(A \oplus B) &\rightarrow F(A) \oplus F(B) \end{aligned}$$

are mutually inverse. □

Two special cases of this lemma are of particular interest: in a compact closed category with biproducts, the tensor is distributive with respect to the biproducts, i.e., there is a canonical distributivity isomorphism $(A \oplus B) \otimes C \cong A \otimes C \oplus B \otimes C$.

Also, if \mathbf{C} is a dagger category with biproducts, then the functor \dagger automatically preserves biproducts up to isomorphism. However, in general, preservation “up to isomorphism” is not quite sufficient; a more desirable property is that the biproduct should be preserved “on the nose”. The following lemma gives four equivalent conditions that achieve this.

Lemma 2.12 *Let \mathbf{C} be a dagger category with biproducts. Then the following conditions are equivalent. Here $\Delta = \langle \text{id}, \text{id} \rangle : A \rightarrow A \oplus A$, and $\nabla = [\text{id}, \text{id}] : A \oplus A \rightarrow A$.*

- (a) $p_i^\dagger = q_i$, for $i = 1, 2$,
- (b) $(f \oplus g)^\dagger = f^\dagger \oplus g^\dagger$ and $\Delta^\dagger = \nabla$,

(c) $\langle f, g \rangle^\dagger = [f^\dagger, g^\dagger]$,

(d) The following diagram commutes.

$$\begin{array}{ccc}
 A^\dagger \oplus B^\dagger & & \\
 \text{id} \downarrow & \searrow [p_1^\dagger, p_2^\dagger] & \\
 A \oplus B & \xrightarrow{\text{id}} & (A \oplus B)^\dagger
 \end{array}$$

□

Remark 2.13 The diagram in Lemma 2.12(d) can be seen as a coherence diagram, analogous to the diagram in Remark 2.5.

Definition 2.14 (Biproduct dagger compact closed category) A *biproduct dagger compact closed category* is a dagger compact closed category with biproducts, such that $p_i^\dagger = q_i : A_i \rightarrow A_1 \oplus A_2$, for all objects A_1, A_2 and $i = 1, 2$.

2.6 Examples

Example 2.15 The category **Vect** of finite dimensional complex vector spaces and linear maps is a compact closed category with biproducts, but it is not dagger compact closed.

Example 2.16 The category **FdHilb** of finite dimensional Hilbert spaces is biproduct dagger compact closed. The adjoint of $f : A \rightarrow B$ is defined in terms of the inner product; namely, it is the unique map $f^\dagger : B \rightarrow A$ satisfying $\langle fv|w \rangle = \langle v|f^\dagger w \rangle$, for all $v \in A, w \in B$.

Example 2.17 The category **Rel** of sets and relations is biproduct dagger compact closed. In this category, $A \otimes B = A \times B$ is the cartesian product of sets, $A \oplus B = A + B$ is the disjoint union, and $A^* = A$. For a relation $R : A \rightarrow B$, we have $R^* = R^\dagger = \{(y, x) \mid (x, y) \in R\}$ and $R_* = R$.

3 Graphical languages

The graphical languages for symmetric monoidal categories and for compact closed categories are well-known [5,6]. We briefly recall these languages, and then extend them to dagger symmetric monoidal categories and dagger compact closed categories.

3.1 Symmetric monoidal categories: term language

Before describing the *graphical language* of symmetric monoidal categories, it is useful to briefly summarize their *term language*.

We assume given a countable set of *object variables*, denoted α, β, γ etc. The set of *object terms* is freely built from object variables and the unit object term I via the binary operation \otimes . Examples of object terms are $\alpha, I \otimes \alpha, (\alpha \otimes \beta) \otimes \gamma$, and so forth. We use the letters A, B, C to denote object terms.

Further, we assume given a countable set of *morphism variables*, denoted f, g, h etc. We assume that each morphism variable has been assigned a fixed *type* of the form $A \rightarrow B$, and we sometimes write $f^{A \rightarrow B}$ to indicate the type explicitly. We assume that there are countably many morphism variables for each type. We are further given *morphism constants* $\text{id}_A, \alpha_{A,B,C}, \alpha_{A,B,C}^{-1}, \lambda_A, \lambda_A^{-1}, \sigma_{A,B}$, with their usual types, parameterized by object terms A, B, C . The set of typed *morphism terms* is defined recursively: morphism variables and morphism constants are morphism terms, and if $t : A \rightarrow B, s : B \rightarrow C$, and $r : C \rightarrow D$ are morphism terms, then so are $s \circ t : A \rightarrow C$ and $t \otimes s : A \otimes C \rightarrow B \otimes D$.

We also consider equations $s = t$ between morphism terms $s, t : A \rightarrow B$ of matching types. The *equational theory* of symmetric monoidal categories is the set of equations generated by the axioms of symmetric monoidal categories.

Remark 3.1 As is usual in formal languages, one distinguishes the concept of a *variable*, such as α or f , from the concept of a *term*, such as A or t . The symbols “ A ” and “ t ” are, of course, variables of the meta-language. We will make a similar distinction in the graphical language described below.

3.2 Symmetric monoidal categories: graphical language

The graphical language of symmetric monoidal categories was first defined in a mathematically rigorous way by Joyal and Street [5]. Before that, it was used informally for many years by physicists, going back to Penrose [8]. In the formulation that we are going to use, an object variable α is represented by a labeled wire

$$\xrightarrow{\alpha} \quad , \tag{4}$$

A morphism variable $f : \alpha_1 \otimes \alpha_2 \otimes \dots \otimes \alpha_n \rightarrow \beta_1 \otimes \beta_2 \otimes \dots \otimes \beta_m$ is represented by a labeled box

$$\begin{array}{ccc} \xrightarrow{\alpha_n} & \blacksquare & \xrightarrow{\beta_m} \\ \dots & & \dots \\ \xrightarrow{\alpha_2} & f & \xrightarrow{\beta_2} \\ \xrightarrow{\alpha_1} & & \xrightarrow{\beta_1} \end{array} \tag{5}$$

The graphical language associates to each object term A in the language of symmetric monoidal categories a certain wiring, and to each morphism term t a certain diagram, made up from wires (4) and boxes (5). We schematically write

$$\xrightarrow{A} \tag{6}$$

for the representation of an object term A , and

$$\xrightarrow{A} \blacksquare t \xrightarrow{B} \tag{7}$$

for the representation of a morphism term $t : A \rightarrow B$.

(a) Objects

$$\xrightarrow{I} := (\text{empty}) \quad A \otimes B := \begin{array}{c} \xrightarrow{B} \\ \xrightarrow{A} \end{array}$$

(b) Morphisms

$$\begin{array}{c} \xrightarrow{A} \boxed{s \circ t} \xrightarrow{C} \\ \xrightarrow{A} \boxed{t} \xrightarrow{B} \boxed{s} \xrightarrow{C} \end{array} := \quad \xrightarrow{A} \boxed{\text{id}_A} \xrightarrow{A} := \xrightarrow{A}$$

$$\begin{array}{c} \xrightarrow{C} \boxed{t \otimes s} \xrightarrow{D} \\ \xrightarrow{A} \xrightarrow{B} \end{array} := \begin{array}{c} \xrightarrow{C} \boxed{s} \xrightarrow{D} \\ \xrightarrow{A} \boxed{t} \xrightarrow{B} \end{array} \quad \begin{array}{c} \xrightarrow{B} \boxed{\sigma_{A,B}} \xrightarrow{A} \\ \xrightarrow{A} \xrightarrow{B} \end{array} := \begin{array}{c} \xrightarrow{B} \quad \xrightarrow{A} \\ \xrightarrow{A} \quad \xrightarrow{B} \end{array}$$

Table 1

The graphical language of symmetric monoidal categories

Definition 3.2 The wiring (6) is defined by recursion on the object term A as in Table 1(a). Note that the object I is represented by zero wires, i.e., by the empty wiring. The diagram (7) is defined by recursion on the morphism term t as in Table 1(b). In addition to the cases shown in Table 1, the maps $\alpha_{A,B,C}$, λ_A , and their inverses are represented in the same way as the identity morphism.

Remark 3.3 Note that the graphical language strictly speaking only uses the notations (4) and (5), where α and f are variables. The more general notations (6) and (7) belong to the *meta-language*. Therefore, when we speak of properties of diagrams (such as isomorphism) and operations on diagrams (such as composition), we always mean this with respect to the underlying diagram language, and not with respect to the meta-language.

Convention 3.4 When it is convenient to use textual notation, we also write $\llbracket A \rrbracket$ and $\llbracket t \rrbracket$ for the diagram associated to an object term A and a morphism term $t : A \rightarrow B$, respectively. In other words, we write

$$\llbracket A \rrbracket := \xrightarrow{A}$$

$$\llbracket t \rrbracket := \xrightarrow{A} \boxed{t} \xrightarrow{B}$$

Remark 3.5 We have marked each wire with a left-to-right orientation, and each box with a square in the upper-left corner. For now, these markings are useless; their significance will become apparent when we introduce additional markings in the graphical languages of compact closed categories and dagger symmetric monoidal categories below.

Example 3.6 The equation

$$\sigma_{A',B'} \circ (f \otimes g) = (g \otimes f) \circ \sigma_{A,B}$$

of symmetric monoidal categories can be translated into a diagrammatic equation, by translating the left-hand-side and the right-hand-side separately:

The diagrammatic equation shows two equivalent ways to represent the composition of morphisms f and g . On the left, f (input A , output A') and g (input B , output B') are connected by a crossing. On the right, g (input B , output B') and f (input A , output A') are connected by a crossing. The equality is indicated by an equals sign between the two diagrams.

The equation evidently “holds” in the graphical language, in the sense that the graphical representation of the left-hand-side and the right-hand-side are isomorphic as graphs, in an obvious sense with respect to a fixed ordering of the input- and output-wires. For a technically rigorous formalization of graph isomorphism in this context, see [5].

Indeed, one has the following theorem, due to Joyal and Street [5]:

Theorem 3.7 (Graphical language of symmetric monoidal categories) *A well-typed equation between morphisms in the language of symmetric monoidal categories follows from the axioms of symmetric monoidal categories if and only if it holds, up to graph isomorphism, in the graphical language. \square*

3.3 Compact closed categories

The graphical language of compact closed categories extends that of symmetric monoidal categories. Individual wires can now be oriented right-to-left or left-to-right. The interpretation of A^* is a wiring obtained by turning the wiring $\llbracket A \rrbracket$ “upside down”; this means reversing the orientation of each wire, as well as reversing the bottom-up order of the wires. We schematically write $\llbracket A^* \rrbracket$ as $\overleftarrow{\llbracket A \rrbracket}$. More formally, we define the following notations in the meta-language:

$$\begin{aligned} \overleftarrow{\llbracket A \rrbracket} &:= \overleftarrow{\llbracket A \rrbracket} \\ \overleftarrow{\llbracket A \rrbracket} &:= \overleftarrow{\llbracket A \rrbracket} & \llbracket A \otimes B \rrbracket &:= \overleftarrow{\llbracket A \rrbracket} \overleftarrow{\llbracket B \rrbracket} \\ \overleftarrow{\llbracket I \rrbracket} &:= (\text{empty}) \end{aligned}$$

Note that the ordering of the wires in a tensor product for the right-to-left orientation is top-to-bottom, not bottom-to-top. Therefore, $\llbracket A^* \rrbracket$ is $\llbracket A \rrbracket$ turned “upside-down”.

The structural maps $\eta_A : I \rightarrow A^* \otimes A$ and $\epsilon_A : A \otimes A^* \rightarrow I$ are represented as follows:

$$\llbracket \eta_A \rrbracket = \overleftarrow{\llbracket A \rrbracket} \overleftarrow{\llbracket A \rrbracket} \quad \llbracket \epsilon_A \rrbracket = \overleftarrow{\llbracket A \rrbracket} \overleftarrow{\llbracket A \rrbracket}$$

Note that if $t : A \rightarrow B$, then the morphism $t^* : B^* \rightarrow A^*$, which was defined in Remark 2.1, is given as follows:

$$\llbracket t^* \rrbracket = \begin{array}{c} \overleftarrow{B} \\ \hline \begin{array}{c} \overrightarrow{A} \quad \boxed{t} \quad \overrightarrow{B} \\ \hline \overleftarrow{A} \end{array} \end{array}$$

This is effectively the same as “the diagram of t turned upside down”, and we introduce the special graphical notation

$$\llbracket t^* \rrbracket = \overleftarrow{B} \boxed{t} \overleftarrow{A}$$

Since we have marked one corner of the box representing t , it is possible to tell when such a box has been turned upside down. Of course, this notation belongs to the meta-language.

Equations in the language of compact closed categories, such as $\lambda_A^{-1} \circ (\eta_A \otimes A) \circ \alpha_{A,A^*,A}^{-1} \circ (A \otimes \eta_A) \circ \rho_A = \text{id}_A$ translate into graphical equations, such as

$$\begin{array}{c} \overrightarrow{A} \\ \hline \overleftarrow{A} \\ \hline \overrightarrow{A} \end{array} = \overrightarrow{A}$$

With the appropriate notion of graph isomorphism, one has the following theorem, which was implicitly proved by Kelly and Laplaza [6]:

Theorem 3.8 (Graphical language of compact closed categories) *A well-typed equation between morphisms in the language of compact closed categories follows from the axioms of compact closed categories if and only if it holds, up to graph isomorphism, in the graphical language. \square*

3.4 Dagger symmetric monoidal categories

The graphical language of dagger symmetric monoidal categories extends that of symmetric monoidal closed categories. There are no additional operations on objects. If $f : A \rightarrow B$ is a morphism variable

$$\overrightarrow{A} \boxed{f} \overrightarrow{B},$$

then $f^\dagger : B \rightarrow A$ is represented by a new type of box as follows:

$$\llbracket f^\dagger \rrbracket := \overleftarrow{B} \boxed{f} \overleftarrow{A}$$

We think of this as “reflecting f about the y -axis”. Note that the objects A and B have been exchanged, but the orientation of the wires remains unchanged

(in this case, left to right). More generally, if $t : A \rightarrow B$ is a term, then $\llbracket t^\dagger \rrbracket$ is obtained by reflecting the entire diagram $\llbracket t \rrbracket$ about the y -axis. We use the notation:

$$\llbracket t^\dagger \rrbracket = \begin{array}{c} \overleftarrow{B} \\ \hline \boxed{t} \\ \hline \overleftarrow{A} \end{array},$$

We skip the formal recursive definition of “reflecting” and give an example instead:

Example 3.9 (Reflection of a diagram) Suppose $f : A \rightarrow B \otimes C$ and $g : E \rightarrow C \otimes D$, and let

$$t = (\text{id}_B \otimes g^\dagger) \circ \alpha_{B,C,D} \circ (f \otimes \text{id}_D).$$

Then the respective graphical representations of t and t^\dagger are:

$$t = \begin{array}{c} \overleftarrow{D} \\ \hline \overrightarrow{A} \quad \boxed{f} \quad \overrightarrow{C} \quad \boxed{g} \quad \overrightarrow{E} \\ \hline \overrightarrow{B} \end{array}$$

$$t^\dagger = \begin{array}{c} \overrightarrow{E} \quad \boxed{g} \quad \overrightarrow{C} \quad \boxed{f} \quad \overrightarrow{A} \\ \hline \overrightarrow{B} \quad \overleftarrow{D} \end{array}$$

The graphical language of dagger symmetric monoidal categories satisfies the usual coherence theorem:

Theorem 3.10 (Graphical language of dagger symmetric monoidal categories) *A well-typed equation between morphisms in the language of dagger symmetric monoidal categories follows from the axioms of dagger symmetric monoidal categories if and only if it holds, up to graph isomorphism, in the graphical language.*

Proof. Soundness is by induction. For completeness, suppose that the graphs of $s, t : A \rightarrow B$ are isomorphic. Using equations (1) and (2), we can find terms s', t' such that $s = s'$ and $t = t'$ holds in the equational theory of dagger categories, and such that the only applications of \dagger in s', t' are to morphism variables. Now regard each daggered variable α^\dagger as a fresh variable $\tilde{\alpha}$. Since $\llbracket s' \rrbracket$ and $\llbracket t' \rrbracket$ are isomorphic, $s' = t'$ holds in all symmetric monoidal categories by Theorem 3.7. Therefore $s = s' = t' = t$ in the language of dagger symmetric monoidal categories. \square

3.5 Dagger compact closed categories

The graphical language of dagger compact closed categories is obtained by combining the graphical language of compact closed categories with that of dagger symmetric monoidal categories. We also introduce notations for η_A^\dagger and ϵ_A^\dagger :

$$\llbracket \epsilon_A^\dagger \rrbracket = \begin{array}{c} \overrightarrow{A} \\ \hline \overrightarrow{A} \end{array} \quad \llbracket \eta_A^\dagger \rrbracket = \begin{array}{c} \overleftarrow{A} \\ \hline \overleftarrow{A} \end{array}$$

The defining equation of dagger compact closed categories, $\eta_A = \sigma_{A,A^*} \circ \epsilon_A^\dagger$, is then evidently satisfied up to graph isomorphism in the graphical language:

$$\llbracket \eta_A \rrbracket = \begin{array}{c} \text{A} \\ \xrightarrow{\quad} \\ \text{A} \end{array} = \begin{array}{c} \text{A} \quad \text{A} \\ \xrightarrow{\quad} \quad \xrightarrow{\quad} \\ \text{A} \quad \text{A} \end{array} = \llbracket \sigma_{A,A^*} \circ \epsilon_A^\dagger \rrbracket.$$

Theorem 3.11 (Graphical language of dagger compact closed categories) *A well-typed equation between morphisms in the language of dagger compact closed categories follows from the axioms of dagger compact closed categories if and only if it holds, up to graph isomorphism, in the graphical language.*

Proof. The proof is the same as that of Theorem 3.10, only using the additional equation (3) and working relative to Theorem 3.8. \square

Remark 3.12 If $f : A \rightarrow B$, then $f, f^\dagger : B \rightarrow A, f_* : A^* \rightarrow B^*$, and $f^* : B^* \rightarrow A^*$ are graphically represented as follows:

$$\begin{array}{c} f = \begin{array}{c} \text{A} \quad \blacksquare \\ \xrightarrow{\quad} \quad \square \\ \text{B} \end{array} \quad f^\dagger = \begin{array}{c} \text{B} \quad \square \\ \xrightarrow{\quad} \quad \blacksquare \\ \text{A} \end{array} \\ \\ f_* = \begin{array}{c} \text{A} \quad \square \\ \xleftarrow{\quad} \quad \blacksquare \\ \text{B} \end{array} \quad f^* = \begin{array}{c} \text{B} \quad \blacksquare \\ \xleftarrow{\quad} \quad \square \\ \text{A} \end{array} \end{array}$$

Thus, f_* can be viewed graphically as f “reflected about the x -axis”.

3.6 Biproduct dagger compact closed categories

The above approach to graphical languages does not easily generalize to the presence of biproducts. This is due to the presence of two binary operations on objects ($A \otimes B$ and $A \oplus B$). Since one cannot represent both of these operations by juxtaposition of objects, a more structured syntax is needed to represent object terms. An approach to a graphical representation for this language, based on Girard’s proof nets, was recently proposed by Abramsky and Duncan [2]. We will not consider the issue further in this paper.

4 Positivity and the CPM construction

Dagger compact closed categories were introduced by Abramsky and Coecke as an axiomatic basis for quantum mechanics, providing an alternative to (and a generalization of) the usual axiomatization in terms of Hilbert spaces. Concepts such as unitary maps, projections, and even measurements and the Born rule can be formulated in this framework [1]. On the other hand, the present author has argued that a suitable framework for the semantics of quantum programming languages is not the category of Hilbert spaces and linear maps, but a category **CPM** of completely positive operators [9,10]. The advantage of working with completely positive maps is that both quantum

features (such as unitary evolution) and classical probabilistic features (such as the different branches of a measurement) can be modeled internally as morphisms within the same category.

The purpose of this section is to show that (1) the construction of **CPM** from **FdHilb** generalizes to any dagger compact closed category **C** in place of **FdHilb**, and (2) the resulting category of completely positive maps is also dagger compact closed. We call this construction the *CPM construction*.

4.1 Positive maps

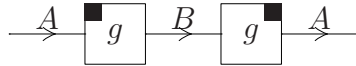
Definition 4.1 (Positive map) A morphism $f : A \rightarrow A$ in a dagger category is called *positive* if there exists an object B and a morphism $g : A \rightarrow B$ such that $f = g^\dagger \circ g$.

Remark 4.2 If $f : A \rightarrow A$ is positive, there exist in general many different decompositions of the form $f = g^\dagger \circ g$. We are only interested in the existence of such a decomposition, not in the particular decomposition.

Example 4.3 In the dagger category **FdHilb** of finite dimensional Hilbert spaces, the positive morphisms are precisely the positive operators, i.e., those that satisfy $\langle fv|v \rangle \geq 0$ for all v .

Example 4.4 In the category **Rel** of sets and relations, the positive morphisms are precisely the relations $R : A \rightarrow A$ that are symmetric and satisfy $xRy \Rightarrow xRx$.

Convention 4.5 In the graphical language, a positive map is a map $f : A \rightarrow A$ that is of the form



for some B and g . By visual abstraction, we use the notation



to denote a generic positive map.

Definition 4.6 (Trace) Given a morphism $f : A \rightarrow A$ in a compact closed category, its *trace* $\text{tr } f : I \rightarrow I$ is defined as

$$\text{tr } f = I \xrightarrow{\sigma \circ \eta_A} A \otimes A^* \xrightarrow{f \otimes A^*} A \otimes A^* \xrightarrow{\epsilon_A} I.$$

With the benefit of the graphical language, the trace of f is defined as follows:

$$\text{tr } f = \text{Diagram of a trace: a box labeled } f \text{ with black squares in the top-left and top-right corners, with an arrow from } A \text{ to the box and an arrow from the box to } A. \text{ This box is enclosed in a rounded rectangle with an arrow labeled } A \text{ pointing from the top to the bottom of the rectangle.} \text{ The entire structure is connected to } I \text{ via the } \sigma \circ \eta_A \text{ and } \epsilon_A \text{ maps.}$$

Lemma 4.7 (Properties of positive maps) *The following properties hold in any biproduct dagger compact closed category:*

- (a) *If f is positive, then $h^\dagger \circ f \circ h$ is positive, for any h .*
- (b) *The identity morphism id_A is positive. Also, if $f : A \rightarrow A$ and $g : B \rightarrow B$ are positive, then so are $f \otimes g : A \otimes B \rightarrow A \otimes B$ and $f \oplus g : A \oplus B \rightarrow A \oplus B$.*
- (c) *The zero morphism $0_{A,A} : A \rightarrow A$ is positive. Also, if $f, g : A \rightarrow A$ are positive, then so is $f + g$.*
- (d) *If f is positive, then $f^\dagger = f$.*
- (e) *If $f : A \rightarrow A$ is positive, then so are $f^* : A^* \rightarrow A^*$ and $\text{tr } f : I \rightarrow I$.*
- (f) *If $f, g : A \rightarrow A$ are positive, then so is $\text{tr}(f \circ g) : I \rightarrow I$.*

Proof. (a)–(e) are straightforward from the definitions and the structure preserved by $(-)^{\dagger}$. For (f), let $g = h^\dagger \circ h$. Then $\text{tr}(f \circ g) = \text{tr}(h \circ f \circ h^\dagger)$, which is positive by (a) and (e). \square

Remark 4.8 The converse of Lemma 4.7(d) does not hold: for instance, the symmetry map $\sigma_{A,A} : A \otimes A \rightarrow A \otimes A$ satisfies $\sigma^\dagger = \sigma$, but is not positive in **Rel** or **FdHilb**.

4.2 Positive matrices

Definition 4.9 (Name, positive matrix) In a compact closed category, the *name* of a morphism $f : A \rightarrow B$ is $\ulcorner f \urcorner : I \rightarrow A^* \otimes B$ defined as follows:

$$\ulcorner f \urcorner = \begin{array}{c} \text{---} A \text{---} \boxed{f} \text{---} B \text{---} \\ \text{---} A \text{---} \end{array}$$

The name of a morphism is also sometimes called its *matrix*, by analogy with vector spaces. A *positive matrix* is a morphism $\ulcorner f \urcorner : I \rightarrow A^* \otimes A$ that is the name of a positive map $f : A \rightarrow A$.

Remark 4.10 In the graphical language, a positive matrix is a morphism of the form

$$\begin{array}{c} \text{---} A \text{---} \boxed{f} \text{---} A \text{---} \\ \text{---} A \text{---} \end{array} = \begin{array}{c} \text{---} A \text{---} \boxed{g} \text{---} B \text{---} \boxed{g} \text{---} A \text{---} \\ \text{---} A \text{---} \end{array} = \begin{array}{c} \text{---} B \text{---} \boxed{k} \text{---} A \text{---} \\ \text{---} B \text{---} \boxed{k} \text{---} A \text{---} \end{array}$$

for some B and $k = g^\dagger$. We therefore use the special notation

$$\begin{array}{c} \text{---} A \text{---} \\ \boxed{h} \\ \text{---} A \text{---} \end{array}$$

for a generic positive matrix.

4.3 Completely positive maps

Definition 4.11 (Completely positive map) Let A, B be objects in a dagger compact closed category. We say that a morphism $f : A^* \otimes A \rightarrow B^* \otimes B$ is *completely positive* if for all objects C and all positive matrices $g : I \rightarrow C^* \otimes A^* \otimes A \otimes C$, the morphism

$$(C^* \otimes f \otimes C) \circ g = \text{Diagram}$$

is a positive matrix.

Lemma 4.12 Let $f : A^* \otimes A \rightarrow B^* \otimes B$. Then f is completely positive if and only if

$$\chi_f := \text{Diagram}$$

is a positive matrix.

Proof. The left-to-right implication is trivial by choosing $C = A^*$ and $g = (\eta_{A^*})_* \otimes \eta_{A^*}$. For the right-to-left implication, assume χ_f is a positive matrix. Then $(C^* \otimes f \otimes C) \circ g$ is equal to the following matrix, which is seen to be positive by the graphical language:

$$(C^* \otimes f \otimes C) \circ g = \text{Diagram}$$

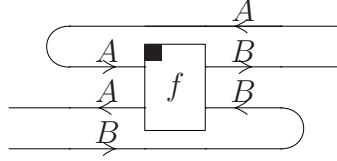
□

Lemma 4.12 is a categorical version of Choi’s Theorem from linear algebra [3]. We also call the matrix χ_f the *characteristic matrix* of f (cf. [9, Sec. 6.7]).

Corollary 4.13 Using Lemma 4.12 and the graphical language, it immediately follows that the following are equivalent:

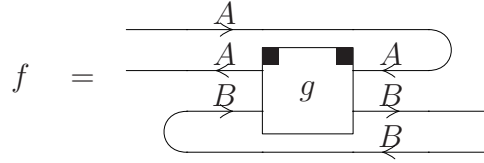
- (a) $f : A^* \otimes A \rightarrow B^* \otimes B$ is completely positive,

(b) the map

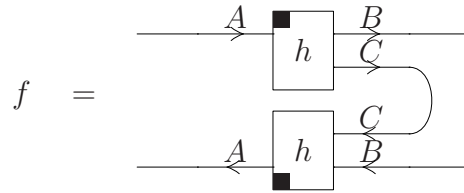


is positive.

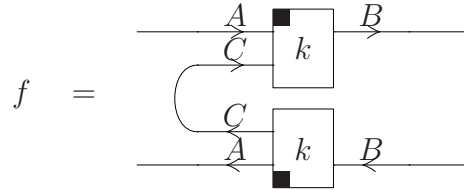
(c) there exists a positive map $g : B \otimes A^* \rightarrow B \otimes A^*$ such that



(d) there exists an object C and a morphism $h : A \rightarrow C \otimes B$ such that

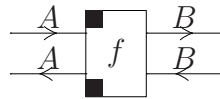


(e) there exists an object C and a morphism $k : C \otimes A \rightarrow B$ such that



□

Convention 4.14 In light of Corollary 4.13(c)–(e), we will use the notation



to denote a generic completely positive map $f : A^* \otimes A \rightarrow B^* \otimes B$.

Remark 4.15 If $f : A^* \otimes A \rightarrow B^* \otimes B$ is completely positive, then $f = f_*$. The converse does not in general hold, cf. Remark 4.8.

Remark 4.16 A positive matrix $g : I \rightarrow A^* \otimes A$ is the same thing as a completely positive map $f : I^* \otimes I \rightarrow A^* \otimes A$ (modulo the isomorphism $I^* \otimes I \cong I$).

Lemma 4.17 (a) The identity map $\text{id} : A^* \otimes A \rightarrow A^* \otimes A$ is completely positive.

(b) If $f : A^* \otimes A \rightarrow B^* \otimes B$ and $g : B^* \otimes B \rightarrow C^* \otimes C$ are completely positive, then so is $g \circ f : A^* \otimes A \rightarrow C^* \otimes C$.

(c) If $f : A^* \otimes A \rightarrow B^* \otimes B$ and $g : C^* \otimes C \rightarrow D^* \otimes D$ are completely positive, then so is

$$\begin{aligned} C^* \otimes A^* \otimes A \otimes C &\xrightarrow{\cong} A^* \otimes A \otimes C^* \otimes C \\ \xrightarrow{f \otimes g} B^* \otimes B \otimes D^* \otimes D &\xrightarrow{\cong} D^* \otimes B^* \otimes B \otimes D. \end{aligned}$$

(d) If $f : A \rightarrow B$ is any morphism, then $f_* \otimes f : A^* \otimes A \rightarrow B^* \otimes B$ is completely positive.

Proof. Immediate, using the graphical language. \square

4.4 The CPM construction

Definition 4.18 (CPM construction) Given a dagger compact closed category \mathbf{C} , we define a new category $\mathbf{CPM}(\mathbf{C})$ whose objects are the same as the objects of \mathbf{C} . A morphism $f : A \rightarrow B$ in $\mathbf{CPM}(\mathbf{C})$ is a completely positive map $f : A^* \otimes A \rightarrow B^* \otimes B$ in \mathbf{C} . Composition of morphisms is as in \mathbf{C} .

Remark 4.19 By Lemma 4.17(a) and (b), $\mathbf{CPM}(\mathbf{C})$ is indeed a category. Moreover, Lemma 4.17(d) yields a functor $F : \mathbf{C} \rightarrow \mathbf{CPM}(\mathbf{C})$, defined via $F(A) = A$ and $F(f) = f_* \otimes f$.

Theorem 4.20 $\mathbf{CPM}(\mathbf{C})$ is again dagger compact closed. The tensor product on objects is inherited from \mathbf{C} ; on morphisms it is given by Lemma 4.17(c). The structural maps $\alpha_{A,B,C}$, λ_A , $\sigma_{A,B}$, η_A and ϵ_A are all given by the images of the respective maps of \mathbf{C} under F . If $f : A^* \otimes A \rightarrow B^* \otimes B$, then f^\dagger in $\mathbf{CPM}(\mathbf{C})$ is given by $f^\dagger : B^* \otimes B \rightarrow A^* \otimes A$ in \mathbf{C} . The functor $F : \mathbf{C} \rightarrow \mathbf{CPM}(\mathbf{C})$ preserves the dagger compact closed structure.

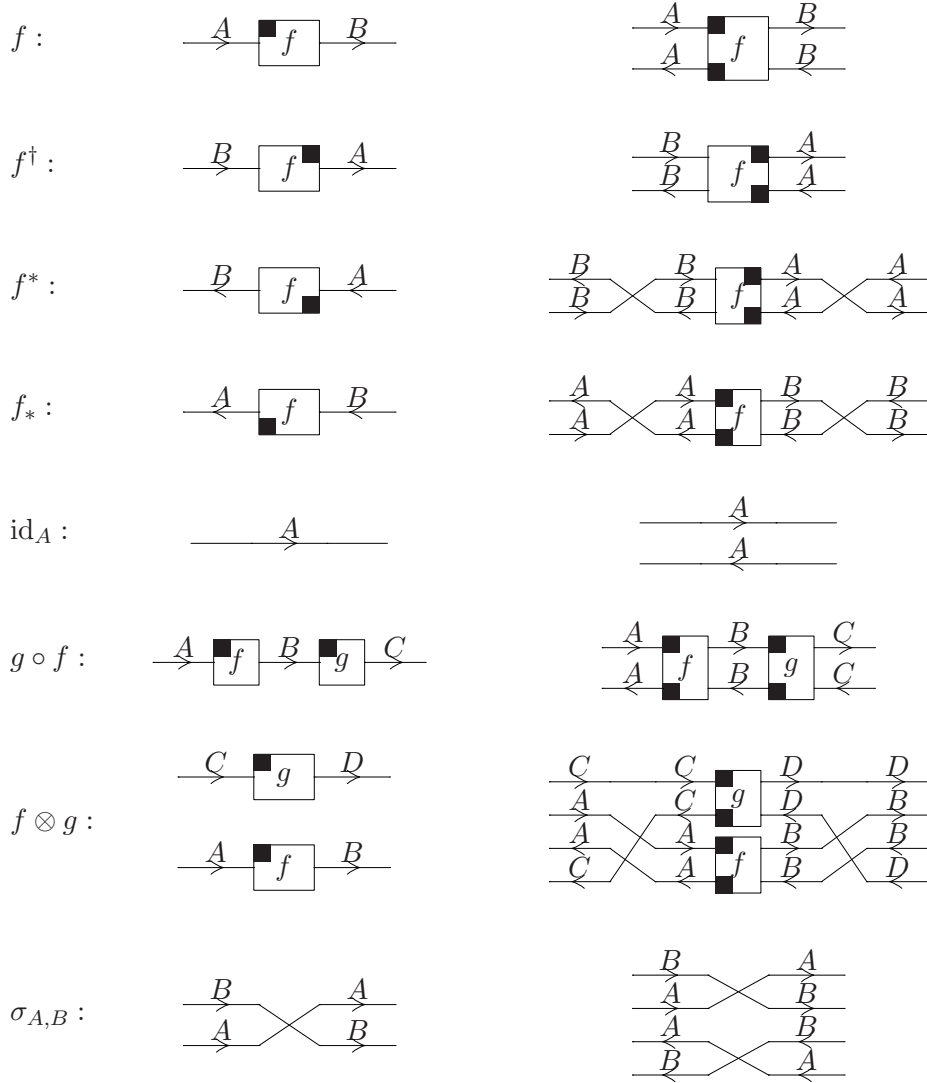
Proof. The required equations are easy to verify. To aid the task, Table 2 gives a translation of the graphical language of $\mathbf{CPM}(\mathbf{C})$ into that of \mathbf{C} . We prove one equation as an example: to prove that

holds in $\mathbf{CPM}(\mathbf{C})$, we must prove the following in \mathbf{C} :

This is evidently true by graph isomorphism. The proof of the other equations is similar. \square

In the graphical language of $\mathbf{CPM}(\mathbf{C})$:

In the graphical language of \mathbf{C} :



and similarly for all constants $c \in \{\text{id}_A, \alpha_{A,B,C}, \alpha_{A,B,C}^{-1}, \lambda_A, \lambda_A^{-1}, \sigma_{A,B}, \eta_A, \epsilon_A\}$:

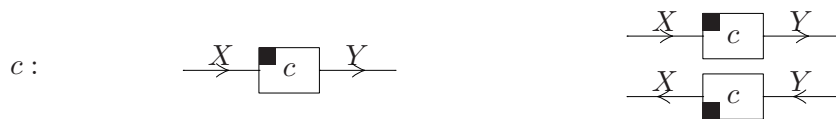


Table 2
The CPM construction in the graphical language

Example 4.21 The category $\mathbf{CPM}(\mathbf{FdHilb})$ has as its objects finite dimensional Hilbert spaces, and as its morphisms completely positive maps $f : A^* \otimes A \rightarrow B^* \otimes B$. It is the same as the full subcategory of simple objects of the category \mathbf{W} of [9]. See also Example 5.4 below. We note that the canonical functor $F : \mathbf{FdHilb} \rightarrow \mathbf{CPM}(\mathbf{FdHilb})$ is “almost” faithful: indeed, if $f, g : A \rightarrow B$ are two morphisms of \mathbf{FdHilb} , then $F(f) = F(g)$ if and only if there exists a unit scalar ϕ such that $f = \phi g$. In other words, the functor F identifies those morphisms that differ only by a global change of phase.

5 Biproducts and the CPM construction

5.1 Biproduct completion of a dagger compact closed category

Let \mathbf{C} be a category that is enriched over commutative monoids (see Section 2.5). Recall that one can construct the biproduct completion \mathbf{C}^\oplus of \mathbf{C} as follows (cf. [7], Exercise VIII.2.6). The objects of \mathbf{C}^\oplus are finite tuples $\langle A_1, \dots, A_n \rangle$ of objects of \mathbf{C} , where $n \geq 0$. Morphisms are given by matrices, and composition by matrix multiplication. Then \mathbf{C}^\oplus is a category with biproducts. Further, the singleton functor $F(A) = \langle A \rangle$ is an embedding $\mathbf{C} \rightarrow \mathbf{C}^\oplus$.

Recall moreover that if \mathbf{C} is compact closed, and if the tensor operation is linear (i.e., $f \otimes (g + g') = f \otimes g + f \otimes g'$ and $f \otimes 0 = 0$), then \mathbf{C}^\oplus is also compact closed.

The construction of \mathbf{C}^\oplus extends to dagger compact closed categories:

Proposition 5.1 *Suppose \mathbf{C} is a dagger compact closed category, enriched in commutative monoids as a compact closed category. Further suppose that the dagger functor is linear, i.e., $(g + g')^\dagger = g^\dagger + g'^\dagger$ and $0^\dagger = 0$. Then \mathbf{C}^\oplus is a biproduct dagger compact closed category.*

Proof. The adjoints of \mathbf{C}^\oplus are defined by matrix transposition and taking the adjoint of each matrix component. \square

5.2 The CPM construction with biproducts

Suppose \mathbf{C} is a biproduct dagger compact closed category. In general, the category $\mathbf{CPM}(\mathbf{C})$ does not have biproducts. However, as the following lemma shows, $\mathbf{CPM}(\mathbf{C})$ is monoid-enriched. Therefore, we can construct its biproduct completion $\mathbf{CPM}(\mathbf{C})^\oplus$.

Lemma 5.2 *In a biproduct dagger compact closed category, if $f, g : A^* \otimes A \rightarrow B^* \otimes B$ are completely positive, then so are $f + g$ and $0 : A^* \otimes A \rightarrow B^* \otimes B$. \square*

Corollary 5.3 *Let \mathbf{C} be a biproduct dagger compact closed category. Then $\mathbf{CPM}(\mathbf{C})$ is enriched in commutative monoids as a dagger compact closed category. Therefore, its biproduct completion $\mathbf{CPM}(\mathbf{C})^\oplus$ is well-defined. \square*

We call the construction of $\mathbf{CPM}(\mathbf{C})^\oplus$ from \mathbf{C} the *CPM construction with biproducts*. Note that the canonical functor $\mathbf{C} \rightarrow \mathbf{CPM}(\mathbf{C})^\oplus$ preserves the dagger compact closed structure, but does *not* preserve the biproducts. The biproducts of $\mathbf{CPM}(\mathbf{C})^\oplus$ do not resemble those of \mathbf{C} .

Example 5.4 Applying the CPM construction with biproducts to $\mathbf{C} = \mathbf{FdHilb}$, we obtain the category $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$, which is identical to the category \mathbf{W} of [9, Remark 6.9]. Concretely, the objects are finite tuples of finite dimensional Hilbert spaces, and the morphisms are matrices of completely positive operators. Some special objects of the category $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$ are $\mathbf{qbit} = \langle \mathbb{C}^2 \rangle$, the type of quantum bits, and $\mathbf{bit} = \langle \mathbb{C}, \mathbb{C} \rangle$, the type of classical bits. Of course, we have $\mathbf{bit} \cong I \oplus I$, where I is the tensor unit, and $\mathbf{bit} \not\cong \mathbf{qbit}$. Composite data types can be constructed by means of the operations \otimes and \oplus .

6 Applications

6.1 Completely positive maps and the interpretation of quantum protocols

In [1], Abramsky and Coecke argue that the flow of classical information within a quantum protocol, such as the classical information that is the outcome of a measurement operation, can be modeled via the biproducts in the category \mathbf{FdHilb} of finite dimensional Hilbert spaces.

In \mathbf{FdHilb} , the tensor unit I is just the base field \mathbb{C} . Abramsky and Coecke define the type of quantum bits as $\mathbf{qbit} := I \oplus I = \mathbb{C}^2$. Initialization $base : I \oplus I \rightarrow \mathbf{qbit}$ and destructive measurement $meas : \mathbf{qbit} \rightarrow I \oplus I$ are both modeled as isomorphisms; in fact as the identity function. It follows that the composition

$$\mathbf{qbit} \xrightarrow{meas} I \oplus I \xrightarrow{base} \mathbf{qbit} \tag{8}$$

is equal to the identity map. This clearly contradicts physical reality, where the outcome of such an experiment would actually be a mixed state. Abramsky and Coecke circumvent this problem by not allowing compositions such as the above to occur in the interpretation of any quantum experiment.

On the other hand, in the category $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$ (see Example 5.4), the initialization and measurement functions $base$ and $meas$ are not isomorphisms, but are given, relative to the canonical bases of $\mathbf{qbit}^* \otimes \mathbf{qbit}$ and $I \oplus I$, by

$$meas \begin{pmatrix} a & b \\ c & d \end{pmatrix} = (a, d), \quad base(a, d) = \begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix}.$$

We see that the composition is *not* the identity, but correctly maps a pure state (a rank 1 positive matrix) to a possibly mixed state (a positive matrix of rank ≥ 2).

Recall that the category \mathbf{FdHilb} is “embedded” inside $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$ (up to a collapse of global phases, see Example 4.21). Therefore, any construc-

tion that can be done in \mathbf{FdHilb} can also be done within $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$. In fact, since the embedding $F : \mathbf{FdHilb} \rightarrow \mathbf{CPM}(\mathbf{FdHilb})^\oplus$ is structure preserving, any such constructions will stay entirely within \mathbf{FdHilb} as long as the biproducts are not used.

However, as we have already seen, the biproducts in \mathbf{FdHilb} do not at all resemble those of $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$. In \mathbf{FdHilb} , we have $I \oplus I = \mathbf{qbit}$, whereas in $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$, we have $I \oplus I = \mathbf{bit}$. Informally speaking, the scalars of \mathbf{FdHilb} correspond to *amplitudes*, whereas the scalars of $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$ correspond to *probabilities*. Since the functor $F : \mathbf{FdHilb} \rightarrow \mathbf{CPM}(\mathbf{FdHilb})^\oplus$ maps a linear function f to the completely positive map $f_* \otimes f$, it actually carries out the “squaring” operation that maps amplitudes to probabilities. Thus, the fact that the CPM construction does not preserve biproducts can be seen as a categorification of the inequality $|x + y|^2 \neq |x|^2 + |y|^2$.

Note that the axiomatic framework of Abramsky and Coecke, while originally developed to describe the category \mathbf{FdHilb} , applies almost without change to the category $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$. Thus, it is trivial to adapt e.g. the interpretation of quantum protocols to this new setting. Only one small change is necessary: one should drop the condition of [1, Sec. 8.2] that the canonical map $I \oplus I \rightarrow \mathbf{qbit}$ is an isomorphism, and should instead require only a retraction:

$$\begin{array}{ccc} I \oplus I & \xrightarrow{\text{base}} & \mathbf{qbit} \\ & \searrow \text{id} & \downarrow \text{meas} \\ & & I \oplus I \end{array}$$

6.2 Dagger categories and predicate transformer semantics

We briefly discuss another application of dagger categories. As it turns out, the duality embodied by the dagger operation can be rediscovered in a programming language setting as a duality between “denotational semantics” and “weakest precondition semantics”.

In [9], I described a simple first-order quantum programming language QPL. This language was given a straightforward denotational semantics in the category $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$ as follows. The types of the language are built from basic types I , \mathbf{qbit} , and \mathbf{bit} via the type constructors \otimes and \oplus , and they are interpreted as objects of the category $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$ as in Example 5.4 above. Further, each program $P : A \rightarrow B$ is interpreted as a morphism in the category $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$:

$$\llbracket P \rrbracket : A \rightarrow B \tag{9}$$

D’Hondt and Panangaden gave a weakest precondition semantics for the same programming language, using the idea of predicate transformers [4]. In summary, types are again interpreted as objects in $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$, just like in the denotational semantics. D’Hondt and Panangaden define a *predicate* at a type A to be an element of the object A (more precisely, a morphism

$I \rightarrow A$). They then associate to each program $P : A \rightarrow B$ a *predicate transformer* $\text{wp}(P)$, which is a function that associates to each predicate at type B a predicate at type A called its “weakest precondition”. Thus, we have:

$$\text{wp}(P) : B \rightarrow A \tag{10}$$

Both interpretations $\llbracket P \rrbracket$ and $\text{wp}(P)$ are defined by induction on the program P , starting from simple base cases, and using the categorical structure. It is interesting to note that neither of these interpretations explicitly relies on the dagger structure of the category $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$. However, we find that the dagger structure plays a crucial role in *relating* the two semantics: indeed, they are adjoint to each other in the sense that

$$\text{wp}(P) = \llbracket P \rrbracket^\dagger$$

for all programs P . Since both semantics are compositional, one can prove this by induction on P . It immediately follows that the two semantics are equivalent, i.e., for two programs $P, Q : A \rightarrow B$, we have $\llbracket P \rrbracket = \llbracket Q \rrbracket$ if and only if $\text{wp}(P) = \text{wp}(Q)$.

Remark 6.1 Strictly speaking, the target of the original interpretation $\llbracket P \rrbracket$ of [9] is not the category $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$ of completely positive maps, but its subcategory \mathbf{Q} of *superoperators*. Superoperators are completely positive maps satisfying an additional trace condition. Similarly, the target of the original interpretation $\text{wp}(P)$ of [4] is not $\mathbf{CPM}(\mathbf{FdHilb})^\oplus$, but rather a certain subcategory of *healthy* maps. As expected, superoperators are precisely the \dagger -duals of healthy maps.

7 Conclusion

We have shown that Abramsky and Coecke’s framework of dagger (“strongly”) compact closed categories applies to the world of completely positive maps. With the CPM construction, we have provided a general way of constructing a category of completely positive maps out of any dagger compact closed category. This construction corresponds to the passage from “amplitudes” to “probabilities”, or from “purely quantum” to “mixed quantum and classical” systems. The fact that dagger compact closed categories provide a single axiomatic basis for both these settings is remarkable, and shows that dagger compact closed categories are indeed a flexible and unifying framework for the semantics of quantum programming languages and protocols.

References

- [1] Abramsky, S. and B. Coecke, *A categorical semantics of quantum protocols*, in: *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, LICS 2004*, IEEE Computer Society Press, 2004, pp. 415–425.

- [2] Abramsky, S. and R. Duncan, *A categorical quantum logic*, in: *Proceedings of the 2nd International Workshop on Quantum Programming Languages*, TUCS General Publication No 33, Turku Centre for Computer Science, 2004, pp. 3–20.
- [3] Choi, M.-D., *Completely positive linear maps on complex matrices*, *Linear Algebra and its Applications* **10** (1975), pp. 285–290.
- [4] D’Hondt, E. and P. Panangaden, *Quantum weakest preconditions*, in: *Proceedings of the 2nd International Workshop on Quantum Programming Languages*, TUCS General Publication No 33, Turku Centre for Computer Science, 2004, pp. 75–90.
- [5] Joyal, A. and R. Street, *The geometry of tensor calculus I*, *Advances in Mathematics* **88** (1991), pp. 55–112.
- [6] Kelly, G. M. and M. L. Laplaza, *Coherence for compact closed categories*, *Journal of Pure and Applied Algebra* **19** (1980), pp. 193–213.
- [7] Mac Lane, S., “Categories for the Working Mathematician,” Springer Graduate Texts in Mathematics 5, 1971.
- [8] Penrose, R., *Applications of negative dimensional tensors*, in: D. J. A. Welsh, editor, *Combinatorial Mathematics and its Applications*, Academic Press, New York, 1971 pp. 221–244.
- [9] Selinger, P., *Towards a quantum programming language*, *Mathematical Structures in Computer Science* **14** (2004), pp. 527–586.
- [10] Selinger, P., *Towards a semantics for higher-order quantum computation*, in: *Proceedings of the 2nd International Workshop on Quantum Programming Languages*, TUCS General Publication No 33, Turku Centre for Computer Science, 2004, pp. 127–143.