

Completeness of Graphical Languages for Mixed States Quantum Mechanics

Titouan Carette, Simon Perdrix, Renaud Vilmart, Emmanuel Jeandel

► **To cite this version:**

Titouan Carette, Simon Perdrix, Renaud Vilmart, Emmanuel Jeandel. Completeness of Graphical Languages for Mixed States Quantum Mechanics. ICALP, 2019, Patras, Greece. 10.4230/LIPIcs.ICALP.2019.108 . hal-02025720

HAL Id: hal-02025720

<https://hal.archives-ouvertes.fr/hal-02025720>

Submitted on 19 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Completeness of Graphical Languages for Mixed States Quantum Mechanics

Titouan Carette

Emmanuel Jeandel

Simon Perdrix

Renaud Vilmart

{firstname.name}@loria.fr

Université de Lorraine, CNRS, Inria, LORIA, F 54000 Nancy, France

Abstract. There exist several graphical languages for quantum information processing, like quantum circuits, ZX-Calculus, ZW-Calculus, etc. Each of these languages forms a \dagger -symmetric monoidal category (\dagger -SMC) and comes with an interpretation functor to the \dagger -SMC of (finite dimension) Hilbert spaces. In the recent years, one of the main achievements of the categorical approach to quantum mechanics has been to provide several equational theories for most of these graphical languages, making them complete for various fragments of pure quantum mechanics.

We address the question of the extension of these languages beyond pure quantum mechanics, in order to reason on mixed states and general quantum operations, i.e. completely positive maps. Intuitively, such an extension relies on the axiomatisation of a *discard* map which allows one to get rid of a quantum system, operation which is not allowed in pure quantum mechanics.

We introduce a new construction, the *discard construction*, which transforms any \dagger -symmetric monoidal category into a symmetric monoidal category equipped with a discard map. Roughly speaking this construction consists in making any isometry causal.

Using this construction we provide an extension for several graphical languages that we prove to be complete for general quantum operations. However this construction fails for some fringe cases like the Clifford+T quantum mechanics, as the category does not have enough isometries.

1 Introduction

Graphical languages that speak of quantum information can be formalised through the notion of symmetric monoidal categories. Hence, it has a nice graphical representation using string diagrams [39]. Qubits are represented by wires, and morphisms by graphical elements where some wires go in, and some others go out, just as in quantum circuits (which is actually a particular case of symmetric monoidal category), and where these graphical elements can be composed either in sequence (usual composition) or in parallel (tensor product). They usually come with an additional structure, a contravariant functor called dagger.

Examples of graphical languages for quantum mechanics and quantum computing are the quantum circuits and the ZX-Calculus [10]. Some variants of the ZX-calculus have been introduced more recently like the ZW-calculus [24] and the ZH-calculus [6]. All these languages are defined using generators (elementary gates) and come with an interpretation functor which associates with any diagram a pure quantum evolution, i.e. a morphism in the category of Hilbert spaces. Given a graphical language, there are generally several ways to represent a quantum evolution, thus a graphical language is also equipped with an equational theory which allows to transform a diagram into another equivalent diagram. A fundamental property, generally hard to prove, is the completeness of the language: given two diagrams representing the same quantum evolution, one can be turned into the other using only the transformation rules in the theory.

The languages considered have usually been built so as to be able to represent any pure quantum evolution. In this case, the language is called universal for pure quantum mechanics. The hardness of the completeness problem, as well as constraints given by the complexity to physically achieve some gates, focused the research on some restrictions of the languages. On the one hand, finite presentations for the quantum circuits were shown to be complete for some restrictions – namely Clifford [40], one-qubit Clifford+T [35], two-qubit Clifford+T [41], CNot-dihedral [1] –, however none of these restrictions is universal, nor approximately universal. Regarding the ZX-calculus,

completeness results exist for non-universal restrictions of the ZX-Calculus [3,4,15,23], but also for the many-qubit Clifford+T ZX-Calculus [29], which was the first completeness result for an approximately universal fragment of the language. Then complete theories have been introduced for the universal ZX-Calculus [26,30,31,42] and ZW-Calculus [25,26]. The completeness of the graphical languages for pure quantum mechanics is one of the main achievements of the categorical approach to quantum mechanics, and is the cornerstone for the application of this formalism in many areas of quantum information processing. The ZX-Calculus already proved to be useful for quantum information processing [13] (e.g. measurement-based quantum computing [17,22,27], quantum codes [8,16,19,21], circuit optimisation [20], foundations [5,18] ...). Moreover the ZX-calculus can be concretely used through two softwares: Quantomatic [34] and PyZX [32].

The existence of complete graphical languages beyond pure quantum mechanics for more general, not necessarily pure, quantum evolutions is an open question that we address in the present paper.

While pure quantum evolutions correspond to linear maps over Hilbert spaces, probability distributions over quantum states as well as some quantum evolutions like discarding a quantum system can be represented, following the van Neumann approach, by means of density matrices and completely positive maps. The category of completely positive maps has been already studied [37], and in particular the connections between the pure and the van Neumann approaches is a central question in categorical quantum mechanics. Selinger introduced a construction called CPM to turn a category for pure quantum mechanics into a category for density matrices and completely positive maps [38]. Another approach to relate pure quantum mechanics to the general one is the notion of environment structure [9,11,14]. The notion of *purification* is central in the definition of environment structure. The CPM-construction and the environment structure approaches have been proved to be equivalent [11].

In terms of graphical languages, the environment structure approach cannot be used in a straightforward way to extend a graphical language beyond pure quantum mechanics. Roughly speaking the environment structure approach provides second order axioms which associates with any equation on arbitrary (non necessarily pure) evolutions an equivalent equation on pure evolutions. Such a second order axiom cannot be easily handled by a equational theory on diagrams. Regarding the CPM-construction, the main property which has been exploited in [13] is that $\text{CPM}(\mathbf{C})$ is essentially a subcategory of \mathbf{C} , thus one can use a graphical language which has been designed for \mathbf{C} in order to represent morphisms in $\text{CPM}(\mathbf{C})$: Given a complete graphical language for \mathbf{C} , we can use a subset of the pure diagrams to represent the evolutions in $\text{CPM}(\mathbf{C})$. The main caveat of this approach is that this subset is not necessarily closed under the equational theory on pure diagrams, and as a consequence does not provide a complete graphical language for $\text{CPM}(\mathbf{C})$.

Our contributions. We introduce a new construction, the *discard construction*, which transforms any \dagger -symmetric monoidal category into a symmetric monoidal category equipped with a discard map. Roughly speaking this construction consists in making any isometry causal. Indeed, in quantum mechanics, the isometries (linear maps U such $U^\dagger \circ U = I$) are known to be causal, i.e. applying U and then discard the subsystem on which it has been applied is equivalent to discarding the subsystem straightaway. Concretely, the discard construction proceeds as follows: first the discard is added to the subcategory of isometries, making the unit of the tensor a terminal object in this sub-category, as pointed out in [28]. Then the discard construction is obtained as the pushout of the resulting category and the initial one.

We show that the discard construction does not always produce an environment structure for the original category, and thus is not equivalent to the CPM construction. We show that a necessary and sufficient condition for the two constructions to be equivalent is that the initial category has enough isometries. We show that most of the categories usually used in the context of the categorical quantum mechanics, like **FHilb** and **Stab**, do have enough isometries, however **Clifford+T** does not.

Finally, we show that the discard construction provide a simple recipe to extend graphical languages beyond pur quantum mechanics. We provide an extension for several graphical languages that we prove to be complete for general quantum operations.

Structure of the paper. In section 2, we review some categorical notions used in categorical quantum mechanics. Section 3 is dedicated to the definition of the discard construction and the

relation with the CPM construction. Finally, in section 4 we use the discard construction to extend the ZX-calculus to make it complete for general (not necessarily pure) quantum evolutions. The construction is also applied to other graphical languages.

2 Background

2.1 Dagger symmetric monoidal categories

To avoid any size issue, all our categories are small, the homset of a category \mathbf{C} will be denoted $\mathbf{C}[A, B]$. Recall a *strict symmetric monoidal category* (SMC) \mathbf{C} is a category together with a tensor product bifunctor $\otimes : \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$, a unit object I such that $A \otimes I = I \otimes A = A$ and $A \otimes (B \otimes C) = (A \otimes B) \otimes C$, and a symmetry natural isomorphism: $\sigma_{A,B} : A \otimes B \rightarrow B \otimes A$ satisfying $\sigma_{A,I} = 1_A$, $\sigma_{A,B \otimes C} = (1_B \otimes \sigma_{A,C}) \circ (\sigma_{A,B} \otimes 1_C)$, and $\sigma_{A,B} \circ \sigma_{B,A} = 1_{B \otimes A}$. A *prop* is an SMC which set of objects is freely spanned by one object. There is an associated notion of strict symmetric monoidal functor $F : \mathbf{C} \rightarrow \mathbf{D}$ which preserves unit, tensors and symmetries.

We will use *string diagram* notations for SMC where morphisms are described as boxes and

$$g \circ f := \begin{array}{|c|} \hline f \\ \hline g \\ \hline \end{array} \quad f \otimes g := \begin{array}{|c|} \hline f \\ \hline g \\ \hline \end{array} \quad 1_A := |^A \quad 1_I := \begin{array}{|c|} \hline \\ \hline \\ \hline \end{array} \quad \sigma_{A,B} := \begin{array}{c} \diagup \quad \diagdown \\ \diagdown \quad \diagup \end{array}$$

A \dagger -SMC \mathbf{C} , is an SMC with an i.o.o. (identity on object) involutive and contravariant SMC-functor $(\cdot)^\dagger : \mathbf{C} \rightarrow \mathbf{C}$. That is, every morphism $f : A \rightarrow B$ has a dagger $f^\dagger : B \rightarrow A$ such that $f^{\dagger\dagger} = f$, moreover the dagger respects the symmetries $\sigma_{A,B}^\dagger = \sigma_{B,A}$. The dagger is a central notion in categorical quantum computing and can be used to define specific properties of morphisms:

Definition 1. $f : A \rightarrow B$ is an isometry if $f^\dagger \circ f = 1_A$, i.e. $\begin{array}{|c|} \hline f \\ \hline f^\dagger \\ \hline \end{array} = |^A$.

In this paper most of the categories considered are furthermore compact closed: A dagger compact category (\dagger -CC) is a \dagger -SMC where every object A has a dual object A^* such that for all objects A , there are two morphisms $A \cup_{A^*} : A \otimes A^* \rightarrow I$ and $A^* \cap_A : I \rightarrow A^* \otimes A$ satisfying $A \cup_{A^*} \cap_A = |^A$ and $A^* \cap_A \cup_{A^*} = |^{A^*}$ and $(A \cup_{A^*})^\dagger = A^* \cap_A$.

2.2 Examples

We are considering two kinds of SMCs in this paper: the categories of quantum evolutions and the graphical languages.

Quantum evolutions. Pure quantum evolutions correspond the category of Hilbert spaces. We will consider various subcategories of it: **FHilb** is the category of finite dimensional Hilbert spaces which objects are \mathbb{C}^n and morphisms are linear maps. Its tensor is the usual tensor product of vector spaces and its dagger is the adjoint with respect to the usual scalar product. It is the mathematical model for pure quantum mechanics. In quantum information processing, the quantum data are usually carried by qubits, hence **Qubit** is the full subcategory of **FHilb** with objects of the form \mathbb{C}^{2^n} . **Stab** is the sub-category of **Qubit** which is finitely generated by the Clifford operators: H, S, CNot, the state $|0\rangle$, the projector $\langle 0|$, and the scalars 2 and i where:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad \text{CNot} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \langle 0| = (1 \ 0)$$

Those are amongst the most commonly used gates in quantum computation see [36] for details. **Clifford+T** is the same as **Stab** but with the additional generator $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$, the morphisms of **Clifford+T** are exactly the matrices with entries in the ring $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$ [29]. Contrary to **Stab**, **Clifford+T** is *approximately universal* in the sense that $\forall n, m \in \mathbb{N}, \forall f \in \mathbf{Qubit}[\mathbb{C}^{2^n}, \mathbb{C}^{2^m}]$ and $\forall \epsilon > 0$, there exists $g \in \mathbf{Clifford+T}[\mathbb{C}^{2^n}, \mathbb{C}^{2^m}]$ such that $\|f - g\| < \epsilon$. **FHilb**, **Qubit**, **Clifford+T**, and **Stab** are all \dagger -CC. Notice that **Qubit**, **Clifford+T**, and **Stab** are props, but **FHilb** is not.

Probability distributions over pure quantum states as well as some quantum evolutions like discarding a quantum system are not *pure* but can be represented, following the van Neumann

approach, by means of density matrices and completely positive maps. Let **CPM** be the category of finite dimension completely positive maps which objects are \mathbb{C}^n and $\mathbf{CPM}[\mathbb{C}^n, \mathbb{C}^m] = \{U : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{m \times m} \mid U \text{ is a completely positive linear map}\}$. Similarly to the pure case, one can define various subcategories of **CPM**. Notice it can be achieved by the CPM construction described in the next section.

Graphical languages. The second kind of categories we are considering in this paper are graphical languages. They are props which come with interpretation functors defining their semantics. A prop is in fact the equivalent of Lawvere theories for symmetric monoidal theories. They can be presented by generators and relations as one would do for usual theories, see [43] and [7] for a detailed discussion.

Definition 2. A graphical language \mathcal{G} is a prop presented by a set of generators Σ and a set of equations E together with a function $\llbracket \cdot \rrbracket : \Sigma \rightarrow \text{hom}(S)$ called the interpretation of \mathcal{G} in S . \mathcal{G} is said to be sound if $\llbracket \cdot \rrbracket$ defines an interpretation functor $\llbracket \cdot \rrbracket : \mathcal{G} \rightarrow S$, and universal (resp. complete) when this functor is surjective (resp. faithful).

The ZX-, ZW- and ZH-calculi or the quantum circuits are examples of such categories with semantics in **Qubit**.

2.3 Environment structures and CPM-construction

Connecting the Hilbert approach – for pure quantum mechanics – and the van Neumann approach – for open systems – is a central question in categorical quantum mechanics. Selinger pointed out that any \dagger -CC for pure quantum mechanics can be turned into a category for density matrices and completely positive maps via the CPM construction [38]:

Definition 3. Given a \dagger -CC \mathbf{C} , let $\mathbf{CPM}(\mathbf{C})$ be the \dagger -CC with the same objects as \mathbf{C} such that

$$\mathbf{CPM}(\mathbf{C})[A, B] = \left\{ \begin{array}{c} \begin{array}{ccc} A & & A^* \\ \downarrow & & \downarrow \\ \boxed{f} & & \boxed{f^*} \\ \downarrow & & \downarrow \\ B & C & C^* & B^* \end{array} \\ , f \in \mathbf{C}[A, B \otimes C] \end{array} \right\}, \text{ where } \boxed{g^*} := \begin{array}{c} A^* \\ \downarrow \\ \begin{array}{c} \boxed{g^\dagger} \\ \downarrow \\ A \end{array} \\ \downarrow \\ B^* \end{array}.$$

Applying it to **FHilb** one obtains the category **CPM** of completely positives maps. The CPM construction can also be applied to **Qubit**, **Clifford+T**, and **Stab**. Notice that the CPM-construction has been then extended to the non necessarily compact categories $\llbracket \cdot \rrbracket$.

Another approach to relate pure quantum mechanics to the general one is the notion of environment structure [9,11,14]. The notion of *purification* is central in the definition of environment structure. Intuitively, it means that (1) there is a discard morphism for every object; (2) any morphism can be purified, i.e. decomposed into a pure morphism followed by a discarding map, and (3) this purification is essentially unique. More formally:

Definition 4. An environment structure for a \dagger -CC \mathbf{C} is an CC $\overline{\mathbf{C}}$ with the same objects as \mathbf{C} , an i.o.o SMC-functor $\iota : \mathbf{C} \rightarrow \overline{\mathbf{C}}$ and for each object A a morphism $\underline{\iota}_A : A \rightarrow I$ such that:

- (1) $\underline{\iota}_I = 1_I$, and for all $A, B : \mathbf{C}$, $\underline{\iota}_A \otimes \underline{\iota}_B = \underline{\iota}_{A \otimes B}$.
- (2) For all $f : A \rightarrow B$ in $\overline{\mathbf{C}}$, there is an $f' : A \rightarrow B \otimes X$ in \mathbf{C} such that: $\begin{array}{c} \boxed{f} \\ \downarrow \\ \underline{\iota}_B \end{array} = \begin{array}{c} \boxed{\iota(f')} \\ \downarrow \\ \underline{\iota}_{B \otimes X} \end{array}$
- (3) For any $f : A \rightarrow B \otimes X$ and $g : A \rightarrow B \otimes Y$ in \mathbf{C} : $f \sim_{\text{cp}} g \Leftrightarrow \begin{array}{c} \boxed{\iota(f)} \\ \downarrow \\ \underline{\iota}_{B \otimes X} \end{array} = \begin{array}{c} \boxed{\iota(g)} \\ \downarrow \\ \underline{\iota}_{B \otimes Y} \end{array}$

where the relation \sim_{cp} is defined as: $f \sim_{\text{cp}} g \Leftrightarrow \begin{array}{c} \boxed{f} \\ \downarrow \\ \boxed{f^\dagger} \end{array} = \begin{array}{c} \boxed{g} \\ \downarrow \\ \boxed{g^\dagger} \end{array}$

Notice that \sim_{cp} is technically not a relation on morphisms but on tuples (A, B, X, f) with $f \in \mathbf{C}[A, B \otimes X]$: $(A, B, X, f) \sim_{\text{cp}} (C, D, Y, g)$ if $A = C, B = D$ and f, g satisfy the graphical condition represented above. As an abuse of notation, we write $f \sim_{\text{cp}} g$, as the other components of the tuple will be usually obvious from context. We will do the same for our relation \sim_{iso} below.

CPM is actually an environment structure for the category **FHilb**, and more generally for any \dagger -CC \mathbf{C} , $\text{CPM}(\mathbf{C})$ is an environment structure for \mathbf{C} and conversely any environment structure for \mathbf{C} is equivalent to $\text{CPM}(\mathbf{C})$ [11]. Actually one can notice that $\text{CPM}(\mathbf{C})[A, B]$ is nothing but the set of equivalent classes of \sim_{cp} .

The notion of environment structures has also been generalised to the non compact case [11]. We chose here to focus on the compact case.

3 The Discard Construction

We introduce a new construction, the discard construction which consists in adding a discard map for every object of a \dagger -SMC, and thus intuitively transforming a category for pure quantum mechanics into a category for general quantum evolutions.

Causality is a central notion in quantum mechanics which has been axiomatised using a discard map as follows [33]: $f : A \rightarrow B$ is *causal* if and only if $\begin{array}{c} \boxed{f} \\ \perp \\ \perp \end{array} = \underline{\underline{\perp}}$. Among the pure quantum evolutions, the isometries are causal evolutions. The discard construction essentially consists in making any isometry causal. Thus, whereas the CPM construction relies on completely positive maps and the environment structures on the concept of purification, the discard construction relies on causality.

3.1 Definition

We introduce the new construction in three steps. First, given a \dagger -SMC, one can consider its subcategory of isometries:

Definition 5. *Given a \dagger -SMC \mathbf{C} , \mathbf{C}_{iso} is the subcategory with the same object as \mathbf{C} and isometries as morphisms, i.e. for all $A, B : \mathbf{C}$, $\mathbf{C}_{\text{iso}}[A, B] = \{f : \mathbf{C}[A, B], f^\dagger \circ f = 1_A\}$.*

Notice that \mathbf{C}_{iso} is a SMC but usually not a \dagger -SMC. Any \dagger -SMC-functor $F : \mathbf{C} \rightarrow \mathbf{D}$ between two \dagger -SMC can be restricted to their subcategories of isometries leading to an SMC-functor $F_{\text{iso}} : \mathbf{C}_{\text{iso}} \rightarrow \mathbf{D}_{\text{iso}}$. Thus there is a restriction functor $\text{iso} : \dagger\text{-SMC} \rightarrow \text{SMC}$. Remark that this functor preserves fullness and faithfulness. One always has an inclusion i.o.o. faithful SMC-functor: $i_{\text{iso}} : \mathbf{C} \rightarrow \mathbf{C}_{\text{iso}}$.

In quantum mechanics, isometries are causal evolutions, i.e. applying an isometry and then discarding all outputs is equivalent to discarding the inputs straight away. As pointed out in [28], adding discard maps to the category of isometries would make I a terminal object. Such a category is said to be *affine symmetric monoidal category* (ASMC). We define the affine completion of an SMC:

Definition 6. *Given an SMC \mathbf{C} , we define $\mathbf{C}^!$ as \mathbf{C} with an additional morphism $!_A : A \rightarrow I$ for each object $A : \mathbf{C}$, such that, for all $f : \mathbf{C}_{\text{iso}}[A, B]$, $!_B \circ f = !_A$. This makes I a terminal object in $\mathbf{C}^!$, and then $\mathbf{C}^!$ is an ASMC.*

Again given a functor $F : \mathbf{C} \rightarrow \mathbf{D}$, one can define a functor $F^! : \mathbf{C}^! \rightarrow \mathbf{D}^!$ by $F^!(!_A) = !_i^!(F(A))$ and $F^!(f) = i^!(F(f))$ for the other morphisms. In [28], Huot and Staton show that **CPTPM**, the category of completely positive trace preserving maps, is equivalent to $\mathbf{FHilb}_{\text{iso}}^!$, thus giving a characterisation of it via a universal property. We extend this idea to non-trace preserving maps by proceeding to a local affine completion of the subcategory of isometries.

We define the category \mathbf{C}^\cong as the pushout of \mathbf{C} and $\mathbf{C}_{\text{iso}}^!$:

Definition 7. *Given a \dagger -SMC \mathbf{C} , \mathbf{C}^\cong is defined as the pushout:*

$$\begin{array}{ccc}
\mathbf{C}_{\text{iso}} & \xrightarrow{i_{\text{iso}}} & \mathbf{C} \\
i^! \downarrow & & \downarrow \iota_{\mathbf{C}} \\
\mathbf{C}_{\text{iso}}^! & \xrightarrow{\iota_{\mathbf{C}_{\text{iso}}^!}} & \mathbf{C}^{\neq}
\end{array}$$

Classical results on enriched categories show that the pushout of two SMCs always exists. As all our functors are i.o.o, we can also describe it simply combinatorially. The objects of \mathbf{C}^{\neq} are the same as \mathbf{C} . Its morphisms are equivalence classes generated by formal composition and tensoring of morphisms in $\mathbf{C}_{\text{iso}}^!$ and \mathbf{C} . The equivalence relation is generated by the equations of both categories augmented with equations $i^!(f) = i_{\text{iso}}(f)$ for all f in \mathbf{C}_{iso} . The functors $\iota_{\mathbf{C}}$ and $\iota_{\mathbf{C}_{\text{iso}}^!}$ are the natural ways to embed \mathbf{C} and $\mathbf{C}_{\text{iso}}^!$. We will see those formal compositions as string diagrams whose components are morphisms of \mathbf{C} and $\mathbf{C}_{\text{iso}}^!$ wired to each others. Two diagrams represent the same morphism if we can rewrite one into the other applying the equations of both categories and $i^!(f) = i_{\text{iso}}(f)$ for all f in \mathbf{C}_{iso} . This forms a well defined SMC.

Since the only morphisms in \mathbf{C}_{iso} which are not identified with the morphisms of \mathbf{C} are those that contain $!_A$, we can see \mathbf{C}^{\neq} as \mathbf{C} augmented with discard maps which delete isometries.

Definition 8. The discard map on an object A is defined in \mathbf{C}^{\neq} by $\underline{\underline{!_A}} := \iota_{\mathbf{C}_{\text{iso}}^!}(!_A)$.

Notice, that for any isometry $f : A \rightarrow B$ in \mathbf{C}^{\neq} , $\underline{\underline{f}} = \underline{\underline{!_A}}$, thus any isometry is causal.

3.2 Relation to environment structures and CPM

In order to compare the \mathbf{C}^{\neq} construction with environment structures and the CPM construction we need to study in details the purification process in \mathbf{C}^{\neq} . First notice that any morphism of \mathbf{C}^{\neq} admits a purification:

Lemma 1. Let \mathbf{C} be a \dagger -SMC, For all $f : \mathbf{C}^{\neq}[A, B]$, there is an $X : \mathbf{C}$ and an $f' : \mathbf{C}[A, B \otimes X]$ such that $\underline{\underline{f}} = \underline{\underline{\iota_{\mathbf{C}}(f')}}$.

The proof is in appendix at page 13.

The purification needs not be unique, however it satisfies an essential uniqueness condition. To state it we define the relation \sim_{iso} :

Definition 9. Let \mathbf{C} be a \dagger -SMC, and two morphisms $f : A \rightarrow B \otimes X$, $g : A \rightarrow B \otimes Y$, $f \sim_{\text{iso}} g$ if there are two isometries $u : X \rightarrow Z$ and $v : Y \rightarrow Z$, such that $\underline{\underline{f}} = \underline{\underline{g \circ v \circ u}}$.

Notice that the relation \sim_{iso} is not transitive, thus we consider \sim_{iso}^+ its transitive closure to make it an equivalence relation. It is easy to show that if $f \sim_{\text{iso}}^+ g$ then f and g purify the same morphism of \mathbf{C}^{\neq} . The converse is also true:

Lemma 2. For all $f : A \rightarrow B \otimes X$ and $g : A \rightarrow B \otimes Y$: $f \sim_{\text{iso}}^+ g \Leftrightarrow \underline{\underline{\iota_{\mathbf{C}}(f)}} = \underline{\underline{\iota_{\mathbf{C}}(g)}}$

The proof is in appendix at page 13.

So the purification is unique up to \sim_{iso}^+ . Lemma 2 also gives an alternative definition of \mathbf{C}^{\neq} which relates more easily to the CPM construction. It is the same construction as CPM with \sim_{cp} replaced by \sim_{iso}^+ . In other words $\mathbf{C}^{\neq}[A, B]$ is the set of equivalent classes of \sim_{iso}^+ .

As we have introduced a new discard construction, a natural question is whether \mathbf{C}^{\neq} is an environment structure for \mathbf{C} . To be an environment structure, three conditions are required. The first two are satisfied: \mathbf{C}^{\neq} has a discard morphism for every object, and every morphism can

be purified. The third one is the uniqueness of the purification: according to the definition of the environment structures, f and g purify the same morphism if and only if $f \sim_{\text{cp}} g$ whereas according to Lemma 2, f and g purify the same morphism if and only if $f \sim_{\text{iso}}^+ g$. As a consequence \mathbf{C}^\sharp is an environment structure for \mathbf{C} if and only if $\sim_{\text{cp}} = \sim_{\text{iso}}^+$. It turns out that one of the inclusions is always true:

Lemma 3. *For any \dagger -SMC category \mathbf{C} , we have $\sim_{\text{iso}}^+ \subseteq \sim_{\text{cp}}$.*

The proof is in appendix at page 14.

As a consequence, if $\sim_{\text{cp}} \neq \sim_{\text{iso}}^+$, it means that there are some morphisms f, g that are equal in \sim_{cp} but cannot be proved equal in \sim_{iso}^+ . Intuitively it means the category has not enough isometries to prove those terms equal, which leads to the following definition:

Definition 10. *A \dagger -SMC category \mathbf{C} has enough isometries if the equivalences relations \sim_{cp} and \sim_{iso}^+ of \mathbf{C} are equal.*

Lemma 4. *Given a \dagger -SMC \mathbf{C} , the following properties are equivalent:*

1. \mathbf{C} has enough isometries;
2. \mathbf{C}^\sharp is an environment structure for \mathbf{C} ;
3. $\mathbf{C}^\sharp \simeq \text{CPM}(\mathbf{C})$.

The proof is in appendix at page 15.

Notice that if \mathbf{C} has enough isometries, the discard construction provides a definition of $\text{CPM}(\mathbf{C})$ via a universal property. This gives a more direct way to built the environment, avoiding to deal with the equivalence classes of the CPM construction.

Remark 1. Let's focus for a moment on the category $\text{Causal CPM}(\mathbf{C})$ of causal maps, that is the subcategory of maps cancelled by the discards in $\text{CPM}(\mathbf{C})$. We have that: $\sim_{\text{cp}} \subseteq \sim_{\text{iso}}^+ \Rightarrow \mathbf{C}_{\text{iso}}^! \simeq \text{Causal CPM}(\mathbf{C})$. In fact by Lemma 4, $\text{CPM}(\mathbf{C}) \simeq \mathbf{C}^\sharp$, and then the subcategory $\text{Causal CPM}(\mathbf{C})$ is equivalent to the subcategory of maps cancelled by the discards in \mathbf{C}^\sharp which is equivalent to $\mathbf{C}_{\text{iso}}^!$. $\text{Causal CPM}(\mathbf{FHilb})$ being exactly \mathbf{CPTPM} , we have recovered the result of [28].

3.3 Examples

We consider the usual subcategories of \mathbf{FHilb} used for pure quantum mechanics and show in each case whether the discard construction produces an environment structure or not. First of all, thanks to the Stinespring dilation theorem, \mathbf{FHilb}^\sharp is not only an environment structure for \mathbf{FHilb} , but the relation \sim_{iso} is also transitive in this case:

Proposition 1. *\mathbf{FHilb}^\sharp is an environment structure for \mathbf{FHilb} . Furthermore $\sim_{\text{iso}}^+ = \sim_{\text{iso}}$.*

The proof is in appendix at page 15.

When dealing with graphical languages we will be more interested in the full subcategory \mathbf{Qubit} of \mathbf{FHilb} :

Proposition 2. *\mathbf{Qubit}^\sharp is an environment structure for \mathbf{Qubit} .*

The proof is in appendix at page 15.

Notice that in general, the property of having enough isometries does not transfer to full subcategories: If \mathbf{D} is a full subcategory of \mathbf{C} , we might have $f \sim_{\text{iso}}^+ g$ on \mathbf{C} but $f \not\sim_{\text{iso}}^+ g$ on \mathbf{D} . This could happen for two reasons: First the chain of intermediate morphisms that prove that $f \sim_{\text{iso}}^+ g$ might live outside of \mathbf{D} . Second, the isometries that “prove” that $f \sim_{\text{iso}}^+ g$ on \mathbf{C} might have codomain outside of \mathbf{D} .

If our category is not a full subcategory, then all hell breaks loose, and finding conditions that guarantees that \mathbf{C}^\sharp is an environment structure for \mathbf{C} is not easy.

For subcategories of \mathbf{Qubit} , necessary conditions can be given. This category has the peculiarity that \cdot^* is the identity on object and that $f^{**} = f$ for all morphisms (\cdot^* maps a matrix

to its conjugate matrix). In particular, for any state $\phi : I \rightarrow I \otimes X$, we have $\phi^* \sim_{\text{cp}} \phi$. Indeed

$$\begin{array}{c} \boxed{\phi} \quad \boxed{\phi^*} \\ \text{---} \quad \text{---} \\ \text{---} \quad \text{---} \end{array} = \begin{array}{c} \boxed{\phi^*} \quad \boxed{\phi} \\ \text{---} \quad \text{---} \\ \text{---} \quad \text{---} \end{array} .$$

So a necessary condition for a subcategory of **Qubit** to behave nicely is that for all states ϕ , we have $\phi^* \sim_{\text{iso}}^+ \phi$. This is the case in **Stab**: Given a stabilizer state ϕ , there always exists a stabilizable unitary U s.t. $U\phi = \phi^*$. In fact:

Proposition 3. Stab^{\neq} is an environment structure for **Stab**.

The proof is in appendix at page 15.

The main idea of the proof is to use the map/state duality, and structural results about bipartite stabilizer states [2].

No such unitary exist in general in **Clifford+T**: For almost all states ϕ , there are no unitary U (and even no morphism at all) s.t. $U\phi = \phi^*$. **Clifford+T** therefore has not enough isometries:

Proposition 4. $(\text{Clifford+T})^{\neq}$ is not an environment structure for **Clifford+T**. More precisely, there exists a state ϕ s.t. $\phi \sim_{\text{cp}} \phi^*$ but $\phi \not\sim_{\text{iso}}^+ \phi^*$. One can take for example $\phi = 1 + 2i$ (in this case ϕ is a state with no input and outputs, hence a scalar).

The proof is in appendix at page 16.

4 Application to the ZX-Calculus and other graphical languages

We now focus on the behavior of interpretation functors with respect to the discard construction. The discard construction defines a functor $(_)^{\neq} : \dagger\text{-SMC} \rightarrow \text{SMC}$. Indeed, given a \dagger -SMC functor F, F_{iso} and $F_{\text{iso}}^!$ uniquely define a functor F^{\neq} by pushout.

$$\begin{array}{ccccc} & & \mathbf{D}_{\text{iso}} & \xrightarrow{\quad} & \mathbf{D} \\ & \nearrow F_{\text{iso}} & \downarrow & \nearrow F & \downarrow \\ \mathbf{C}_{\text{iso}} & \xrightarrow{\quad} & \mathbf{C} & & \\ & \searrow F_{\text{iso}}^! & \downarrow & \searrow & \downarrow \\ & & \mathbf{D}_{\text{iso}}^! & \xrightarrow{\quad} & \mathbf{D}^{\neq} \\ & \downarrow & \downarrow & \downarrow & \downarrow \\ \mathbf{C}_{\text{iso}}^! & \xrightarrow{\quad} & \mathbf{C}^{\neq} & & \end{array}$$

The following lemma and theorem are the main tools to apply the discard construction to graphical languages:

Lemma 5. If F is faithful and if $F_{\text{iso}} : \mathbf{C}_{\text{iso}} \rightarrow \mathbf{D}_{\text{iso}}$ is surjective, then $F(f) \sim_{\text{iso}}^+ F(g) \Rightarrow f \sim_{\text{iso}}^+ g$.

The proof is in appendix at page 16.

Theorem 1. Let \mathbf{C} and \mathbf{D} be two \dagger -SMCs and $F : \mathbf{C} \rightarrow \mathbf{D}$ a \dagger -SMC-functor. If F is faithful and if $F_{\text{iso}} : \mathbf{C}_{\text{iso}} \rightarrow \mathbf{D}_{\text{iso}}$ is surjective, then $F^{\neq} : \mathbf{C}^{\neq} \rightarrow \mathbf{D}^{\neq}$ is faithful. If furthermore F is surjective then F^{\neq} is surjective and faithful.

The proof is in appendix at page 17.

Notice that the hypothesis on F_{iso} is very strong, as it makes it an isomorphism: We want it to be surjective as we do not want to lose even one isometry. In particular we do not know if the theorem still applies if F is merely an equivalence of category.

Reformulating for graphical languages this gives:

Corollary 1 (of Theorem 1). Given a \dagger -CC \mathbf{C} with enough isometries, if \mathcal{G} is a \dagger -CC universal complete graphical language for \mathbf{C} then \mathcal{G}^{\neq} is a universal complete language for $\text{CPM}(\mathbf{C})$.

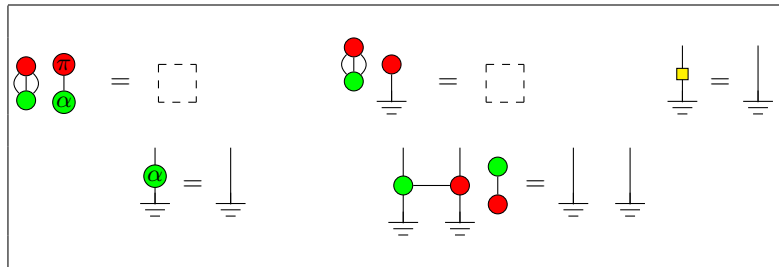
For any $n, m \geq 0$ and $\alpha \in \mathbb{R}$:

$$\left[\begin{array}{c} \dots \\ n \\ \text{red circle} \\ \dots \\ m \end{array} \right] = \left[\begin{array}{c} \dots \\ \text{yellow square} \\ \dots \end{array} \right]^{\otimes m} \circ \left[\begin{array}{c} \dots \\ n \\ \text{green circle} \\ \dots \\ m \end{array} \right] \circ \left[\begin{array}{c} \dots \\ \text{yellow square} \\ \dots \end{array} \right]^{\otimes n}$$

(where $M^{\otimes 0} = (1)$ and $M^{\otimes k} = M \otimes M^{\otimes k-1}$ for $k \in \mathbb{N}^*$).

Theorem 1 provides a recipe for transforming the language for mixed states and CPMs. The resulting language ZX^{\neq} can be seen as a prop with the generators of the ZX -Calculus, augmented with \perp and with the axiomatisation enriched with $\{\perp \circ D = \perp \mid D^\dagger \circ D = I\}$. We actually do not need an infinite axiomatisation. Indeed, the set of isometries of the ZX -Calculus can be finitely generated.

Using $(e^{i\alpha}, |0\rangle, H, R_Z(\alpha), \text{CNot})$ as spanning set of the isometries [36], we obtain only five axioms:



4.2 The $\frac{\pi}{2}$ fragment of ZX -calculus

The $ZX_{\frac{\pi}{2}}$ is obtained from ZX by restricting phases α to $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. It is universal and complete for **Stab** [3] with the axiomatisation provided in Figure 4 in appendix. Moreover according to Lemma 3 **Stab** $^{\neq}$ is an environment structure for **Stab**.

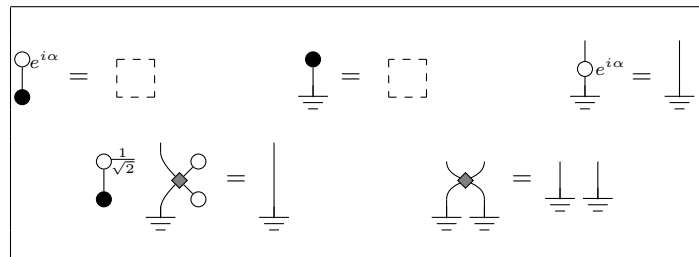
The set $(e^{i\alpha}, |0\rangle, H, R_Z(\alpha), \text{CNot})$, with α restricted to multiples of $\frac{\pi}{2}$, remains a spanning set of isometries in **Stab**, so adding the same set of equations than in ZX^{\neq} will provide a complete axiomatisation for $ZX_{\frac{\pi}{2}}^{\neq}$.

4.3 The Clifford+T fragment of ZX -calculus

Restricting ZX to angles multiples of $\pi/4$, we obtain a languages which is known to be universal and complete for **Clifford+T** [29]. However, as shown by Lemma 4, the semantic category **Clifford+T** does not have enough isometries. The discard construction is strictly coarser than CPM for this fragment. So we leave open the complete axiomatisation of quantum operations for this fragment.

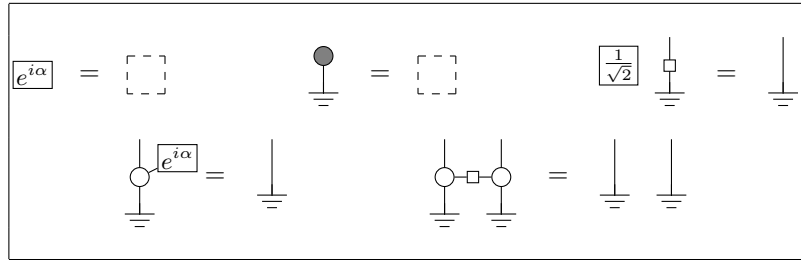
4.4 The ZW -calculus

The ZW -Calculus was introduced in [24], deriving from the GHZ/W-Calculus [12], where the main two generators are two non-equivalent ways to entangle three qubits, the so-called GHZ and W states. The language was made complete for pure quantum mechanics in [26]. The generators, rules and interpretation of the calculus are given in the appendix at page 17. Since CNot is hard to express in this calculus, we choose another set of universal diagrams, more suited to ZW , namely $(e^{i\alpha}, |1\rangle, R_Z(\alpha), H, \text{CZ} \circ \text{SWAP})$. The resulting rules for ZW^{\neq} are:



4.5 The ZH-Calculus

The ZH-Calculus was introduced and proved to be complete in [6]. A presentation of the language is given in appendix at page 18. The point of this language is to easily represent hypergraph-states, a generalisation of graph-states, a useful resource for quantum computing. This language has been specifically designed to easily represent the multi-controlled Z (which constitute the hyperedges in the hypergraph-states). So in particular, CZ and $R_Z(\alpha)$ are easily representable. Up to a scalar, H is also easily doable, and $\llbracket X^{(0,1)} \rrbracket = |0\rangle$. Hence, choosing $(e^{i\alpha}, |0\rangle, H, R_Z(\alpha), CZ)$ as spanning set, we only need the axioms:



Acknowledgements

The authors acknowledge support from the projects ANR-17-CE25-0009 SoftQPro, ANR-17-CE24-0035 VanQuTe, PIA-GDN/Quantex, and STIC-AmSud 16-STIC-05 FoQCoSS. All diagrams were written with the help of TikZit.

References

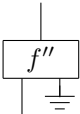
1. Matthew Amy, Jianxin Chen, and Neil J. Ross. A finite presentation of cnot-dihedral operators. In Bob Coecke and Aleks Kissinger, editors, Proceedings 14th International Conference on *Quantum Physics and Logic*, Nijmegen, The Netherlands, 3-7 July 2017, volume 266 of *Electronic Proceedings in Theoretical Computer Science*, pages 84–97. Open Publishing Association, 2018. doi:10.4204/EPTCS.266.5.
2. Koenraad M R Audenaert and Martin B Plenio. Entanglement on mixed stabilizer states: Normal forms and reduction procedures. *New Journal of Physics*, 7:170–170, aug 2005. URL: <https://doi.org/10.1088/1367-2630/7/1/170>, doi:10.1088/1367-2630/7/1/170.
3. Miriam Backens. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*, 16(9):093021, sep 2014. URL: <https://doi.org/10.1088/1367-2630/16/9/093021>, doi:10.1088/1367-2630/16/9/093021.
4. Miriam Backens. The ZX-calculus is complete for the single-qubit Clifford+T group. *Electronic Proceedings in Theoretical Computer Science*, 172:293–303, dec 2014. URL: <https://doi.org/10.4204/EPTCS.172.21>, doi:10.4204/EPTCS.172.21.
5. Miriam Backens and Ali Nabi Duman. A complete graphical calculus for Spekkens’ toy bit theory. *Foundations of Physics*, pages 1–34, 2014. arXiv:arXiv:1411.1618, doi:10.1007/s10701-015-9957-7.
6. Miriam Backens and Aleks Kissinger. Zh: A complete graphical calculus for quantum computations involving classical non-linearity. In Peter Selinger and Giulio Chiribella, editors, Proceedings of the 15th International Conference on *Quantum Physics and Logic*, Halifax, Canada, 3-7th June 2018, volume 287 of *Electronic Proceedings in Theoretical Computer Science*, pages 23–42. Open Publishing Association, 2019. doi:10.4204/EPTCS.287.2.
7. John C. Baez, Brandon Coya, and Franciscus Rebro. Props in network theory. In *Theory and Applications of Categories*, volume 33, pages 727–783, Jul 2017. URL: <http://arxiv.org/abs/1707.08321>.
8. Nicholas Chancellor, Aleks Kissinger, Joschka Roffe, Stefan Zohren, and Dominic Horsman. Graphical structures for design and verification of quantum error correction. last revised Jan. 2018, 2016. URL: <https://arxiv.org/abs/1611.08012>.
9. Bob Coecke. Axiomatic description of mixed states from Selinger’s CPM-construction. *Electronic Notes in Theoretical Computer Science*, 210:3 – 13, 2008. Proceedings of the 4th International Workshop on Quantum Programming Languages (QPL 2006). URL: <http://www.sciencedirect.com/science/article/pii/S1571066108002296>, doi:<https://doi.org/10.1016/j.entcs.2008.04.014>.

10. Bob Coecke and Ross Duncan. Interacting quantum observables: Categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, apr 2011. URL: <https://doi.org/10.1088%2F1367-2630%2F13%2F4%2F043016>, doi:10.1088/1367-2630/13/4/043016.
11. Bob Coecke and Chris Heunen. Pictures of complete positivity in arbitrary dimension. *Information and Computation*, 250:50–58, 2016.
12. Bob Coecke and Aleks Kissinger. The compositional structure of multipartite quantum entanglement. In *Automata, Languages and Programming*, pages 297–308. Springer Berlin Heidelberg, 2010. URL: https://doi.org/10.1007%2F978-3-642-14162-1_25, doi:10.1007/978-3-642-14162-1_25.
13. Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017. doi:10.1017/9781316219317.
14. Bob Coecke and Simon Perdrix. Environment and Classical Channels in Categorical Quantum Mechanics. *Logical Methods in Computer Science*, Volume 8, Issue 4, November 2012. URL: <https://lmcs.episciences.org/719>, doi:10.2168/LMCS-8(4:14)2012.
15. Bob Coecke and Quanlong Wang. ZX-rules for 2-qubit Clifford+T quantum circuits, 2018. arXiv:1804.05356.
16. Niel de Beaudrap and Dominic Horsman. The ZX-calculus is a language for surface code lattice surgery. *CoRR*, abs/1704.08670, 2017. URL: <http://arxiv.org/abs/1704.08670>, arXiv:1704.08670.
17. Ross Duncan. A graphical approach to measurement-based quantum computing. In *Quantum Physics and Linguistics*, pages 50–89. Oxford University Press, feb 2013. URL: <https://doi.org/10.1093%2Facprof%3Aoso%2F9780199646296.003.0003>, doi:10.1093/acprof:oso/9780199646296.003.0003.
18. Ross Duncan and Kevin Dunne. Interacting Frobenius algebras are Hopf. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2016*, pages 535–544, New York, NY, USA, 2016. ACM. URL: <http://doi.acm.org/10.1145/2933575.2934550>, doi:10.1145/2933575.2934550.
19. Ross Duncan and Liam Garvie. Verifying the smallest interesting colour code with quantomatic. In Bob Coecke and Aleks Kissinger, editors, *Proceedings 14th International Conference on Quantum Physics and Logic*, Nijmegen, The Netherlands, 3-7 July 2017, volume 266 of *Electronic Proceedings in Theoretical Computer Science*, pages 147–163. Open Publishing Association, 2018. doi:10.4204/EPTCS.266.10.
20. Ross Duncan, Aleks Kissinger, Simon Perdrix, and John van de Wetering. Graph-theoretic simplification of quantum circuits with the zx-calculus. *arXiv preprint arXiv:1902.03178*, 2019.
21. Ross Duncan and Maxime Lucas. Verifying the Steane code with Quantomatic. *Electronic Proceedings in Theoretical Computer Science*, 171:33–49, dec 2014. URL: <https://doi.org/10.4204%2FEptcs.171.4>, doi:10.4204/eptcs.171.4.
22. Ross Duncan and Simon Perdrix. Rewriting measurement-based quantum computations with generalised flow. *Lecture Notes in Computer Science*, 6199:285–296, 2010. URL: <http://personal.strath.ac.uk/ross.duncan/papers/gflow.pdf>, doi:10.1007/978-3-642-14162-1_24.
23. Ross Duncan and Simon Perdrix. Pivoting makes the ZX-calculus complete for real stabilizers. In *QPL 2013*, *Electronic Proceedings in Theoretical Computer Science*, pages 50–62, 2013. arXiv:arXiv:1307.7048, doi:10.4204/EPTCS.171.5.
24. Amar Hadzihasanovic. A diagrammatic axiomatisation for qubit entanglement. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 573–584, July 2015. doi:10.1109/LICS.2015.59.
25. Amar Hadzihasanovic. *The Algebra of Entanglement and the Geometry of Composition*. PhD thesis, University of Oxford, 2017. URL: <https://arxiv.org/abs/1709.08086>.
26. Amar Hadzihasanovic, Kang Feng Ng, and Quanlong Wang. Two complete axiomatisations of pure-state qubit quantum computing. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '18*, pages 502–511, New York, NY, USA, 2018. ACM. URL: <http://doi.acm.org/10.1145/3209108.3209128>, doi:10.1145/3209108.3209128.
27. Clare Horsman. Quantum pictorialism for topological cluster-state computing. *New Journal of Physics*, 13(9):095011, sep 2011. URL: <https://doi.org/10.1088%2F1367-2630%2F13%2F9%2F095011>, doi:10.1088/1367-2630/13/9/095011.
28. Mathieu Huot and Sam Staton. Universal properties in quantum theory. In Peter Selinger and Giulio Chiribella, editors, *Proceedings of the 15th International Conference on Quantum Physics and Logic*, Halifax, Canada, 3-7th June 2018, volume 287 of *Electronic Proceedings in Theoretical Computer Science*, pages 213–223. Open Publishing Association, 2019. doi:10.4204/EPTCS.287.12.
29. Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A complete axiomatisation of the ZX-calculus for Clifford+T quantum mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '18*, pages 559–568, New York, NY, USA, 2018. ACM. URL: <http://doi.acm.org/10.1145/3209108.3209131>, doi:10.1145/3209108.3209131.

30. Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Diagrammatic reasoning beyond Clifford+T quantum mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '18*, pages 569–578, New York, NY, USA, 2018. ACM. URL: <http://doi.acm.org/10.1145/3209108.3209139>, doi:10.1145/3209108.3209139.
31. Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A generic normal form for zx-diagrams and application to the rational angle completeness. 2018. [arXiv:1805.05296](https://arxiv.org/abs/1805.05296).
32. A. Kissinger and John van de Wetering. Pyzx, 2018. URL: <https://github.com/Quantomatic/pyzx>.
33. Aleks Kissinger and Sander Uijlen. A categorical semantics for causal structure. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.
34. Aleks Kissinger and Vladimir Zamdzhiev. Quantomatic: A proof assistant for diagrammatic reasoning. In Amy P. Felty and Aart Middeldorp, editors, *Automated Deduction - CADE-25*, pages 326–336, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-21401-6_22.
35. Ken Matsumoto and Kazuyuki Amano. Representation of Quantum Circuits with Clifford and $\pi/8$ Gates, June 2008. [arXiv:0806.3834](https://arxiv.org/abs/0806.3834).
36. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. doi:10.1017/CB09780511976667.
37. Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Comp. Sci.*, 14(4):527–586, August 2004. URL: <https://doi.org/10.1017/S0960129504004256>, doi:10.1017/S0960129504004256.
38. Peter Selinger. Dagger compact closed categories and completely positive maps. *Electronic Notes in Theoretical Computer Science*, 170:139–163, mar 2007. URL: <https://doi.org/10.1016/j.entcs.2006.12.018>, doi:10.1016/j.entcs.2006.12.018.
39. Peter Selinger. A survey of graphical languages for monoidal categories. In *New structures for physics*, pages 289–355. Springer, 2010.
40. Peter Selinger. Generators and Relations for n-qubit Clifford Operators. *Logical Methods in Computer Science*, Volume 11, Issue 2, June 2015. URL: <https://lmcs.episciences.org/1570>, doi:10.2168/LMCS-11(2:10)2015.
41. Peter Selinger and Xiaoning Bian. Relations for Clifford+T operators on two qubits, 2015. URL: <https://www.mathstat.dal.ca/~xbian/talks/>.
42. Renaud Vilmart. A near-optimal axiomatisation of ZX-calculus for pure qubit quantum mechanics. 2018. URL: <https://arxiv.org/abs/1812.09114>, [arXiv:arXiv:1812.09114](https://arxiv.org/abs/1812.09114).
43. Fabio Zanasi. *Interacting Hopf Algebras – the theory of linear systems*. PhD thesis, Université de Lyon, 2015. URL: <http://www.zanasi.com/fabio/#/publications.html>.

A Proofs

Proof (Proof of Lemma 1). Given any morphism $f : \mathbf{C}^\oplus[A, B]$, we take a diagram representing it. Using the naturality of the symmetry we obtain an equivalent diagram in \mathbf{C}^\oplus where all the discards

have been pushed to the bottom right: . There are no discards among the components of

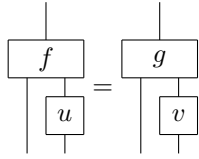
the part f'' of this diagram. So it represents a morphism in the range of $\iota_{\mathbf{C}}$ and then there is an $f' : \mathbf{C}[A, B \otimes X]$ such that:

$$\begin{array}{c} | \\ \boxed{f} \\ | \end{array} = \begin{array}{c} | \\ \boxed{\iota_{\mathbf{C}}(f')} \\ | \end{array}$$

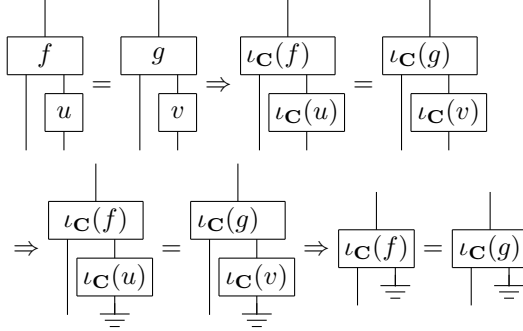
In other words, f' is a purification of f .

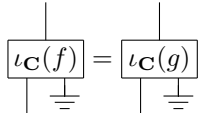
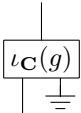
Proof (Proof of Lemma 2).

(\Rightarrow) It is enough to show $f \sim_{\text{iso}} g \Rightarrow \begin{array}{c} | \\ \boxed{\iota_{\mathbf{C}}(f)} \\ | \end{array} = \begin{array}{c} | \\ \boxed{\iota_{\mathbf{C}}(g)} \\ | \end{array}$ since equality is transitive.

$f \sim_{\text{iso}} g \Leftrightarrow$ there are two isometries $u : X \rightarrow Z$ and $v : Y \rightarrow Z$ such that  and

then:

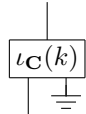


(\Leftarrow) We have  =  in \mathbf{C}^\sharp . To do the proof, we will have to go back to the definition of

the category \mathbf{C}^\sharp as a pushout. Recall that two terms are equal if one can rewrite one into the other using the equations defining \mathbf{C}^\sharp .

We can assume that, among those steps, the only one involving discards are isometry deletion/creation. Diagrammatically this amounts to say that the discards are never moved, in fact one can always moves the other morphisms to make them interact with the discards.

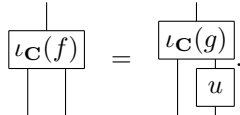
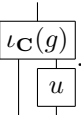
Doing this, we ensure that all intermediary diagrams in the chain of equations are of the form

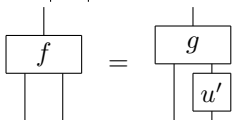
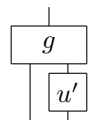


$\iota_{\mathbf{C}}(k)$ for some k . Therefore, to prove the result for a chain of equations of arbitrary size, it

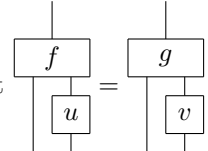
is enough to do it just for one step of rewriting.

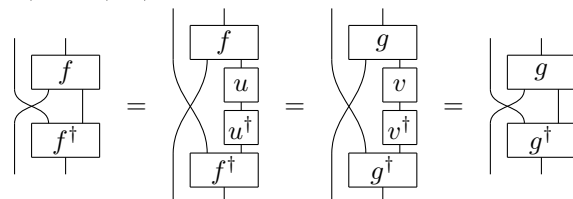
Consider then this step of rewriting. There are two cases. Either we have used an equation which, by identification, can be seen as an equation of \mathbf{C} , that is which involves no discards. Then by functoriality of $\iota_{\mathbf{C}}$ we recover that $f = g$ and therefore $f \sim_{\text{iso}} g$. Or the equation involves a discard which has deleted an isometry u . Then one of the upper part, let's say

$\iota_{\mathbf{C}}(f)$, can be written  = . But u being an isometry, there exists u' in \mathbf{C} such

that $\iota_{\mathbf{C}}(u') = u$. Hence, we have  =  in \mathbf{C} . It follows that $f \sim_{\text{iso}} g$.

Proof (Proof of Lemma 3). Since \sim_{cp} is transitive it is enough to show that $\sim_{\text{iso}} \subseteq \sim_{\text{cp}}$. Let $f : A \rightarrow B \otimes X$ and $g : A \rightarrow B \otimes Y$ s.t. $f \sim_{\text{iso}} g$. Then there are two isometries $u : X \rightarrow Z$ and

$v : Y \rightarrow Z$ such that  and then:



So $f \sim_{\text{cp}} g$.

Proof (Proof of Lemma 4).

[(i) \Leftrightarrow (ii)] First \mathbf{C}^\neq has the same object as \mathbf{C} and $\iota_{\mathbf{C}} : \mathbf{C} \rightarrow \overline{\mathbf{C}}$ is a SM-functor. We need to check the three conditions hold:

- Since $\iota_{\mathbf{C}^{\text{iso}}}$ is strict monoidal one has:

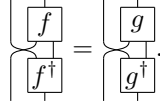
$$\begin{aligned} \frac{I}{\underline{\quad}} &= \iota_{\mathbf{C}^{\text{iso}}} (!_I) = \iota_{\mathbf{C}^{\text{iso}}} (id_I) = id_I \\ \frac{A}{\underline{\quad}} \otimes \frac{B}{\underline{\quad}} &= \iota_{\mathbf{C}^{\text{iso}}} (!_A) \otimes \iota_{\mathbf{C}^{\text{iso}}} (!_B) = \iota_{\mathbf{C}^{\text{iso}}} (!_A \otimes !_B) \\ &= \iota_{\mathbf{C}^{\text{iso}}} (!_{A \otimes B}) = \frac{A \otimes B}{\underline{\quad}} \end{aligned}$$

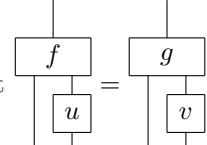
So the first condition is satisfied.

- The second condition is Lemma 1.
- According to Lemma 3, $\sim_{\text{iso}}^+ \subseteq \sim_{\text{cp}}$, thus the third condition is satisfied if and only if $\sim_{\text{cp}} \subseteq \sim_{\text{iso}}^+$.

[(i) \Leftrightarrow (ii)] Direct consequence of the fact that \mathbb{D} is an environment structure for \mathbb{C} iff D is equivalent to $CPM(\mathbb{C})$ [11].

Proof (Proof of Proposition 1). Let $f : A \rightarrow B \otimes X$ and $g : A \rightarrow B \otimes Y$ be two linear maps such

that $f \sim_{\text{cp}} g$. By definition: . It follows that the two superoperators $\rho \mapsto \text{tr}_X(f^\dagger \rho f)$ and $\rho \mapsto \text{tr}_Y(g^\dagger \rho g)$ are equal and then by the Stinespring dilation theorem (see for example [28]),

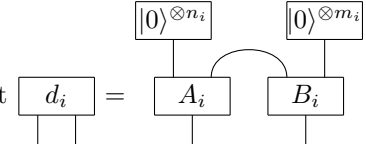
there are isometries u and v such that . In other words $f \sim_{\text{iso}} g$. This shows that

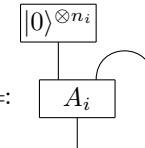
$\sim_{\text{cp}} \subseteq \sim_{\text{iso}}$ which is even stronger than the CP-condition. From Lemma 3 it follows that $\sim_{\text{iso}}^+ \subseteq \sim_{\text{iso}}$.

Proof (Proof of Proposition 2). It suffices to remark that, in the preceding proof for \mathbf{FHilb} , we might suppose wlog that u and v have codomain of the form \mathbb{C}^{2^n} , by postcomposing them if necessary with an isometry from \mathbb{C}^m to \mathbb{C}^{2^n} .

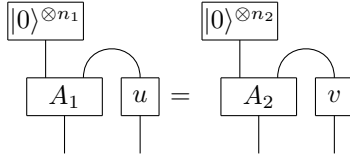
Therefore $f \sim_{\text{cp}} g$ on **Qubit** implies $f \sim_{\text{iso}} g$ on **Qubit**.

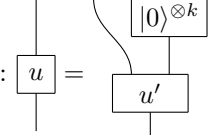
Proof (Proof of Proposition 3). First of all, since **Stab** is compact closed, using the map/state duality, proving the result for states is sufficient. Since all the non-zero scalars are invertible in **Stab** we can furthermore without loss of generality focusing on normalized states. Consider two states $d_1 : A \otimes X$ and $d_2 : A \otimes Y$ in **Stab** such that $d_1 \sim_{\text{cp}} d_2$. The point of focusing on normalized states is that

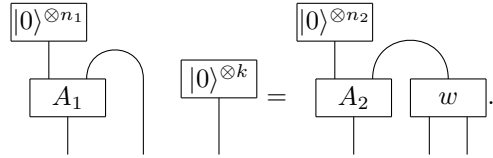
we can decompose using [2] so that  where A_i and B_i are unitaries

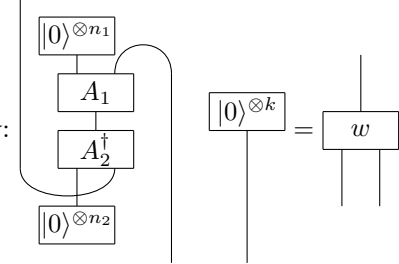
in **Stab**. Defining $A'_i =$  we have that $d_i \sim_{\text{iso}} A'_i$ since we just have deleted isometries.

So, by transitivity, to prove $d_1 \sim_{\text{iso}}^+ d_2$ we just have to show $A'_1 \sim_{\text{iso}} A'_2$. But since $d_1 \sim_{\text{cp}} d_2$ in **Stab** we also have $d_1 \sim_{\text{cp}} d_2$ in **FHilb** and so by Lemma 1, $d_1 \sim_{\text{iso}}^+ d_2$ in **FHilb**. By transitivity $A'_1 \sim_{\text{iso}}^+ A'_2$ in **FHilb** and so by Lemma 1 $A'_1 \sim_{\text{iso}} A'_2$ in **FHilb**. So there are two unitaries u and

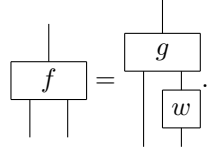
v such that . In **FHilb** any isometry can be written as an unitary

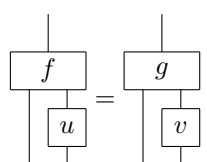
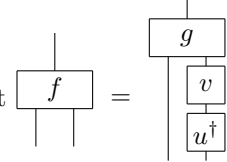
with ancillas. In other words there is an unitary u' such that: , composing by u'^{\dagger}

on both side and denoting $w = u'^{\dagger} \circ v$ one has: . It only remains to show that the isometry w is in **Stab** since the isometry on left hand side is clearly in

it. This is given by:  so $A_1' \sim_{\text{iso}} A_1'$ in **Stab** and then $d_1 \sim_{\text{cp}} d_2$.

Proof (Proof of Proposition 4). First remark that, in any \dagger -SMC category, if $f \sim_{\text{iso}}^{\dagger} g$ then there

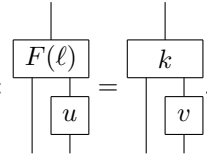
is a morphism (usually not an isometry) w such that .

This is true if $f \sim_{\text{iso}} g$: From  we immediately get .

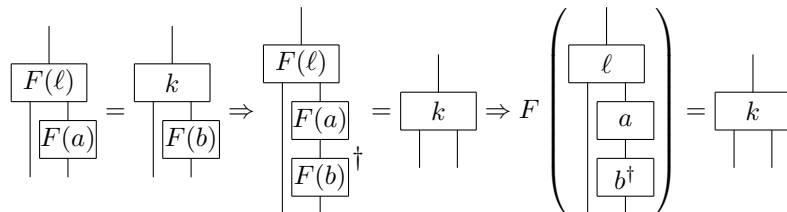
The result then follows by a straightforward induction.

Now take $\phi = 1 + 2i$ and $\phi^* = 1 - 2i$. The scalars are in **Clifford+T** since their entries are in $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$, and are clearly \sim_{cp} equivalent. Now let's suppose $1 + 2i \sim_{\text{iso}}^{\dagger} 1 - 2i$. Then by the previous remark, there exists a morphism u such that $(1 - 2i)u = 1 + 2i$. But the only possibility for u is $\frac{4i-3}{5}$, which is not in $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$, a contradiction.

Proof (Proof of Lemma 5). First, remark that if $F(\ell) \sim_{\text{iso}} k$, then there exists h s.t. $F(h) = k$.

Indeed, under the hypothesis, there are two isometries u and v such that: . Since

F_{iso} is surjective, there are two isometries a and b such that $F(a) = u$ and $f(b) = v$.



The first implication uses the fact that $F(b)$ is an isometry. So k is in the image of F .

By the first remark, it is therefore sufficient to prove the result if $F(f) \sim_{\text{iso}} F(g)$. Since F_{iso} is surjective, there are two isometries a and b such that $F(a) = u$ and $f(b) = v$. Therefore

$$\begin{array}{c} \boxed{F(f)} \\ \boxed{F(a)} \end{array} = \begin{array}{c} \boxed{F(g)} \\ \boxed{F(b)} \end{array} \Rightarrow F \left(\begin{array}{c} \boxed{f} \\ \boxed{a} \end{array} \right) = F \left(\begin{array}{c} \boxed{g} \\ \boxed{b} \end{array} \right) \Rightarrow \begin{array}{c} \boxed{f} \\ \boxed{a} \end{array} = \begin{array}{c} \boxed{g} \\ \boxed{b} \end{array}$$

The second one holds because F is faithful. The last equation is the definition of $f \sim_{\text{iso}} g$.

Proof (Proof of Theorem 1). Let f and g be two morphisms such that $F^\cong(f) = F^\cong(g)$. By Lemma 1, f and g can be purified:

$$F^\cong \left(\begin{array}{c} \boxed{\iota_C(f')} \\ \underline{\quad} \end{array} \right) = F^\cong \left(\begin{array}{c} \boxed{\iota_C(g')} \\ \underline{\quad} \end{array} \right) \Rightarrow \begin{array}{c} \boxed{\iota_D F(f')} \\ \underline{\quad} \end{array} = \begin{array}{c} \boxed{\iota_D F(g')} \\ \underline{\quad} \end{array}$$

The implication follows from the upper face of the commutative cube. By Lemma 2 we have

$F(f') \sim_{\text{iso}}^+ F(g')$. By Lemma 5, $f' \sim_{\text{iso}}^+ g'$. Then Lemma 2 gives $\begin{array}{c} \boxed{\iota_C(f')} \\ \underline{\quad} \end{array} = \begin{array}{c} \boxed{\iota_C(g')} \\ \underline{\quad} \end{array}$ that is $f = g$, F

is faithful.

B ZW and ZH Calculi

B.1 ZW-calculus

ZW-diagrams are generated by:

$Z^{(n,m)}(r) : n \rightarrow m$		$\mathbb{I} : 1 \rightarrow 1$	
$W^{(n,m)} : n \rightarrow m$		$e : 0 \rightarrow 0$	
$\sigma : 2 \rightarrow 2$		$\sigma' : 2 \rightarrow 2$	
$\epsilon : 2 \rightarrow 0$		$\eta : 0 \rightarrow 2$	

where $n, m \in \mathbb{N}$, $r \in \mathbb{C}$, and the generator e is the empty diagram.

and the two compositions: spacial ($\cdot \otimes \cdot$) and sequential ($\cdot \circ \cdot$).

The standard interpretation is defined as:

[[·]]

$$[[D_1 \otimes D_2]] := [[D_1]] \otimes [[D_2]]$$

$$[[D_2 \circ D_1]] := [[D_2]] \circ [[D_1]]$$

$$[[\boxed{\quad}]] := | \rangle$$

$$[[|]] := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$[[\sigma]] := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$[[\sigma']] := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

$Z^{(n,m)} : n \rightarrow m$		$\mathbb{I} : 1 \rightarrow 1$
$X^{(n,m)} : n \rightarrow m$		$e : 0 \rightarrow 0$
$H^{(n,m)}(a) : n \rightarrow m$		$\neg : 1 \rightarrow 1$
$\sigma : 2 \rightarrow 2$		$\epsilon : 2 \rightarrow 0$
$\eta : 0 \rightarrow 2$		

where $n, m \in \mathbb{N}$, $a \in \mathbb{C}$, and the generator e is the empty diagram.

and the two compositions: spacial ($\cdot \otimes \cdot$) and sequential ($\cdot \circ \cdot$).

The language was introduced to allow a simple representation of hypergraph states and multi-controlled-Z gates. To do so it features a node called H -spider, which can be seen as a generalisation of the Hadamard gate. By convention, when no parameter is specified in H , the implicit parameter taken is -1 : $\square := \square_{-1}$.

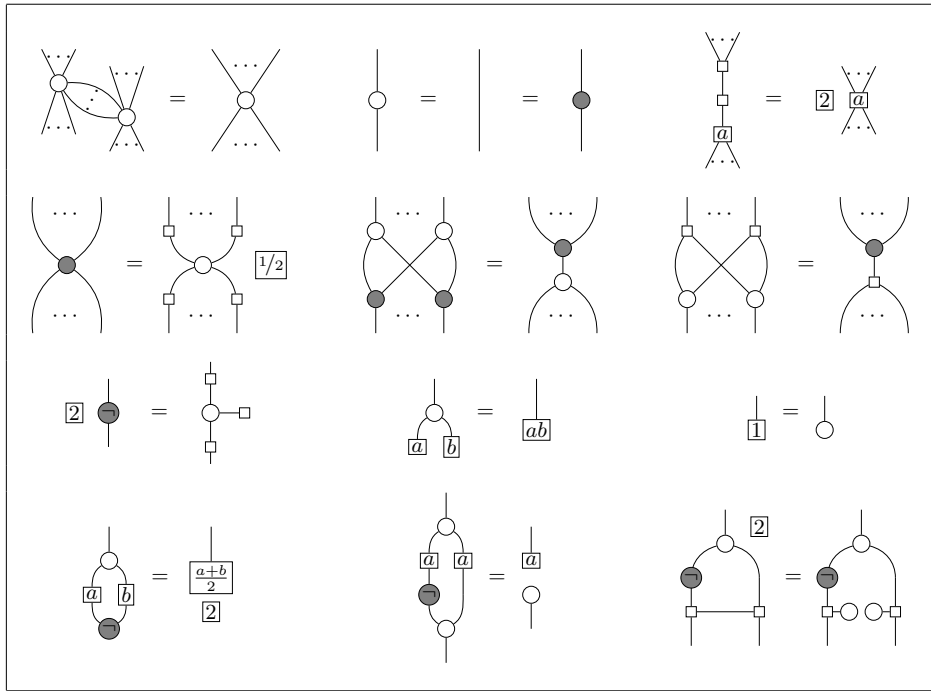


Fig. 2. Set of rules ZH. (...) denote zero or more wires, while ($\cdot \cdot$) denote one or more wires.

The language comes with a standard interpretation defined as:

$$\begin{aligned}
 \llbracket D_1 \otimes D_2 \rrbracket &:= \llbracket D_1 \rrbracket \otimes \llbracket D_2 \rrbracket & \llbracket D_2 \circ D_1 \rrbracket &:= \llbracket D_2 \rrbracket \circ \llbracket D_1 \rrbracket \\
 \llbracket \square \rrbracket &:= (1) & \llbracket \square_a \rrbracket &:= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \llbracket \square_a \square_b \rrbracket &:= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
 \end{aligned}$$

$$\begin{aligned}
\llbracket \text{X} \rrbracket &:= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & \llbracket \text{C} \rrbracket &:= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\
\llbracket \text{C} \rrbracket &:= (1 \ 0 \ 0 \ 1) & \llbracket \text{C}_a \rrbracket &:= \begin{pmatrix} 1 \cdots 1 & 1 \\ \vdots & \vdots \\ 1 \cdots 1 & 1 \\ 1 \cdots 1 & a \end{pmatrix} \\
\llbracket \text{C}_a \rrbracket &:= \begin{pmatrix} 1 & 0 \cdots 0 & 0 \\ 0 & 0 \cdots 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 \cdots 0 & 0 \\ 0 & 0 \cdots 0 & 1 \end{pmatrix} \\
\llbracket \text{C}_a \rrbracket &:= \frac{1}{2} \llbracket \text{C} \rrbracket^{\otimes m} \circ \llbracket \text{C} \rrbracket \circ \llbracket \text{C} \rrbracket^{\otimes n}
\end{aligned}$$

A set of rules was proposed together with the language (Figure 2). It makes the ZH-Calculus complete for **Qubit**.

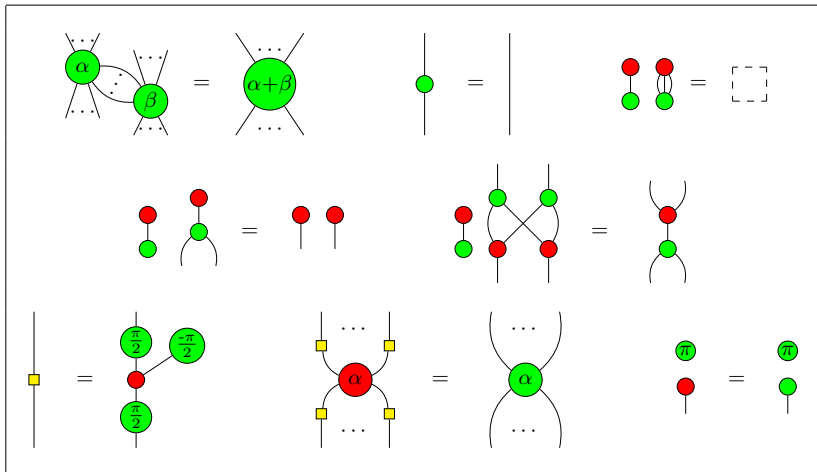


Fig. 4. Set of rules $ZX_{\frac{\pi}{2}}$ for the $\frac{\pi}{2}$ -fragment of the ZX-Calculus with scalars. All of these rules also hold when flipped upside-down, or with the colours red and green swapped. The right-hand side of (IV) is an empty diagram. (...) denote zero or more wires, while (. \cdot .) denote one or more wires.