



HAL
open science

Complete Graphical Language for Hermiticity-Preserving Superoperators

Titouan Carette, Timothée Hoffreumon, Émile Larroque, Renaud Vilmart

► **To cite this version:**

Titouan Carette, Timothée Hoffreumon, Émile Larroque, Renaud Vilmart. Complete Graphical Language for Hermiticity-Preserving Superoperators. 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) 2023, Jun 2023, Boston, United States. pp.1-22, 10.1109/LICS56636.2023.10175712 . hal-04001823

HAL Id: hal-04001823

<https://hal.science/hal-04001823>

Submitted on 23 Feb 2023

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.

Complete Graphical Language for Hermiticity-Preserving Superoperators

Titouan Carette,

Centre for Quantum Computer Science, Faculty of Computing, University of Latvia,
Raina 19, Riga, Latvia, LV-1586.

Timothée Hoffreumon,

Centre for Quantum Information and Communication (QuIC),
École polytechnique de Bruxelles,
CP 165, Université libre de Bruxelles, 1050 Brussels, Belgium.

Emile Larroque,

Université Paris-Saclay, ENS Paris-Saclay, Inria, CNRS, LMF,
91190, Gif-sur-Yvette, France.

Renaud Vilmart,

Université Paris-Saclay, ENS Paris-Saclay, Inria, CNRS, LMF,
91190, Gif-sur-Yvette, France.

February 9, 2023

Abstract

Universal and complete graphical languages have been successfully designed for pure state quantum mechanics, corresponding to linear maps between Hilbert spaces, and mixed states quantum mechanics, corresponding to completely positive superoperators. In this paper, we go one step further and present a universal and complete graphical language for Hermiticity-preserving superoperators. Such a language opens the possibility of diagrammatic compositional investigations of antilinear transformations featured in various physical situations, such as the Choi-Jamiolkowski isomorphism, spin-flip, or entanglement witnesses. Our construction relies on an extension of the ZW-calculus exhibiting a normal form for Hermitian matrices.

1 Introduction

Experimentally, all one can infer from a process theory is an outcome distribution. In the case of quantum theory, this distribution is given by the Born rule, stating that the probability of measuring state ¹ $|\phi\rangle$ from a prepared state $|\psi\rangle$ is $|\langle\phi|\psi\rangle|^2$. A symmetry of the theory is a transformation of the states that leave this rule invariant, *i.e.* T is a symmetry of quantum theory if it obeys

$$|\langle\phi|\psi\rangle|^2 = |\langle T(\phi)|T(\psi)\rangle|^2, \quad (1)$$

¹We use the usual Dirac notation for complex vectors ($|\Psi\rangle \in \mathbb{C}^n$ called *ket*) and their dual ($\langle\Psi| = |\Psi\rangle^\dagger$, called *bra*, and where \dagger is the *dagger*, that is, the transpose conjugate or Hermitian adjoint). See [37] for more info.

for any two states $|\psi\rangle$ and $|\phi\rangle$. In 1931, Wigner proved that the symmetries of quantum theory should be either unitary or antiunitary (see the appendix to Chapter 20 of [48]), meaning that the general form of T in Eq. (1) is

$$|T(\psi)\rangle = U|\psi\rangle \quad \text{or} \quad |T(\psi)\rangle = U\overline{|\psi\rangle}, \quad (2)$$

where U is a linear operator respecting $U^\dagger U = I = U U^\dagger$, and $\overline{|\psi\rangle}$ is the complex conjugation of $|\psi\rangle$.

This abstract point of view turned out to be extremely useful for all aspects of quantum theory. On the one hand, the study of symmetries has been a very fruitful method for simplifying problems encountered at all scales, from solid-state physics to high-energy physics, with notable uses in atomic physics and quantum information sciences. On the other hand, symmetries have been a guiding principle for the construction of theories such as the standard model and, by extension, Yang-Mills theory (see e.g. [45, 46]).

So far, physicists and quantum computer scientists have been mainly concerned with unitary transformations, the ubiquitous ingredient in the study of dynamics, and little attention has been devoted to antiunitary transformations. There is a clear reason why: unitary transformations represent what can be realised and tested in a (closed) lab. In contrast, one year after his theorem, Wigner showed that antiunitaries are typically involved in the case of a time reversal [47] (see, for example, §11.4.2 of [41] for a source in English), something experimentalists cannot achieve (at least within the current theoretical framework of physics). Still, some quantum computer scientists have started investigating the possible computational advantages processes involving time reversal could provide. We mention in particular the complexity-theoretic study of [1] and the categorical semantics of [39]. However, little has been done on the computational power of antiunitaries in general, although some results suggest possible advantages in quantum information [8, 22, 36], and that they may be simulable [5, 19, 40]. Such investigations are actually made difficult by the lack of a clearly defined computational framework because it would require going beyond the usual mathematics underlying quantum circuits. Rigorously put, antiunitaries are associated with positive but not completely positive mappings. Hence, they are outside the framework of quantum computing with open systems, and are assumed non-physical.

Despite this practical limitation and even apart from speculative investigations of computational advantages, these transformations are still of fundamental importance, for example, in the proofs of the CPT and spin statistics theorems [42]. In particular, there is a long history of the use of antiunitary transformation in the mathematics of quantum information theory to identify non-classical behaviour. The canonical example is the Positive Partial Transpose criterion (PPT, also called Peres-Horodecki [28, 38]), giving a necessary condition for states to be separable. Transposition is indeed the prototypical example of an antiunitary transformation in the space of density matrices. It is also involved in the definition of Wootters concurrence [26, 50], and it has been argued to play a particular role in the EPR paradox as well as in quantum teleportation (see [43]).

To better understand the role of antiunitaries in quantum weirdness, it is crucial to have a language that can handle both unitary and antiunitary transformations on the same footing. Therefore, this paper aims to introduce a universal and complete graphical language with enough generators to represent both transformations.

Different graphical languages for depicting quantum states and evolutions exist, from the ubiquitous circuits to the more lax and abstract ZX-, ZW- and ZH-calculi. By weakening the requirement of unitarity, the three latter enjoy a richer structure, namely compact closure, making them more convenient to manipulate. All the aforementioned languages now enjoy a *complete* equational theory, meaning that any two (pure) circuits or diagrams that represent the same operator can be turned into one another [3, 11, 25, 44]. Using the *discard*-construction [7], it is possible to extend the expressiveness of the languages to *mixed* states while at the same time preserving their completeness.

In this paper, we add a “anti-unitarity-inducing” generator to one of these languages and provide an equational theory, the completeness of which is proven via normal forms, rather than by a characterization of equivalence in the new semantics as it was done for mixed states. The ZW-calculus turns out to be well suited for the matter, as its pure version enjoys a natural normal form that can be leveraged in our case.

2 Preliminaries on Pictorial Quantum Mechanics

In this section, we recall the necessary definitions and properties of the pictorial approach to quantum computing.

2.1 Props and Graphical Languages

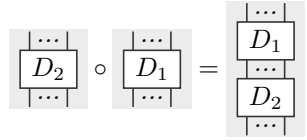
We represent quantum processes as boxes with n input and m output wires, each wire corresponding to a qubit:



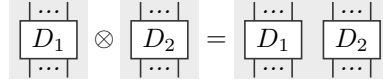
Formally those different processes are organised into props.

Definition 1 (Prop). A **prop** \mathbf{P} is a collection of sets of processes $\mathbf{P}[n, m]$ indexed by integers $n, m \in \mathbb{N}$, together with:

▷ An associative vertical composition operation: $-\circ - : \mathbf{P}[b, c] \times \mathbf{P}[a, b] \rightarrow \mathbf{P}[a, c]$ pictured as:

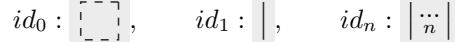


▷ An associative horizontal composition operation: $-\otimes - : \mathbf{P}[a, b] \times \mathbf{P}[c, d] \rightarrow \mathbf{P}[a + c, b + d]$ pictured as:



and satisfying $(f \circ g) \otimes (h \circ k) = (f \otimes h) \circ (g \otimes k)$.

▷ Identity processes $id_0 \in \mathbf{P}[0, 0]$ and $id_1 \in \mathbf{P}[1, 1]$, which by horizontal compositions combine into identity processes $id_n \in \mathbf{P}[n, n]$ pictured as:



and satisfying: $f \otimes id_0 = id_0 \otimes f = id_n \circ f = f \circ id_m = f$ for all $f \in \mathbf{P}[n, m]$.

▷ A swap $\sigma_{1,1} \in \mathbf{P}[1 + 1, 1 + 1]$ that, combined with identities, provides generalized swaps $\sigma_{a,b} \in \mathbf{P}[a +$

$b, b + a]$ depicted as satisfying: = and = for all $D \in \mathbf{P}[n, m]$.

We often use the type-theoretic notation $f : n \rightarrow m$ for $f \in \mathbf{P}[n, m]$. Processes with no inputs, i.e. those in $\mathbf{P}[0, m]$ for some m , are usually referred to as states.

In category-theoretic terms, a prop is a strict symmetric monoidal category (SMC) with $(\mathbb{N}, +)$ as a monoid of objects. In other words, there is a distinguished object X in the category such that every object has the form $X^{\otimes n}$ for $n \in \mathbb{N}^+$. In categorical quantum information theory, one usually deals with **FdHilb**, the category whose processes (or arrows) are linear maps between finite-dimensional Hilbert spaces over complex numbers. However, **FdHilb** is not a prop since its generating objects (wires) can have different dimensions in general. We thus need to fix the dimension of each wire.

This paper will focus on the prop spanned by 2-dimensional complex vectors and its relevant sub-props. We define it as:

Definition 2 (Hilb). *The prop **Hilb** has for processes $n \rightarrow m$ the linear maps $\mathbb{C}^{2^n} \rightarrow \mathbb{C}^{2^m}$. That is, the space of $2^m \times 2^n$ matrices with complex entries, denoted $\mathcal{M}_{2^m \times 2^n}(\mathbb{C})$.*

Hilb is thus the prop whose wires are qubit spaces. For this reason, it is sometimes called **Qubit** in the literature.

A graphical language is then a set of elementary processes (or gates) called generators, together with a set of equations between diagrams formed from the generators called rules.

Categorically, graphical languages are axiomatisations of props. We will often identify a graphical language with the prop it defines, and whose processes are diagrams obtained by combining generators, quotiented by the rewriting rules. A graphical language comes with an interpretation functor defining its semantics; see [4] for more details.

Definition 3 (Graphical language). *A graphical language is a tuple $(\Sigma, E, \llbracket \cdot \rrbracket : G \rightarrow \mathbf{P})$ where Σ is a set of processes, called the generators; E is a set of equations relating diagrams made of generators, called the rules; G is the prop axiomatised by Σ and E ; and \mathbf{P} is a prop where we define the semantics. The functor $\llbracket \cdot \rrbracket : G \rightarrow \mathbf{P}$ is called the interpretation of the graphical language in \mathbf{P} .*

A desirable property is that the interpretations of generators span the full semantics.

Definition 4 (Universality). *A graphical language is called **universal** if $\llbracket \cdot \rrbracket$ is full, meaning that all processes in \mathbf{P} can be represented by a diagram built using the tensor product and composition from swaps, identities and generators of the language.*

The equations are required to be sound, meaning that elements of the graphical language which are equivalent up to rewriting rules have the same interpretation. The converse property – that diagrams with the same interpretation are equivalent in the graphical language – is called completeness.

Definition 5 (Completeness). *A graphical language is called **complete** if $\llbracket \cdot \rrbracket$ is faithful, meaning that the equational theory identifies together all the diagrams that have the same interpretation.*

Several graphical languages have been shown to be universal and complete for **Hilb**: the ZX-calculus, the ZH-calculus and the ZW-calculus.

2.2 Doubling

The maps in **Hilb** allow us to represent pure quantum processes like unitary gates but also un-physical scalars; to represent faithfully quantum mechanics, one must shift to the ‘doubled’ theory.

In this picture, the prop corresponding to **Hilb** is called **Lin**; its processes are linear maps between operators, called superoperators, and its states are operators themselves. We will often see a superoperator state $0 \rightarrow n$ as a matrix in $\mathcal{M}_{2^n \times 2^n}(\mathbb{C})$.

Definition 6 (Lin). *The prop **Lin** has for processes $n \rightarrow m$ the linear maps $\mathcal{M}_{2^n \times 2^n}(\mathbb{C}) \rightarrow \mathcal{M}_{2^m \times 2^m}(\mathbb{C})$.*

The two props are connected by a construction called ‘doubling’. Colloquially, it amounts to passing from states being ‘ket’ vectors to them being ‘ket-bra’ matrices (i.e. projectors).

Definition 7 (Doubling). *The **doubling** prop functor **double** : **Hilb** \rightarrow **Lin** is defined on all $A \in \mathcal{M}_{2^m \times 2^n}(\mathbb{C})$ as **double**(A) : $\rho \mapsto A\rho A^\dagger$. In particular on a state $|\phi\rangle : 0 \rightarrow n$ we get: **double**($|\phi\rangle$) = $|\phi\rangle\langle\phi| \in \mathcal{M}_{2^n \times 2^n}(\mathbb{C})$.*

The image of **Hilb** in **Lin** under doubling is a sub-prop of **Lin** that we define below.

Definition 8 (Pure maps). *A linear map $n \rightarrow m$ is said pure if it has a Kraus rank of 1.*

We denote by **Pure** the prop whose processes $n \rightarrow m$ are pure maps. Equivalently, **Pure** is the prop obtained by doubling **Hilb**.

2.3 Process-State Duality

In the language of categories, the props **Hilb** and **Lin** are dagger compact categories [15]. It means that in addition to being SMC, they come equipped with an involution, the dagger \dagger , as well as a distinguished state, the cap $\cap : 0 \rightarrow 2$, and its corresponding dagger dual, the cup $\cup : 2 \rightarrow 0$. Pictorially, the dagger reverses the direction of the diagrams while conjugating their coefficients. As for the cup and the cap, they satisfy the ‘yanking equations’:

$$\begin{array}{c} \cap \\ \cup \end{array} = \text{---} = \begin{array}{c} \cup \\ \cap \end{array}, \quad (3)$$

and they are invariant to the swap:

$$\begin{array}{c} \cap \\ \cup \end{array} = \begin{array}{c} \cup \\ \cap \end{array} \quad \text{and} \quad \begin{array}{c} \cup \\ \cap \end{array} = \begin{array}{c} \cap \\ \cup \end{array}. \quad (4)$$

The compactness of these props, i.e. having a cup and cap, induces a process-state (sometimes called map/state or channel/state) duality: any process can be made into a state by appending caps to all of its inputs:

$$\mathbf{P}[n, m] \rightarrow \mathbf{P}[0, m+n] : \begin{array}{c} \dots \\ \vdots \\ \text{---} \\ \vdots \\ \dots \end{array} \mapsto \begin{array}{c} \dots \\ \vdots \\ \text{---} \\ \vdots \\ \dots \end{array}. \quad (5)$$

In the language of quantum information, the cap is a maximally entangled state, and the cup is the corresponding postselected measurement. Since there are several maximally entangled states, it is customary to pick $\frac{1}{\sqrt{d}} \left(\sum_{k=0}^{d-1} |k\rangle \otimes |k\rangle \right)$. Therefore, for **Hilb** whose underlying Hilbert spaces are two-dimensional, it is the $|\phi^+\rangle$ Bell state.

The process-state duality of **(Fd)Hilb** is usually called *vectorisation* in the literature [21]. In computer science and category theory, this vectorisation process corresponds to the geometry of interaction construction **Int** [33]. In fact **Lin** is exactly **Int(Hilb)** which is equivalent to **Hilb** as **Hilb** is compact closed. We will be primarily interested in the process-state duality for **Lin**. Since linear maps between density operators are easier to handle when they are turned into density operators themselves, process-state duality occurs much more often in the literature for **Lin** than for **Hilb**, where it is common practice to omit the normalization of the cap. This specific instance of process-state duality is usually referred to as the *Choi-Jamiołkowski isomorphism*.

Definition 9 (Choi-Jamiołkowski isomorphism [10, 30]). *Let \mathcal{F} be a linear map between square matrices of dimensions n^2 and m^2 . The **Choi-Jamiołkowski isomorphism** is a bijective isomorphism between \mathcal{F} and a matrix F given by*

$$\begin{aligned} & (\mathcal{M}_{n \times n}(\mathbb{C}) \rightarrow \mathcal{M}_{m \times m}(\mathbb{C})) \rightarrow \mathcal{M}_{(n+m) \times (n+m)}(\mathbb{C}) : \\ & \mathcal{F} \mapsto F := (\mathcal{I} \otimes \mathcal{F}) \left(\sum_{k,l=0}^{n-1} |k\rangle\langle l| \otimes |k\rangle\langle l| \right), \end{aligned} \quad (6)$$

where \mathcal{I} is the identity map. F is called the **Choi matrix** of \mathcal{F} .

The **reverse direction** of the isomorphism is the image in $\mathcal{M}_{m \times m}(\mathbb{C})$ of the action of \mathcal{F} on a matrix ρ :

$$\mathcal{F}(\rho) = \text{Tr}_{\mathcal{M}_{2^n \times 2^n}(\mathbb{C})} \left[F \left(\rho^T \otimes \mathbf{1}_{\mathcal{M}_{2^m \times 2^m}(\mathbb{C})} \right) \right]. \quad (7)$$

In the above, T is the transposition with respect to a fixed basis, and $\text{Tr}_{\mathcal{M}_{2^n \times 2^n}(\mathbb{C})}$ is the partial trace over subsystem $\mathcal{M}_{2^n \times 2^n}(\mathbb{C})$.

2.4 Completely Positive Maps

Pure only represents closed quantum dynamics between pure states. In order to be more general, we have to consider another sub-prop of **Lin** containing mixed-state quantum theory. That is, the prop generated by Completely Positive (CP) maps.

Definition 10 (Complete Positivity). *A superoperator \mathcal{F} is said to be **Completely-Positive** (CP) if, for all input positive operators ρ , it produces a positive operator, even if applied locally:*

$$\forall \rho, \quad \rho \geq 0 \quad \Rightarrow \quad (\mathcal{I} \otimes \mathcal{F})(\rho) \geq 0, \quad (8)$$

where \mathcal{I} is the identity superoperator.

We denote by **CP** the prop whose processes $n \rightarrow m$ are the CP superoperators in $\mathcal{M}_{2^n \times 2^n}(\mathbb{C}) \rightarrow \mathcal{M}_{2^m \times 2^m}(\mathbb{C})$.

Through process-state duality, CP maps have a nice characterisation in terms of matrices:

Theorem 1 (Choi [10]). *The Choi-Jamiołkowski isomorphism maps CP maps to positive semi-definite matrices.*

Moreover, the structure of CP maps as an extension of pure maps is well understood: Purification – a standard result (see e.g. [37]) – states these can be obtained by composition of pure maps and partial traces. A canonical way to extend a graphical language for pure to CP maps is then to introduce a generator that will perform the partial trace. In [7, 16, 17], this generator is represented as $\frac{\perp}{\perp} : 1 \rightarrow 0$.

Proposition 1 ([12]). *Adding the discard map, i.e. partial trace superoperator, as a generator to a universal set of generators for pure maps provides a universal set of generators for completely positive maps.*

Proposition 2 ([7]). *Any universal and complete graphical language for pure maps extends into a universal and complete graphical language for CP maps by adding the discard generator $\frac{\perp}{\perp}$ and the equation:*

$$\begin{array}{c} | \\ \boxed{V} \\ \frac{\perp}{\perp} \end{array} = \frac{\perp}{\perp}$$

for each isometry V , that is, for each map V satisfying $V^\dagger V = I$.

Proposition 2 was used in [7] to extend the ZX-, ZH-, and ZW-calculi, which are universal and complete for pure maps, to universal and complete graphical languages for CP maps. It is noteworthy that this can be done with a finite number of (parameterised) equations since a finite set of generators for isometries is known.

3 Depicting Hermiticity-Preserving Superoperators

3.1 Antilinearity and the Transpose Map

As the initial motivation for this work is the graphical representation of antiunitary maps, we begin by recollecting some facts about these maps.

Definition 11. *A map $\theta : \mathbb{C}^n \rightarrow \mathbb{C}^m$ is called antilinear (or conjugate-linear) if it verifies*

$$\theta(c_1 |\psi\rangle + c_2 |\phi\rangle) = \overline{c_1} \theta(|\psi\rangle) + \overline{c_2} \theta(|\phi\rangle), \quad (9)$$

for all vectors $|x\rangle, |y\rangle \in \mathbb{C}^n$, and for all complex numbers $c_1, c_2 \in \mathbb{C}$.

If the space has an inner product $\langle \cdot | \cdot \rangle : \mathbb{C}^n \rightarrow \mathbb{C}$, the Hermitian adjoint of an antilinear map can be defined. Let θ be an antilinear map from \mathbb{C}^n to \mathbb{C}^m , then its adjoint is the map $\theta^\dagger : \mathbb{C}^m \rightarrow \mathbb{C}^n$ which obeys

$$\langle y | \theta(x) \rangle = \langle x | \theta^\dagger(y) \rangle \quad \forall x \in \mathbb{C}^n, \forall y \in \mathbb{C}^m. \quad (10)$$

A special class of antilinear operators are the antiunitaries.

Definition 12. *An antilinear operator A is an antiunitary if it is normal, i.e. it commutes with its adjoint $AA^\dagger = A^\dagger A$, and if it conjugates the scalar product,*

$$\langle A(y) | A(x) \rangle = \overline{\langle y | x \rangle} = \langle x | y \rangle. \quad (11)$$

Mathematically, antiunitaries are ‘one coset away’ from unitaries, in the sense that they can always be written as [49]

$$A = U \circ K, \quad (12)$$

where U is a unitary and K is an antilinear involution, meaning that it obeys $K(cId) = \bar{c}K$ for $c \in \mathbb{C}$ and $K^2 = Id$. Without loss of generality, the same K can be used for all A . Thus, constructing a graphical language for antiunitary maps amounts to depicting the map K and its interaction with the \mathbb{C} -linear maps. In the case of pure state (single wire) quantum mechanics, the standard choice is to take K as the operation of complex conjugation,

$$K |x\rangle = \overline{|x\rangle}, \quad (13)$$

so that $K(c_1 |0\rangle + c_2 |1\rangle) = \bar{c}_1 |0\rangle + \bar{c}_2 |1\rangle$ for example.

This choice induces the use of the transposition as the involution for the mixed state (i.e. doubled) representation. The doubling of a pure state can be written as: $|x\rangle \mapsto |x\rangle \otimes |x\rangle^\dagger = |x\rangle \otimes \langle x| \cong |x\rangle \langle x|$. Thus, the doubling of an antiunitary action is: $A|x\rangle \mapsto A|x\rangle \otimes (A|x\rangle)^\dagger = U\overline{|x\rangle} \otimes (U\overline{|x\rangle})^\dagger = U\overline{|x\rangle} \otimes \overline{\langle x|} U^\dagger \cong U\overline{|x\rangle} \langle x| U^\dagger$. And since $|x\rangle \langle x|$ is self-adjoint this yields

$$A|x\rangle \mapsto U(|x\rangle \langle x|)^T U^\dagger. \quad (14)$$

Since the transposition is a linear map on the space of density matrices, this construction linearly extends from the pure processes to all processes.

Using the result of Wigner, unitaries and antiunitaries are symmetries of density matrices,

$$U(\rho) = U\rho U^\dagger \quad \text{or} \quad \mathcal{A}(\rho) = U\rho^T U^\dagger. \quad (15)$$

Surprisingly, going to the doubled picture does not induce any new symmetries beside these two (see App. A and B of [9] for a proof).

When deriving the mixed state representation of antiunitaries, we implicitly assumed that they behave properly under the tensor product. Nonetheless, this is not true in general. Consider the case where we want to apply a complex conjugation on one part in a bipartite state. On the one hand:

$$\begin{aligned} (K \otimes I)e^{i\theta} |xy\rangle &= (K \otimes I)(e^{i\theta} |x\rangle) \otimes |y\rangle \\ &= \overline{e^{i\theta} |x\rangle} \otimes |y\rangle = e^{-i\theta} \overline{|x\rangle} \otimes |y\rangle. \end{aligned} \quad (16)$$

On the other hand:

$$\begin{aligned} (K \otimes I)e^{i\theta} |xy\rangle &= (K \otimes I)|x\rangle \otimes (e^{i\theta} |y\rangle) \\ &= \overline{|x\rangle} \otimes (e^{i\theta} |y\rangle) = e^{i\theta} \overline{|x\rangle} \otimes |y\rangle, \end{aligned} \quad (17)$$

a contradiction! In reality, there is no mathematically consistent direct product of a linear and antilinear operator within the category of complex linear spaces [43]. However, the equality holds up to a phase, a norm one complex number, suggesting that the difficulty disappears when moving to mixed-state quantum mechanics and completely positive maps. As we have seen above, in mixed-state quantum mechanics,

antiunitaries are represented by a transposition followed by a unitary. As this representation is linear on the space of density matrices, there is no longer a *mathematical* problem of mixing unitaries and antiunitaries; a map like

$$(\mathcal{A} \otimes \mathcal{I})(\rho \otimes \sigma) = U\rho^T U^\dagger \otimes \sigma \quad (18)$$

is well-defined. Therefore, we can establish a suitable process theory and start looking for a graphical representation of the antiunitary maps. Moreover, we already know that antiunitary maps decompose into $\mathcal{A} = \mathcal{U} \circ \mathcal{T}$, where \mathcal{U} is a unitary adjoint action, and \mathcal{T} is the transposition map. Consequently, the graphical study of antiunitaries reduces to adding the transposition as a new operation in mixed-states quantum mechanics.

There is nonetheless a *physical* problem that remains: the transpose map is a positive (P) but not completely positive (CP) map. For this reason, it is not a valid quantum evolution; it does not map density matrices to density matrices when applied locally. It can be seen, for example, when applying a transpose on one part of a Bell state: the resulting matrix is no longer positive semi-definite; hence it cannot be a valid quantum state. Thus, adding the transposition requires one to pay the price of leaving the category of CP maps for a larger category of superoperators.

3.2 Hermiticity-Preserving Superoperators

Luckily, the sub-prop of **Lin** freely spanned by tensor and composition of CP maps with \mathcal{T} admits a nice description as Hermiticity-preserving superoperators.

Definition 13 (Hermiticity-Preserving superoperators). *A superoperator \mathcal{H} is said to be **Hermiticity-Preserving (HP)** if for all operator ρ :*

$$\rho = \rho^\dagger \quad \Rightarrow \quad \mathcal{H}(\rho) = \mathcal{H}(\rho)^\dagger. \quad (19)$$

We denote by **HP** the prop whose arrows $n \rightarrow m$ are the HP superoperators in $\mathcal{M}_{2^n \times 2^n}(\mathbb{C}) \rightarrow \mathcal{M}_{2^m \times 2^m}(\mathbb{C})$.

Contrary to positive superoperators, HP superoperators are stable by tensor product. Hence there is no need to consider a notion of “completely HP” superoperator. We have **Pure** \subset **CP** \subset **HP** \subset **Lin**, as represented in Figure 1.

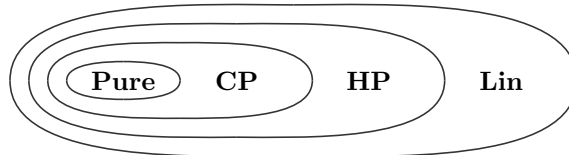


Figure 1: Inclusion of the different props.

Moreover, **HP** has the following property:

Theorem 2 (De Pillis [18], Proposition 1.2). *The Choi-Jamiołkowski isomorphism maps HP maps to Hermitian matrices.*

In category-theoretical terms, this mapping corresponds to the fact that **HP** is a compact closed subcategory of **Lin**, with the same objects but morphisms restricted to HP maps. Thus, to each map $n \rightarrow m$ in **HP**, we can associate a unique state $0 \rightarrow n + m$. This duality allows to show results on HP maps while manipulating only Hermitian matrices.

We can sum up the process-state duality between the different props as follows:

Prop	Processes	States
Pure	pure	rank-1 positive Hermitian matrices
CP	CP	positive Hermitian matrices
HP	HP	Hermitian matrices
Lin	linear	matrices

One may have expected the positive maps to form a sub-prop in-between CP and HP. However, they do not since P maps are not stable by tensor composition. Note for completeness that the corresponding states are the *positive on pure tensor* matrices [34].

4 ZW[†]-Calculus

We choose the ZW-Calculus [14, 23] as the starting graphical language for our extension, as it features a natural normal form that will help us get completeness. For those who already the language, our extension consists in adding a $1 \rightarrow 1$ generator – which we may consider as an edge type as shown in the following – depicted as $\begin{array}{c} | \\ \vdash \\ | \end{array}$.

One ZW-calculus is defined for each commuting subring of \mathbb{C} [24]. This ring defines the domain in which some parameters will live. This generalisation allows us to represent different fragments of quantum computations, each time with a complete equational theory. For simplicity, we fix a subring R right away, such that $\frac{1}{2} \in R$, and take the liberty not to recall it every time. We then define \mathbf{Hilb}_R as the subcategory of \mathbf{Hilb} where maps have coefficients in R . We similarly define \mathbf{Lin}_R and \mathbf{HP}_R .

4.1 Diagrams

The qubit \mathbf{ZW}^\dagger -Calculus is a \dagger -compact prop as defined above, with the following set of generators in string-diagram representation:

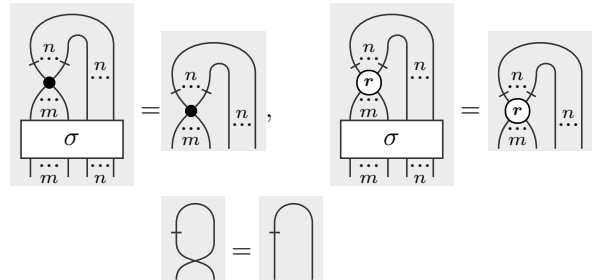


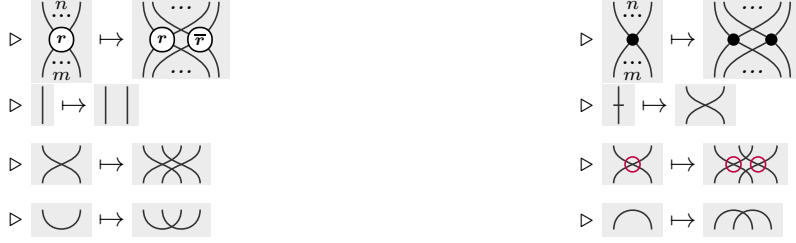
The dagger \dagger acts on the generators as follows:



and leave the other generators unchanged.

A property of these generators is that diagrams built from them can be deformed at will. More formally the tick, the white nodes and black nodes are *flexsymmetric* [6], meaning that for any permutations σ encoded by swaps we have:




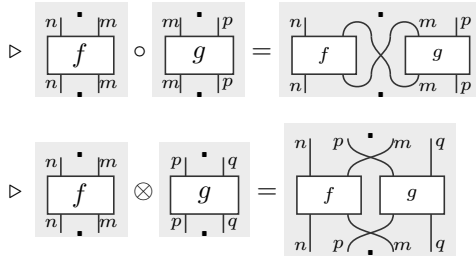


As explained previously, it is possible to give the semantic of \mathbf{ZW}^\dagger -diagrams as $\llbracket \text{Unzip}(\cdot) \rrbracket$, and this is actually enough to define semantical equivalence between diagrams.

4.2.2 Superoperators

Here we first wrap \mathbf{ZW} -diagrams into a category that allows us to define superoperators. It turns out that the so-called “Int”-construction (see e.g. [2]) is well suited.

Definition 15. We define the category $\mathbf{Int}(\mathbf{ZW})$ as the prop whose arrows $n \rightarrow m$ are $n + m \rightarrow n + m$ \mathbf{ZW} -diagrams, the identity (on 1) is , and the compositions are defined as follows:



The \bullet symbols in the definition above are visual cues to avoid specifying the domain and codomain of each diagram (which would otherwise be necessary, as we have to distinguish between e.g. $1 \rightarrow 1$ from $0 \rightarrow 2$ morphisms). This construction forms a monoidal category [33].

Intuitively, time flows from left to right in $\mathbf{Int}(\mathbf{ZW})$ -diagrams, and \bullet separates the inputs, on its left, from the outputs on its right. It is possible to turn $\mathbf{Int}(\mathbf{ZW})[n, m]$ -diagrams into usual $\mathbf{ZW}[n+m, n+m]$ -diagrams through functor ι , which, graphically, simply removes the \bullet symbols.

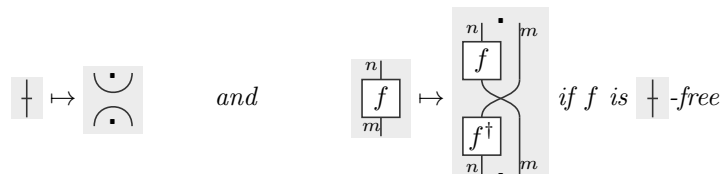
We may define an interpretation $\llbracket \cdot \rrbracket_{\mathbf{Lin}}$ of such diagrams, this time as superoperators, i.e. $\llbracket \cdot \rrbracket_{\mathbf{Lin}} : \mathbf{Int}(\mathbf{ZW}) \rightarrow \mathbf{Lin}$, as:

$$\left[\left[\begin{array}{c|c} n & \bullet \\ \hline f \\ \hline n & \bullet \\ m & \bullet \end{array} \right] \right]_{\mathbf{Lin}} := \rho \mapsto \left[\left[\begin{array}{c|c} n & m \\ \hline D_\rho \\ \hline n & m \end{array} \right] \right]$$

where $\rho \in \mathcal{M}_{2^n \times 2^n}(R)$ and D_ρ is any \mathbf{ZW} -diagram such that $\llbracket D_\rho \rrbracket = \rho$. Such a diagram always exists by universality of \mathbf{ZW} -diagrams [24], and its choice does not change the result of the interpretation.

We may now define how to understand \mathbf{ZW}^\dagger -diagrams as $\mathbf{Int}(\mathbf{ZW})$ -diagrams:

Definition 16. We define the monoidal functor $\mathbf{HP} : \mathbf{ZW}^\dagger \rightarrow \mathbf{Int}(\mathbf{ZW})$ that acts on generators as follows:



We may illustrate the use of \cdot by emphasizing that:

$$\begin{aligned} \text{HP} \left(\begin{array}{c} \text{---} \\ \cup \\ \text{---} \end{array} \right) &= \begin{array}{c} \cdot \\ \text{---} \\ \cup \\ \text{---} \\ \cdot \end{array} & \text{HP} \left(\begin{array}{c} \text{---} \\ \dagger \\ \text{---} \end{array} \right) &= \begin{array}{c} \cdot \\ \text{---} \\ \dagger \\ \text{---} \\ \cdot \end{array} \\ \\ \text{HP} \left(\begin{array}{c} \text{---} \\ \cup \\ \text{---} \end{array} \right) &= \begin{array}{c} \text{---} \\ \cup \\ \cdot \\ \text{---} \\ \cdot \end{array} \end{aligned}$$

The map's name HP stands for ‘‘Hermiticity-preserving’’, since, as we will see in the following, states of \mathbf{ZW}^\dagger represent Hermitian operators, and morphisms represent superoperators that preserve Hermiticity.

We can show that the two semantics above are isomorphic. There exists an invertible map Ψ such that the following diagram commutes:

$$\begin{array}{ccccc} \mathbf{ZW}^\dagger & \xrightarrow{\text{Unzip}} & \text{Unzip}(\mathbf{ZW}^\dagger) & \hookrightarrow & \mathbf{ZW} & \xrightarrow{[\![\cdot]\!] } & \mathbf{Hilb}_R \\ \text{HP} \downarrow & & & \searrow \Psi & & & \uparrow \cong \\ \text{HP}(\mathbf{ZW}^\dagger) & \hookrightarrow & \text{Int}(\mathbf{ZW}) & \xrightarrow{[\![\cdot]\!]_{\text{Lin}}} & \mathbf{Lin}_R & & \end{array} \quad (20)$$

We define Ψ on all $\mathbf{ZW} f : 2n \rightarrow 2m$, and Ψ^{-1} as:

$$\begin{aligned} \Psi \left(\begin{array}{c} \text{---} \\ \text{---} \\ \boxed{f} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right) &:= \begin{array}{c} \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{array} \\ \Psi^{-1} \left(\begin{array}{c} \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{array} \right) &:= \begin{array}{c} \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{array} \end{aligned}$$

Proposition 3. *The above functors Ψ and Ψ^{-1} are inverses of one another.*

It is now possible to combine the standard interpretation $[\![\cdot]\!]$ of \mathbf{ZW} -diagrams with either Unzip or HP to define a standard interpretation $[\![\cdot]\!]^\dagger$ for \mathbf{ZW}^\dagger diagrams. Regarding the upcoming universality of the language, it is more natural to use HP. Hence we choose $[\![\cdot]\!]^\dagger := [\![\text{HP}(\cdot)]\!]_{\text{Lin}}$.

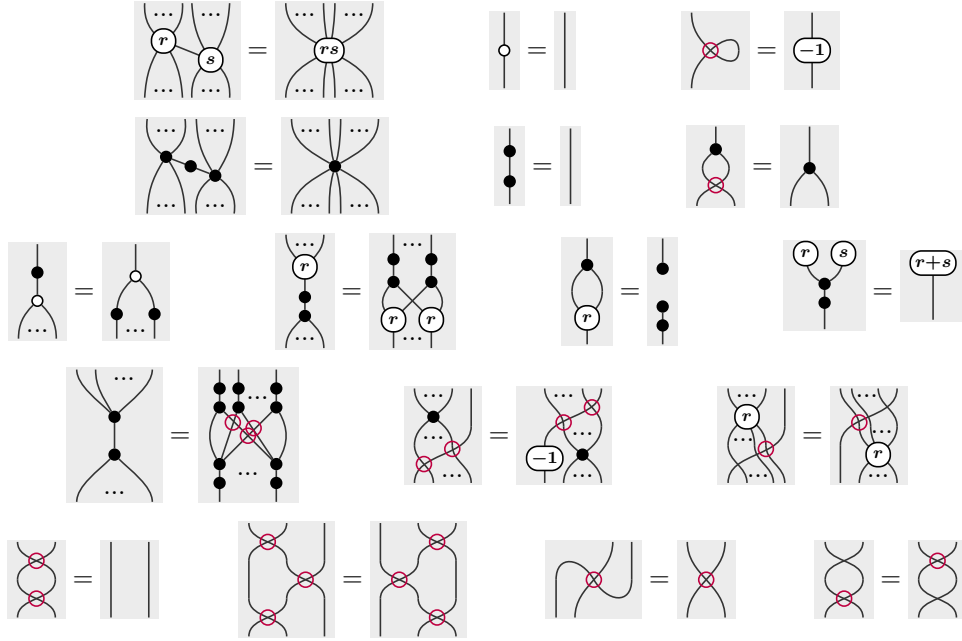
As we will see in the following, all the equations of \dagger -compactness preserve the semantics: if $D_1 = D_2$ then $[\![D_1]\!]^\dagger = [\![D_2]\!]^\dagger$. The reciprocal is the problem of *completeness*: capturing all diagrammatic transformations that keep the semantics unchanged. In the matter, the equations of \dagger -compactness are not enough, we need to add axioms to the equational theory.

4.3 Equational Theory

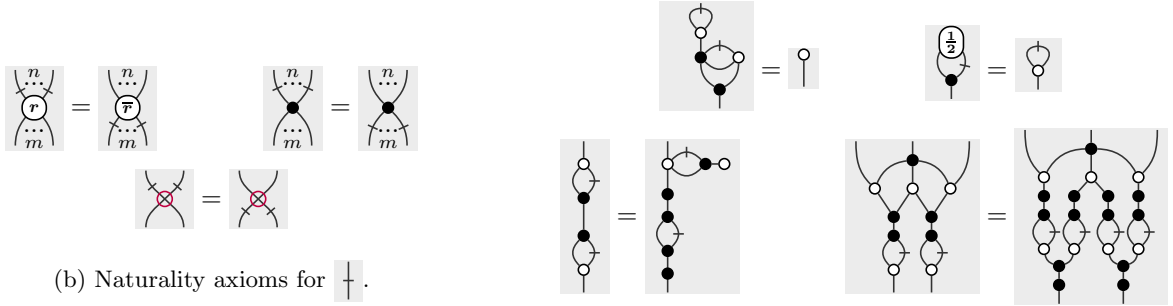
The vanilla \mathbf{ZW} -Calculus already has a complete equational theory for \mathbf{Hilb}_R . It is reminded in Figure 2a.

Each of these equations remains sound in \mathbf{ZW}^\dagger , but they are not enough for completeness. For instance, the following equation: $\begin{array}{c} \text{---} \\ \dagger \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \parallel \\ \text{---} \end{array}$ which is sound since:

$$\text{HP} \left(\begin{array}{c} \text{---} \\ \dagger \\ \text{---} \end{array} \right) = \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} = \begin{array}{c} \cdot \\ \text{---} \\ \cdot \end{array} = \text{HP} \left(\begin{array}{c} \text{---} \\ \parallel \\ \text{---} \end{array} \right)$$



(a) Set of rules for the **ZW**-Calculus, with $r, s \in R$.



(b) Naturality axioms for \dagger .

(c) Additional axioms.

Figure 2: All **ZW**[†] axioms.

where $|\vec{x}_i\rangle = \langle \vec{x}_i |^\dagger = |x_{i_1}, \dots, x_{i_n}\rangle$, $x_{i_j} \in \{0, 1\}$ (similarly for \vec{y}), and such that, between the white node with parameter $\lambda_i/2$ and the k -th output, there is:

▷ an edge \parallel if $x_{i_k} = 1$

▷ a ticked edge \dashv if $y_{i_k} = 1$

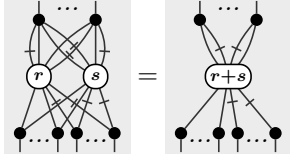
We call this form *normal form*, as it will be used later in the proof of completeness.

Example 1. An arbitrary 1-qubit Hermitian operator will be mapped as follows:

$$\mathcal{N}\left(\begin{pmatrix} a & \bar{b} \\ b & c \end{pmatrix}\right) = \begin{array}{c} \text{Diagram showing a central node connected to four nodes labeled } a/2, b/2, \bar{b}/2, \text{ and } c/2. \end{array}$$

Although the normal form defined above behaves well for the proof of completeness, it seems not to use the Hermiticity of the morphism. In particular, for any term $\lambda_i |\vec{x}_i\rangle\langle \vec{y}_i|$ with $\vec{x}_i \neq \vec{y}_i$ in the sum, there should be another term $\lambda_j |\vec{x}_j\rangle\langle \vec{y}_j|$ such that $\lambda_j = \bar{\lambda}_i$, $\vec{x}_j = \vec{y}_i$ and $\vec{y}_j = \vec{x}_i$. In Example 1, such a pair would be the terms with coefficients b and \bar{b} . Any such pair can be merged into a single term in the normal form, using the naturality axiom for Z-spiders, and the following lemma:

Lemma 1.



When all such simplifications are done, the resulting diagram corresponds to the following way to write a Hermitian matrix exploiting its symmetry:

$$H = \sum_{i=1}^m \lambda_i |\vec{x}_i\rangle\langle \vec{y}_i| = \sum_{i=1}^{\frac{m+d}{2}} \left(\lambda'_i |\vec{x}_i\rangle\langle \vec{y}_i| + \bar{\lambda}'_i |\vec{y}_i\rangle\langle \vec{x}_i| \right),$$

in which d is the number of diagonal coefficients. Notice in particular that diagonal coefficients are counted twice here, so it is important to keep in mind that diagonal coefficients must be halved in the normal form. Such a diagram will be called the *reduced normal form*.

The map \mathcal{N} creates a \mathbf{ZW}^\dagger -state with the semantics of the starting Hermitian operator, i.e.:

Proposition 6. For all $f \in \mathbf{HP}_R[0, n]$, $\llbracket \mathcal{N}(f) \rrbracket^\dagger = f$.

Corollary 2 (Universality). For any $f \in \mathbf{HP}_R[0, n]$, there exists $D \in \mathbf{ZW}^\dagger$, such that: $\llbracket D \rrbracket^\dagger = f$

6 Capturing Completely Positive Maps

In the last section, we proved that adding the tick generator to the ZW-calculus provides a sound and universal graphical language whose interpretation yields the set of Hermiticity-Preserving maps. Considering that the set of \mathbf{CP}_R maps (the intersection of \mathbf{CP} and \mathbf{HP}_R maps) is a subset of the \mathbf{HP}_R maps, the discard generator should be contained within \mathbf{ZW}^\dagger . In this section, we show how to represent specifically the \mathbf{CP}_R maps and how the axioms for the naturality of \dashv are essentially enough for completeness in that fragment.

Discard should have the following images in unzip and HP:

$$\text{unzip} \left(\underline{\underline{=}} \right) = \cup \quad \text{HP} \left(\underline{\underline{=}} \right) = \begin{array}{|c} \bullet \\ \hline \bullet \end{array}.$$

This generator can be represented in \mathbf{ZW}^\dagger as:

$$\text{HP} \left(\begin{array}{c} \circ \\ \uparrow \\ \circ \end{array} \right) = \begin{array}{|c} \bullet \\ \hline \bullet \\ \hline \bullet \end{array} = \begin{array}{|c} \bullet \\ \hline \bullet \end{array} = \text{HP} \left(\underline{\underline{=}} \right).$$

Reference [7] also provides a simple set of rules concerning $\underline{\underline{=}}$ that makes the extended language complete for CP maps. When replacing $\underline{\underline{=}}$ by $\begin{array}{c} \circ \\ \uparrow \\ \circ \end{array}$, these equations become:

$$\begin{array}{ccc} \begin{array}{|c} \bullet \\ \hline \circ \\ \hline e^{i\alpha} \end{array} = \begin{array}{|c} \bullet \\ \hline \bullet \end{array} & \begin{array}{|c} \bullet \\ \hline \circ \\ \hline \bullet \end{array} = \begin{array}{|c} \bullet \\ \hline \bullet \end{array} & \begin{array}{|c} \bullet \\ \hline \circ \\ \hline e^{i\alpha} \\ \hline \circ \\ \hline \bullet \end{array} = \begin{array}{|c} \bullet \\ \hline \circ \\ \hline \bullet \end{array} \\ \\ \begin{array}{|c} \bullet \\ \hline \circ \\ \hline \bullet \end{array} = \begin{array}{|c} \bullet \\ \hline \bullet \end{array} & \begin{array}{|c} \bullet \\ \hline \circ \\ \hline \bullet \end{array} = \begin{array}{|c} \bullet \\ \hline \bullet \end{array} & \end{array} \quad (21)$$

Proposition 7. *If half phases are allowed in R (i.e. $e^{i\alpha} \in R \iff e^{i\frac{\alpha}{2}} \in R$), the rules of Figure 2a and Figure 2b are enough to derive the above equations. If instead we only have $e^{i\frac{\pi}{4}} \in R$, then adding the axiom*

$$\begin{array}{|c} \bullet \\ \hline \circ \\ \hline \frac{1}{2} \end{array} = \begin{array}{|c} \bullet \\ \hline \circ \end{array} \text{ is enough.}$$

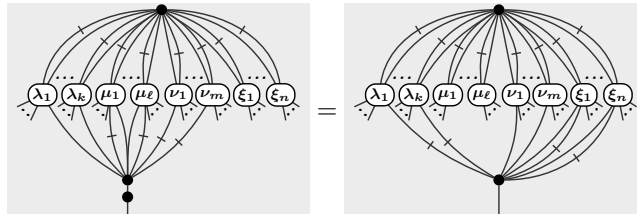
7 Completeness

We show here the central result of the paper: the axiomatisation given in Figures 2a, 2b and 2c entirely captures semantical equivalence. That is to say, any two diagrams with the same interpretation can be turned into one another only by local application of the diagram transformations given in this axiomatisation.

To show this result, we reuse the normal form defined in Section 5. The normal form will constitute the canonical representative of an equivalence class. If we show that any diagram can be put in normal form, we show as a consequence that any two diagrams with the same semantics can be rewritten into one another. We shall prove that all the generators can be put in normal form, and that all compositions of diagrams in normal form can be put in normal form.

First, we show a pretty handy transformation on normal forms, adapted from [24]:



Lemma 2 (Negation). *Given a diagram in (reduced) normal form, the diagram obtained by adding (or removing) \bullet to one of the outputs can be rewritten in (reduced) normal form by independently complementing the $\begin{array}{|c} \bullet \\ \hline \bullet \end{array}$ and $\begin{array}{|c} \bullet \\ \hline \circ \end{array}$ connections of that output to the white vertices, in the following sense:*




This lemma proves useful in particular in the upcoming proof of tensor product of normal forms:


Proposition 8. *The tensor product of two diagrams in normal form can be put in normal form.*



For showing the result on compositions of normal form, as well as for generators, the following two lemmas prove useful:


Lemma 3. *Applying either  or  on an output of a diagram in normal form gives a diagram that can be put in normal form.*

Lemma 4. *Applying  to a pair of outputs of a diagram in normal form gives a diagram that can be put in normal form.*

We can now prove the following:

Proposition 9. *The sequential composition of diagrams in normal form (which, through the process-state duality, amounts to several applications of  to pairs of outputs) gives a diagram that can be put in normal form.*

Proof of Proposition 9. One can easily show that  = . Then, using Lemmas 2, 4 and 3, one can

turn the diagram obtained by application of  to a pair of outputs into normal form. □

Compositions of normal forms can therefore be put in normal form. It now remains to show that the generators of \mathbf{ZW}^\dagger -diagrams can themselves be put in normal form:

Proposition 10. *The generators of the \mathbf{ZW}^\dagger -Calculus can be put in normal form.*

The main result of the paper follows:

Theorem 3. *The axiomatisation in Figures 2a, 2b and 2c is complete for Hermiticity-preserving operators, i.e.:*

$$\forall D_1, D_2 \in \mathbf{ZW}^\dagger, \quad \llbracket D_1 \rrbracket^\dagger = \llbracket D_2 \rrbracket^\dagger \iff \mathbf{ZW}^\dagger \vdash D_1 = D_2$$

Proof. Since all generators can be put in normal form, and all compositions of diagrams in normal form can be put in normal form, any diagram can be put in normal form. Since the rewrite rules are sound, the diagram in normal form has the same semantics as the initial one. By the uniqueness of the normal form, any two diagrams with the same semantics can be put in the same normal form, showing that the two can be turned into one another. □

8 Applications

We conclude by providing some illustrations of the possibilities offered by our graphical language. These are brief sketches of applications whose thorough development could be the subject of further investigations.

8.1 Scalar Product and Dagger

In categorical quantum mechanics, the scalar product is often computed using the cup. However, the corresponding bilinear form is strictly linear in both components and thus does not match the usual Hermitian scalar product of the Hilbert space, which has an antilinear component. This is fine as long as we consider processes in the real-linear subspace. It amounts to identifying the underlying Hilbert space with its dual.

The problem is that, by doing so, the cap and the maximally entangled states get confused into a single equation. Nevertheless, if observed carefully, the cap ‘connects’ the Hilbert space with its dual (which should be represented by a complex conjugated copy of itself since the inner product is sesquilinear). In contrast, the maximally entangled state ‘connects’ the Hilbert space with a copy of itself (i.e. it is a bipartite state).

The two are different on non-real vectors. Consider, for example, the +1 eigenstate of the Y matrix, $|i\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. In single wire, the cap of two such elements is different from their inner product,

$$\langle i | \otimes \langle i | \circ \left(\sum_{k=0}^1 \frac{1}{\sqrt{2}} |k\rangle \otimes |k\rangle \right) = 0 \neq \langle i | i \rangle = 1. \quad (22)$$

In the above, the $|i\rangle$ and the cap were expressed in their usual vector form in the computational basis, $\langle i | \otimes \langle i | = \frac{1}{2} (1 \ -i \ -i \ -1)$, and $\left(\sum_k \frac{1}{\sqrt{2}} |k\rangle \otimes |k\rangle \right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$. Diagrammatically:

$$\langle i | \otimes \langle i | \text{ (cap)} = \langle i | \otimes \langle i | \text{ (tick)} = \langle -1 | = 0 \neq \langle i | i \rangle = 1$$

In doubled, this becomes:

$$\begin{aligned} \text{Tr} \left\{ (|i\rangle\langle i| \otimes |i\rangle\langle i|)^\dagger \cdot \phi^+ \right\} &= \text{Tr} \left\{ |i\rangle\langle i|^\dagger \cdot |i\rangle\langle i|^T \right\} = 0 \\ &\neq \text{Tr} \left\{ |i\rangle\langle i|^\dagger \cdot |i\rangle\langle i| \right\} = 1. \end{aligned} \quad (23)$$

The usual workaround for this problem, is to label direct and dual Hilbert spaces using arrows, resulting in ‘Hairy Spiders’ [15, §8.6.3], which were pioneered in [13, 17]. These can be awkward to work with, but luckily, the tick provides a way to circumvent the Y basis issue without using hairy spiders. In the doubled theory, the tick has a natural ‘arrow-less’ formulation, allowing us to define a new kind of cup which indeed recovers the scalar product:

$$\left[\left[\begin{array}{c} |x\rangle\langle x| \quad |y\rangle\langle y| \\ \text{cup} \end{array} \right] \right] = |\langle x | y \rangle|^2$$

In the same way that the cap and cup can be used to transpose linear transforms, we can use the tick to represent \dagger of diagrams internally as:

$$\boxed{D^\dagger} = \left[\boxed{D} \right]$$

It can be checked that it satisfies the defining property of the adjoint i.e. $\langle D^\dagger x | y \rangle = \langle x | Dy \rangle$, in the doubled picture:

$$\begin{array}{c} x \quad y \\ \boxed{D^\dagger} \end{array} = \begin{array}{c} x \quad y \\ \boxed{D} \end{array} = \begin{array}{c} x \quad y \\ \boxed{D} \end{array}$$

Hence, the definition of a diagram being unitary becomes:


$$\left[\boxed{D} \boxed{D} \right] = \boxed{D} \boxed{D} = \left[\right]$$

8.2 Proper Choi-Jamiołkowski

The reader may have noticed the transposition in the definition of the reverse direction of the Choi-Jamiołkowski (CJ) isomorphism, eq. (7). It is again a symptom of identifying a Hilbert space with its

8.4 Spin Flip

We end this list of illustrations with an example of a new kind of transformation one can represent using ticks: the spin-flip. This transformation acts on a qubit state ρ as $\rho \mapsto Y\rho^TY$, and can be represented

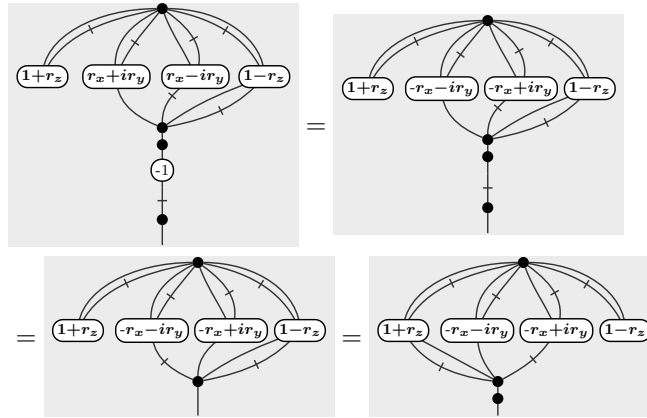
diagrammatically as: .

Recall that a 1-qubit state can be parametrised by a vector in the Bloch ball, given by the tuple $\vec{r} = (r_x, r_y, r_z)$, in the following way:

$$\begin{aligned} \rho &= \frac{I + \vec{r} \cdot \vec{\sigma}}{2} = \frac{I + r_x X + r_y Y + r_z Z}{2} \\ &= \frac{1}{2} \begin{pmatrix} 1 + r_z & r_x - ir_y \\ r_x + ir_y & 1 - r_z \end{pmatrix} \end{aligned}$$

The spin-flip then amounts to a central symmetry of the Bloch sphere, matching each point to its antipodes. In a sense, this is the true (logical) negation of a qubit in the Bloch sphere.

The above morphism is thus supposed to map ρ defined by \vec{r} to ρ' defined by $-\vec{r}$. This can be checked using e.g. the normal form of ρ (ignoring overall scalar of $\frac{1}{4}$):




Spin-flipping offers an example of the peculiar role played by antiunitaries in quantum information. In [22] it was shown that more information could be stored in a pair of antiparallel spin states rather than two parallel ones. This purely quantum phenomenon can be tracked to the fact that, in the former case, the states are related by an antiunitary transformation, the spin-flip. This corroborates the fact that two local operations enriched with an antiunitary generator (in this case, the preparation of the two spin states, followed by the spin-flip of one of them) allows to do more than without the antiunitary (in this case, to succeed a protocol with better efficiency).

9 Discussion

We showed that adding a generator for partial transpose, the tick, is sufficient to extend universal and complete graphical languages for pure quantum operations to one for Hermiticity-preserving operations. In addition, we provided a normal form for diagrams using the ZW-calculus, providing a completeness result. We also extended the doubling construction to include the tick under the name of the Unzip functor and provided a connection to the **Int** functor.

A consequence of our results is a partial answer to the question of completeness for Clifford+T CP maps, left open in [7]. Indeed, when taking $R := \mathbb{Z}[\frac{1}{2}, e^{i\frac{\pi}{4}}]$ (which obviously contains $\frac{1}{2}$) we precisely end up in

the Clifford+T fragment [31]. This means we can now show diagrammatic equality between two Clifford+T CPM diagrams. However, we may have to use the full power of the present axiomatisation and, in the process, have diagrams that are not locally CPMs but merely Hermiticity-preserving operators.

Contrary to [7] we have completeness for the Clifford+T fragment and unrestricted quantum computation (when taking $R := \mathbb{C}$). But, not for the Clifford fragment, which the \mathbf{ZW}^\dagger diagram cannot represent, no matter what ring R we take (since the W-state  itself is outside of the fragment). We leave getting a complete equational theory for Clifford HP maps as an open question.

An important caveat, compared to other more established axiomatisations, is that there does not appear to be a good interpretation of the axioms of Figure 2c. Finding such interpretations, or simplifying the current axioms with ones that do have natural meanings, is left as an open question as well. Another unanswered question in this work is the necessity of the axioms, and consequently the minimality of the equational theory.

As stated above, we managed to provide the completeness result using normal forms –a different method than [7]. The reason is that the positivity of a matrix is challenging to express algebraically. This is why the proof of completeness for \mathbf{CP} uses purification instead of the normal form. Hermiticity is, however, a simpler property, allowing direct proof of completeness using normal forms. The restatement of the proof in terms of purification is left for future works generalising the notion of purification for Hermiticity-preserving operators, paving the way toward a categorical characterisation of \mathbf{HP} via a universal property as done for \mathbf{CP} in [29]. Such uniform characterisation would allow us to extend completeness to other graphical languages like ZX- and ZH-calculi more naturally than by direct translation of the present axioms, and could even prove useful for similar issues in quantum circuits, which are not compact close.

References

- [1] Scott Aaronson and John Watrous. Closed timelike curves make quantum and classical computing equivalent. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2102):631–647, 2009.
- [2] Samson Abramsky, Esfandiar Haghverdi, and Philip J. Scott. Geometry of interaction and linear combinatory algebras. *Mathematical Structures in Computer Science*, 12:625 – 665, 2002.
- [3] 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, 2019.
- [4] John C Baez, Brandon Coya, and Franciscus Rebro. Props in network theory. *arXiv preprint arXiv:1707.08321*, 2017.
- [5] V. Bužek, M. Hillery, and R. F. Werner. Optimal manipulations with qubits: Universal-not gate. *Physical Review A*, 60(4):R2626–R2629, Oct 1999.
- [6] Titouan Carette. *Wielding the ZX-calculus, Flexsymmetry, Mixed States, and Scalable Notations. (Manier le ZX-calcul, flexsymétrie, systèmes ouverts et limandes)*. PhD thesis, University of Lorraine, Nancy, France, 2021.
- [7] Titouan Carette, Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Completeness of Graphical Languages for Mixed States Quantum Mechanics. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 108:1–108:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [8] N. J. Cerf and S. Iblisdir. Quantum cloning machines with phase-conjugate input modes. *Phys. Rev. Lett.*, 87:247903, Nov 2001.

- [9] Giulio Chiribella, Erik Aurell, and Karol Życzkowski. Symmetries of quantum evolutions. *Physical Review Research*, 3(3), jul 2021.
- [10] Man-Duen Choi. Positive Linear Maps on Complex Matrices. *Linear Algebra and its Applications*, 10(3):285–290, 1975.
- [11] Alexandre Clément, Nicolas Heurtel, Shane Mansfield, Simon Perdrix, and Benoit Valiron. A complete equational theory for quantum circuits, 2022.
- [12] 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).
- [13] Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, Apr 2011.
- [14] Bob Coecke and Aleks Kissinger. The compositional structure of multipartite quantum entanglement. In *Automata, Languages and Programming*, pages 297–308. Springer Berlin Heidelberg, 2010.
- [15] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, Cambridge, 2017.
- [16] Bob Coecke and Simon Perdrix. Environment and Classical Channels in Categorical Quantum Mechanics. *Logical Methods in Computer Science*, Volume 8, Issue 4, Nov 2012.
- [17] Bob Coecke, Simon Perdrix, and Éric Oliver Paquette. Bases in diagrammatic quantum protocols. *Electronic Notes in Theoretical Computer Science*, 218:131–152, Oct 2008.
- [18] John de Pillis. Linear transformations which preserve hermitian and positive semidefinite operators. *Pacific J. Math.*, 23(1):129–137, 1967.
- [19] Qingxiuxiong Dong, Marco Túlio Quintino, Akihito Soeda, and Mio Murao. Implementing positive maps with multiple copies of an input state. *Physical Review A*, 99(5), May 2019.
- [20] Markus Fremps and Eric G. Cavalcanti. Variations on the choi-jamiołkowski isomorphism, 2022.
- [21] Alexei Gilchrist, Daniel R. Terno, and Christopher J. Wood. Vectorization of quantum operations and its use, 2009.
- [22] Nicolas Gisin and Sandu Popescu. Spin flips and quantum information for antiparallel spins. *Physical Review Letters*, 83:432–435, Jul 1999.
- [23] Amar Hadzihasanovic. A diagrammatic axiomatisation for qubit entanglement. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 573–584, Jul 2015.
- [24] Amar Hadzihasanovic. *The Algebra of Entanglement and the Geometry of Composition*. PhD thesis, University of Oxford, 2017.
- [25] 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.
- [26] Sam A. Hill and William K. Wootters. Entanglement of a pair of quantum bits. *Physical Review Letters*, 78:5022–5025, Jun 1997.
- [27] Timothée Hoffreumon and Ognjan Oreshkov. The multi-round process matrix. *Quantum*, 5:384, Jan 2021.

- [28] Michał Horodecki, Paweł Horodecki, and Ryszard Horodecki. Separability of mixed states: necessary and sufficient conditions. *Physics Letters A*, 223(1):1–8, 1996.
- [29] Mathieu Huot and Sam Staton. Quantum channels as a categorical completion. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE, 2019.
- [30] Andrzej Jamiołkowski. Linear transformations which preserve trace and positive semidefiniteness of operators. *Reports on Mathematical Physics*, 3(4):275–278, dec 1972.
- [31] 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.
- [32] Min Jiang, Shunlong Luo, and Shuangshuang Fu. Channel-state duality. *Phys. Rev. A*, 87:022310, Feb 2013.
- [33] André Joyal, Ross Street, and Dominic Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119(3):447–468, 1996.
- [34] M. Kläy, C. Randall, and D. Foulis. Tensor products and probability weights. *International Journal of Theoretical Physics*, 26(3):199–219, 1987.
- [35] M. S. Leifer and Robert W. Spekkens. Towards a formulation of quantum theory as a causally neutral theory of bayesian inference. *Physical Review A*, 88(5), Nov 2013.
- [36] S. Massar. Collective versus local measurements on two parallel or antiparallel spins. *Phys. Rev. A*, 62:040101, Sep 2000.
- [37] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [38] Asher Peres. Separability criterion for density matrices. *Physical Review Letters*, 77:1413–1415, Aug 1996.
- [39] Nicola Pinzani, Stefano Gogioso, and Bob Coecke. Categorical semantics for time travel. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–20. IEEE, 2019.
- [40] Bartosz Regula, Ryuji Takagi, and Mile Gu. Operational applications of the diamond norm and related measures in quantifying the non-physicality of quantum maps. *Quantum*, 5:522, Aug 2021.
- [41] Franz Schwabl. *Advanced Quantum Mechanics*. Springer-Verlag, Berlin Heidelberg, fourth edition, 2008.
- [42] Raymond F. Streater and Arthur S. Wightman. *PCT, Spin and Statistics, and All That*. Princeton Landmarks in Physics. Princeton University Press, Princeton, New Jersey, 2001.
- [43] Armin Uhlmann. Anti- (conjugate) linearity. *Science China Physics, Mechanics & Astronomy*, 59(3), jan 2016.
- [44] Renaud Vilmart. A near-minimal axiomatisation of zx-calculus for pure qubit quantum mechanics. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10, June 2019.
- [45] Steven Weinberg. *The Quantum Theory of Fields*, volume 1. Cambridge University Press, 1995.
- [46] Steven Weinberg. *The Quantum Theory of Fields*, volume 2. Cambridge University Press, 1996.
- [47] Eugene P. Wigner. Ueber die operation der zeitumkehr in der quantenmechanik. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1932:546–559, 1932.

- [48] Eugene P. Wigner. *Group Theory and its application to the quantum mechanics of atomic spectra*, volume 5 of *Pure and Applied Physics*. Academic Press, 1959.
- [49] Eugene P. Wigner. Normal form of antiunitary operators. In Arthur S. Wightman, editor, *The Collected Works of Eugene Paul Wigner: Part A: The Scientific Papers*, pages 551–555. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.
- [50] William K. Wootters. Entanglement of formation of an arbitrary state of two qubits. *Physical Review Letters*, 80:2245–2248, Mar 1998.

A Proofs of Section 4

Proof of Proposition 3. That Ψ and Ψ^{-1} are inverses of one another is direct, one merely undoes the permutation of inputs/outputs from the other one. For the rest, we may want to prove that $\Psi \circ \text{unzip} = \text{HP}$. We show this by induction on the structure of the diagrams:

▷ When $D = D_1 \otimes D_2$:

$$\begin{aligned}
 & \Psi \left(\text{unzip} \left(\begin{array}{|c|c|} \hline \dots & \dots \\ \hline D_1 & D_2 \\ \hline \dots & \dots \\ \hline \end{array} \right) \right) \\
 &= \Psi \left(\begin{array}{|c|c|} \hline \dots & \dots \\ \hline \text{unzip}(D_1) & \text{unzip}(D_2) \\ \hline \dots & \dots \\ \hline \end{array} \right) \\
 &= \begin{array}{|c|c|} \hline \dots & \dots \\ \hline \text{unzip}(D_1) & \text{unzip}(D_2) \\ \hline \dots & \dots \\ \hline \end{array} \\
 &= \begin{array}{|c|c|} \hline \dots & \dots \\ \hline \text{HP}(D_1) & \text{HP}(D_2) \\ \hline \dots & \dots \\ \hline \end{array} = \text{HP} \left(\begin{array}{|c|c|} \hline \dots & \dots \\ \hline D_1 & D_2 \\ \hline \dots & \dots \\ \hline \end{array} \right)
 \end{aligned}$$

▷ When $D = D_2 \circ D_1$:

$$\begin{aligned}
 & \Psi \left(\text{unzip} \left(\begin{array}{|c|} \hline \dots \\ \hline D_1 \\ \hline \dots \\ \hline D_2 \\ \hline \dots \\ \hline \end{array} \right) \right) = \Psi \left(\begin{array}{|c|} \hline \dots \\ \hline \text{unzip}(D_1) \\ \hline \dots \\ \hline \text{unzip}(D_2) \\ \hline \dots \\ \hline \end{array} \right) \\
 &= \begin{array}{|c|} \hline \dots \\ \hline \text{unzip}(D_1) \\ \hline \dots \\ \hline \text{unzip}(D_2) \\ \hline \dots \\ \hline \end{array} \\
 &= \begin{array}{|c|} \hline \dots \\ \hline \text{unzip}(D_1) \\ \hline \dots \\ \hline \text{unzip}(D_2) \\ \hline \dots \\ \hline \end{array}
 \end{aligned}$$

$$= \text{HP}(D_1) \dots \text{HP}(D_2) = \text{HP} \left(\begin{array}{c} \dots \\ D_1 \\ \dots \\ D_2 \\ \dots \end{array} \right)$$

▷ For \dagger -free generator g :

$$\begin{aligned} \Psi \left(\text{unzip} \left(\begin{array}{c} \dots \\ g \\ \dots \end{array} \right) \right) &= \Psi \left(\begin{array}{c} \dots \\ g \quad \bar{g} \\ \dots \end{array} \right) \\ &= \dots \begin{array}{c} \dots \\ g \quad \bar{g} \\ \dots \end{array} \dots = \begin{array}{c} \dots \\ g \\ \dots \\ g^\dagger \\ \dots \end{array} = \text{HP} \left(\begin{array}{c} \dots \\ g \\ \dots \end{array} \right) \end{aligned}$$

where \bar{g} stands for the “conjugate” of the generator g : it is g itself for all generators except for the Z-spider, in which case the parameter r becomes \bar{r} . Exchanging inputs and outputs performs the transpose, hence g^\dagger is obtained by composition of inputs-outputs swapping and conjugation.

▷ Finally, for \dagger :

$$\begin{aligned} \Psi \left(\text{unzip} \left(\dagger \right) \right) &= \Psi \left(\begin{array}{c} \times \end{array} \right) = \begin{array}{c} \text{---} \\ \diagup \quad \diagdown \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \diagdown \quad \diagup \\ \text{---} \end{array} \\ &= \text{HP} \left(\dagger \right) \end{aligned}$$

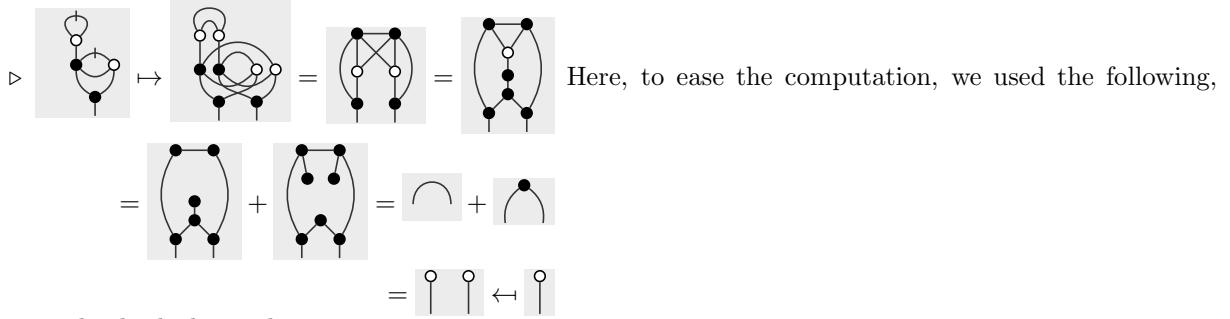
□

Proof of Proposition 4. The axioms of Figure 2a are obviously sound, as they are in **ZW** (through unzip for instance, we simply end up with two copies of the same diagram). It is also pretty straightforward to see that the axioms of 2b are sound using unzip. We can show the one with the Z-spider for example:

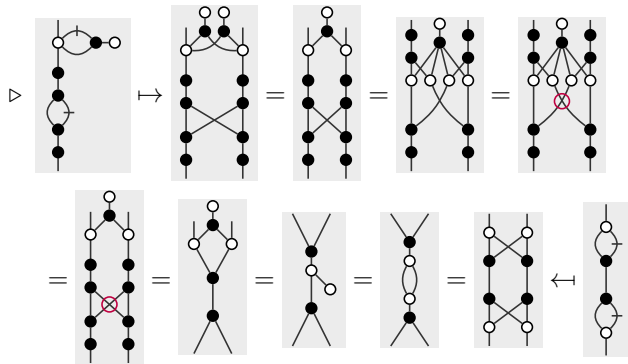
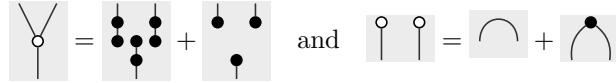
$$\begin{array}{c} \begin{array}{c} \dots \\ n \\ \dots \\ r \\ \dots \\ m \end{array} \end{array} \mapsto \begin{array}{c} \dots \\ \dots \\ r \quad \bar{r} \\ \dots \\ \dots \end{array} = \begin{array}{c} \dots \\ \dots \\ \bar{r} \quad r \\ \dots \\ \dots \end{array} = \begin{array}{c} \dots \\ \dots \\ \bar{r} \quad \bar{r} \\ \dots \\ \dots \end{array} \leftarrow \begin{array}{c} \dots \\ n \\ \dots \\ \bar{r} \\ \dots \\ m \end{array}$$

The axioms from Figure 2c are the ones that really need a proof:

$$\begin{aligned} \triangleright \begin{array}{c} \frac{1}{2} \\ \text{---} \\ \bullet \end{array} &\mapsto \begin{array}{c} \frac{1}{2} \quad \frac{1}{2} \\ \text{---} \\ \bullet \quad \bullet \end{array} = \begin{array}{c} \frac{1}{2} \quad \frac{1}{2} \\ \text{---} \\ \bullet \quad \bullet \end{array} = \begin{array}{c} \frac{1}{2} \quad \frac{1}{2} \\ \text{---} \\ \bullet \quad \bullet \end{array} = \begin{array}{c} \text{---} \\ \bullet \quad \bullet \end{array} \\ &= \begin{array}{c} \text{---} \\ \bullet \quad \bullet \end{array} = \begin{array}{c} \text{---} \\ \bullet \quad \bullet \end{array} \leftarrow \begin{array}{c} \text{---} \\ \bullet \quad \bullet \end{array} \end{aligned}$$

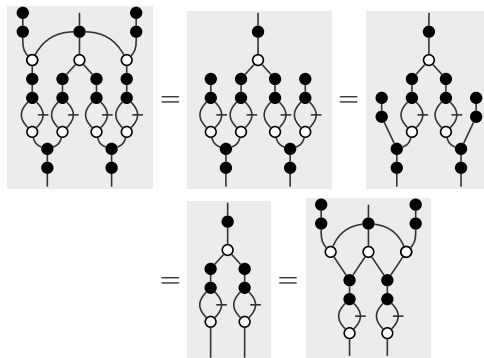


easily checked, equalities:



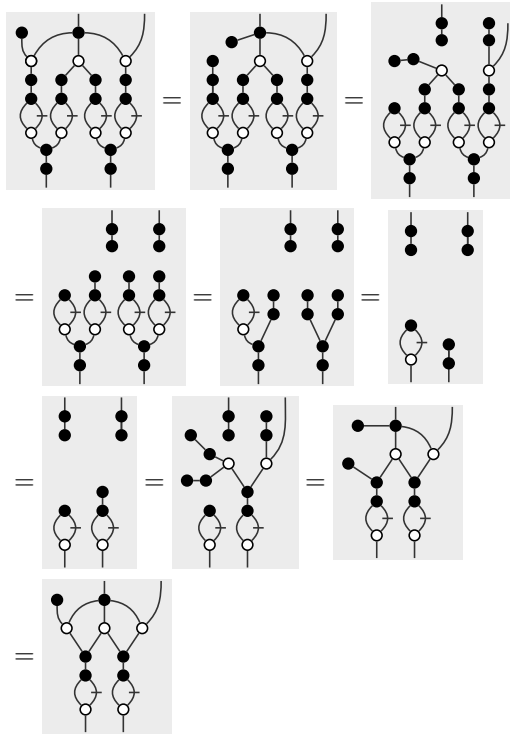
▷ To prove the soundness of the last axiom, we are going to show the equality for every element in a basis that spans the space of some inputs. More precisely, we have to show the equality between two $3 \rightarrow 2$ diagrams. After doubling, these become $6 \rightarrow 4$ diagrams. We are going to show the equality for all classical states spanning the space of the two leftmost and the two rightmost inputs:

- $|00 * * 00\rangle$: Since the two leftmost (resp. the two rightmost) input states are equal, we can reason with the diagram before doubling:

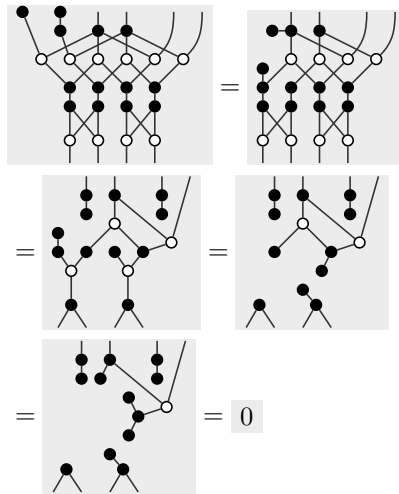


- $|11 * * * * \rangle$: Notice that by symmetry of the diagrams, the derivation for $|* * * * 11\rangle$ is completely

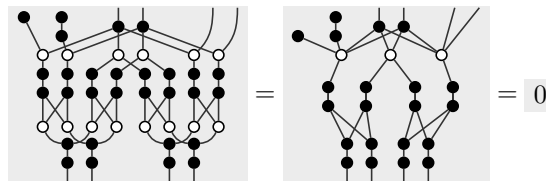
similar. Again, we can reason here with the diagrams before doubling:



– $\mathbf{10}^{***}$: In that case, we are going to show that the obtained diagram is the zero map in each case:



and:

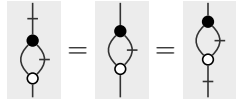


Again, by symmetry, the similar equalities are obtained for inputs $|01****\rangle$, $|****01\rangle$ and $|****10\rangle$.

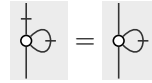
Finally, all the ways to apply a classical state to the two leftmost and the two rightmost inputs end up with the same diagrams on both side. By linearity, this implies that the two diagrams have the same interpretation. □

The following lemmas are fairly direct and are used extensively in the derivations to come:

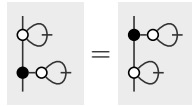
Lemma 7.



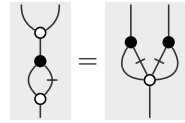
Lemma 8.



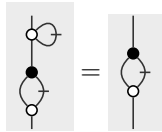
Lemma 9.



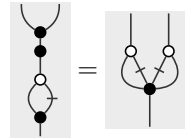
Lemma 10.



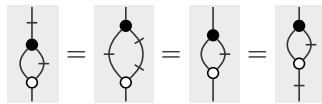
Lemma 11.



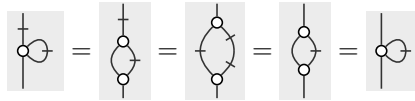
Lemma 12.



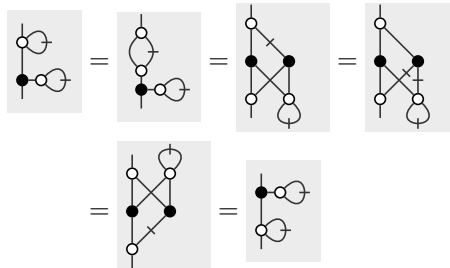
Proof of Lemma 7.



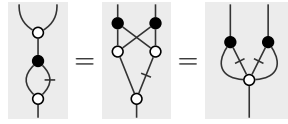
Proof of Lemma 8.



Proof of Lemma 9.

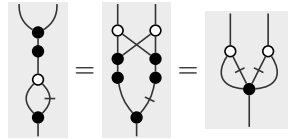


Proof of Lemma 10.



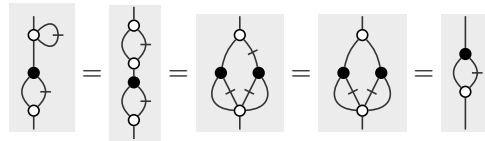
□

Proof of Lemma 12.



□

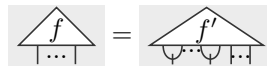
Proof of Lemma 11.



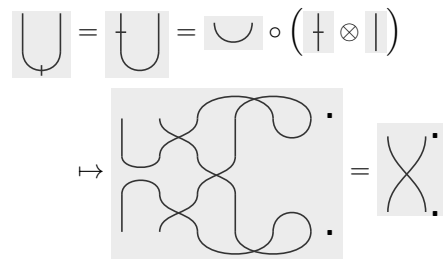
□

B Proofs of Section 5

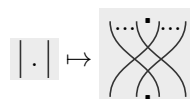
Proof of Proposition 5. Let f be a state in \mathbf{ZW}^\dagger . Suppose it has p occurrences of $\begin{array}{|c} \vdash \\ \hline \end{array}$ (we may try to reduce this number using the equational theory, but this is not a concern). It is then possible to push them out using elementary diagram deformations:



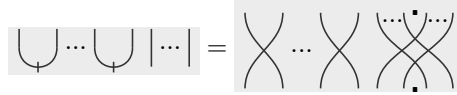
We actually do not have to go through the whole interpretation to show that this is mapped to a Hermitian operator; we only have to look at what HP maps it to. First, we can show that the cup with a tick on it is mapped to the following:



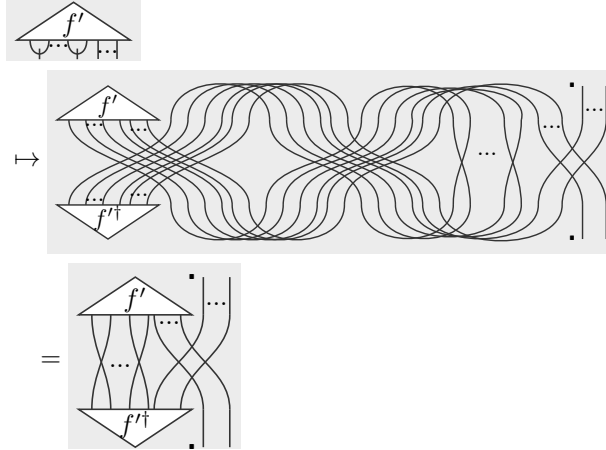
It is direct to check that an n -fold tensor of identities is mapped as follows:



From the previous two mappings, we deduce that their tensor product is mapped to:

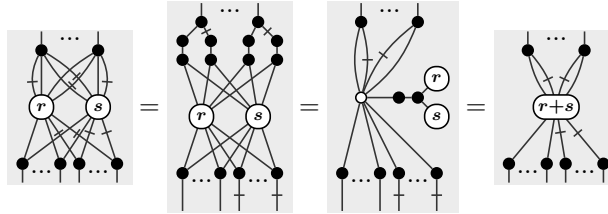


Finally, we get that:



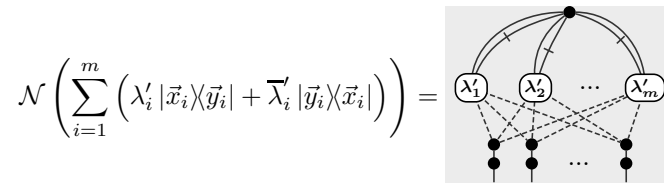
since f' is $\begin{array}{|} \hline \vdots \\ \hline \end{array}$ -free. Now, if we cast this $0 \rightarrow n$ superoperator into the corresponding $(n \rightarrow n)$ operator through ι , we can easily check that the latter is indeed Hermitian in \mathbf{ZW} and hence $\llbracket f' \rrbracket^{\dagger}$ is Hermitian. \square

Proof of Lemma 1.




\square

Proof of Proposition 6. First of all, let us consider that the diagram is already in reduced normal form:

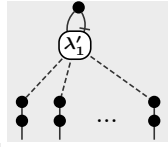


We may prove the result by induction on m .

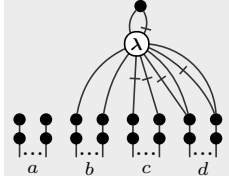
\triangleright **Case $m = 0$:** In this case, the diagram is reduced to: , which indeed represents the 0 map, as $\llbracket \bullet \rrbracket^{\dagger} = 0$.

▷ **Case m = 1:** This case is not necessary, but it will be reused in the following. The diagram in this

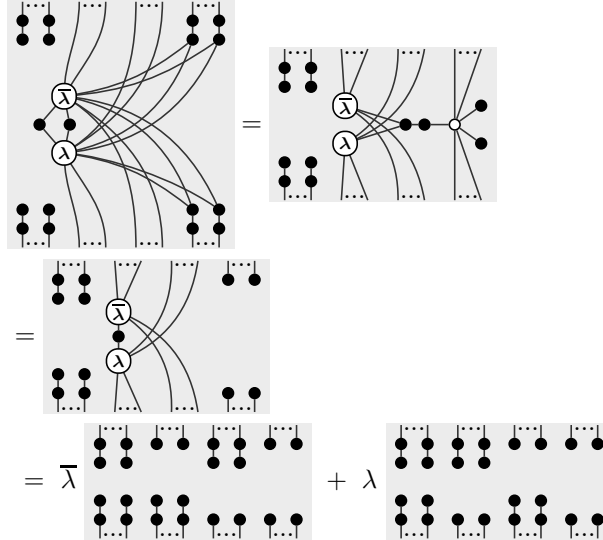
case is reduced to:



. By reordering the outputs, we can write the diagram as follows:



with $\lambda = \lambda'_1$. It is then mapped through HP to:

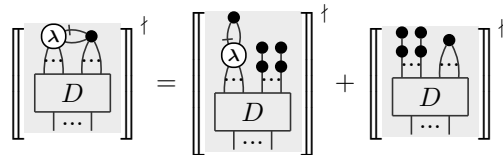


whose interpretation is

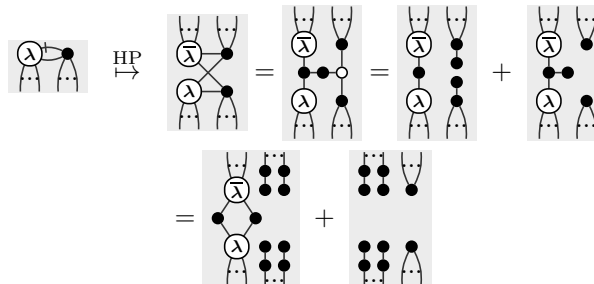
$$\lambda |0^a 1^b 0^c 1^d\rangle\langle 0^a 0^b 1^c 1^d| + \bar{\lambda} |0^a 0^b 1^c 1^d\rangle\langle 0^a 1^b 0^c 1^d|$$


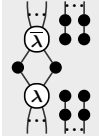
Undoing the permutation we did to group outputs together does put the non-zero coefficients to their designated place.

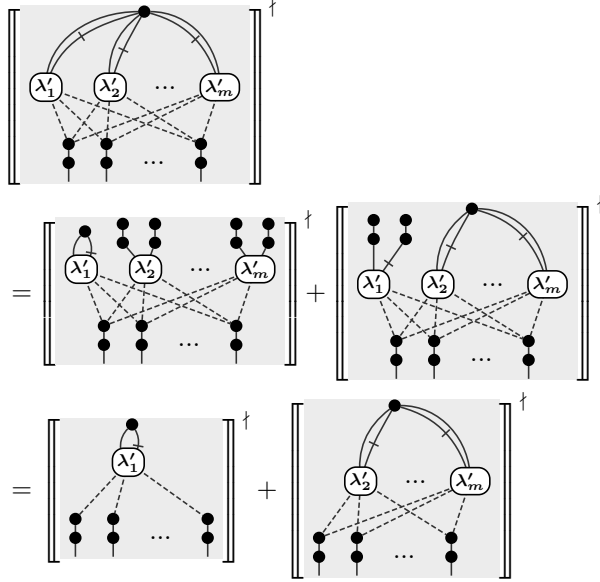
▷ **Case m:** We will use the following identity:



which can be proven as:



and  and . If we apply this to the diagram in normal form, on the first coefficient, we get:



Notice that the left term is the case $m = 1$, and that right term is a diagram in normal form with $m - 1$ white nodes. Hence, we may apply the induction hypothesis on it to get:

$$\begin{aligned}
 & \left[\begin{array}{c} \text{Tree with root and children } \lambda'_1, \lambda'_2, \dots, \lambda'_m \end{array} \right]^\dagger \\
 &= \lambda'_1 |\vec{x}_1\rangle \langle \vec{y}_1| + \overline{\lambda'_1} |\vec{y}_1\rangle \langle \vec{x}_1| \\
 & \quad + \sum_{i=2}^m \left(\lambda'_i |\vec{x}_i\rangle \langle \vec{y}_i| + \overline{\lambda'_i} |\vec{y}_i\rangle \langle \vec{x}_i| \right)
 \end{aligned}$$

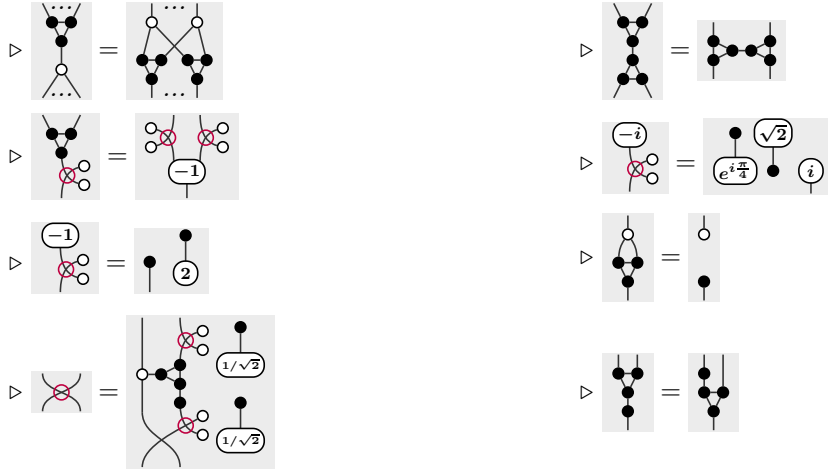
This finishes the proof. □

C Proofs of Section 6

Lemma 13. *If R contains all half angles, then $e^{i\frac{\pi}{4}}, e^{-i\frac{\pi}{4}} \in R$. If $e^{i\frac{\pi}{4}} \in R$, then $\sqrt{2}, \frac{1}{\sqrt{2}} \in R$.*

The following equations are sound and \dagger -free, whence they are derivable in \mathbf{ZW} by completeness, and a fortiori also in \mathbf{ZW}^\dagger :

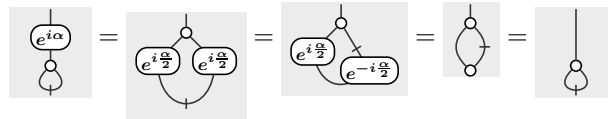




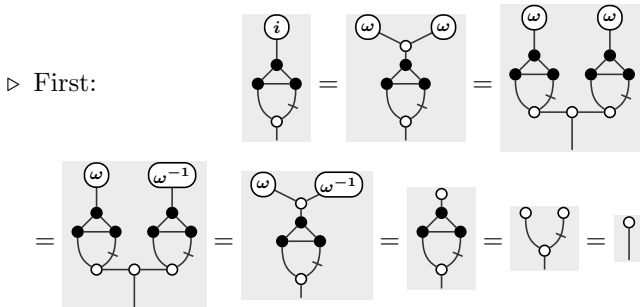
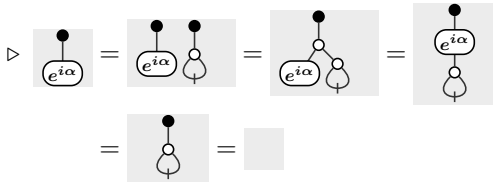
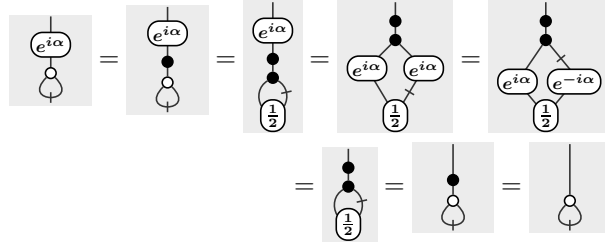
Proof of Proposition 7. If R contains all half angles, $-1 = e^{i\pi} \in R$, $e^{i\frac{\pi}{2}} \in R$ so $e^{i\frac{\pi}{4}} \in R$, hence $e^{-i\frac{\pi}{4}} = (e^{i\frac{\pi}{4}})^7 \in R$. Then $\sqrt{2} = e^{i\frac{\pi}{4}} + e^{-i\frac{\pi}{4}} \in R$ and $\frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2} \in R$.



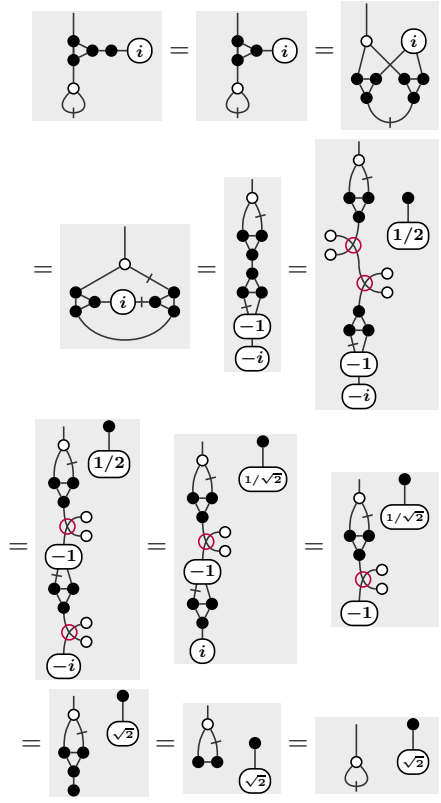
▷ If R contains all half-angles:



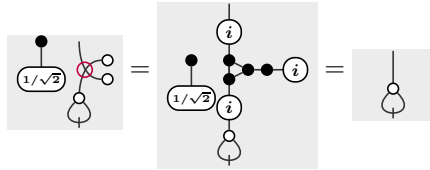
else:



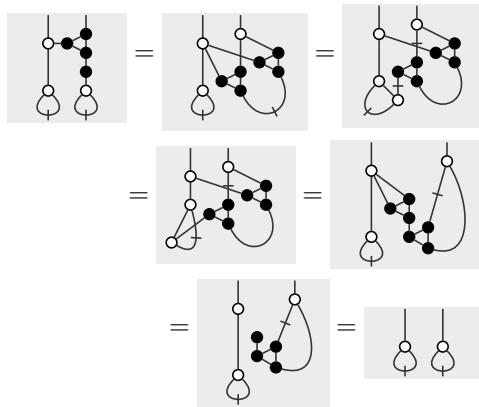
Then:

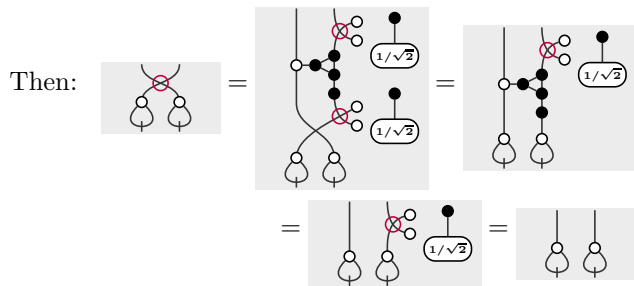


Finally:



▷ First:

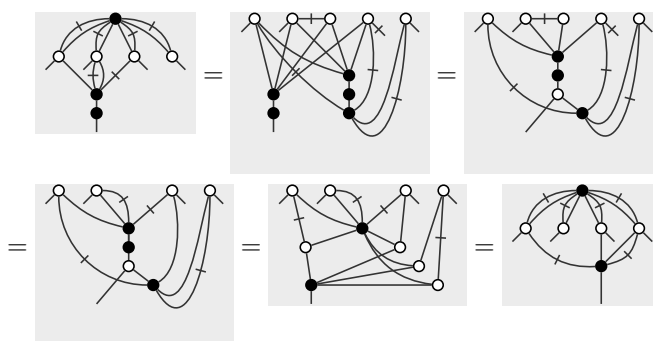




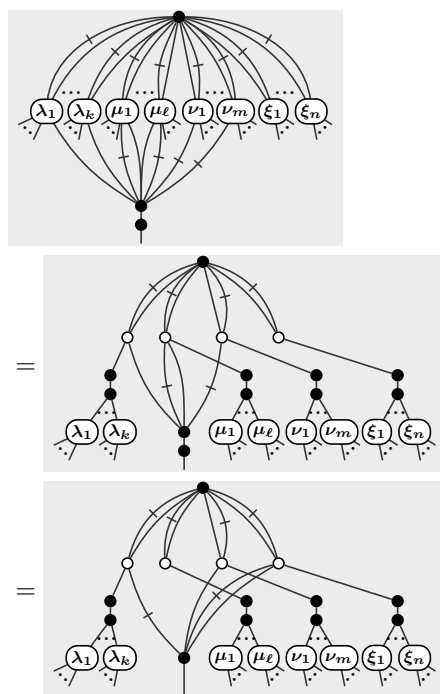
□

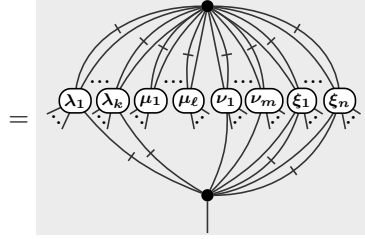
D Proofs of Section 7

Proof of Lemma 2. First:



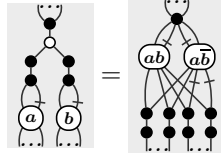
Then:



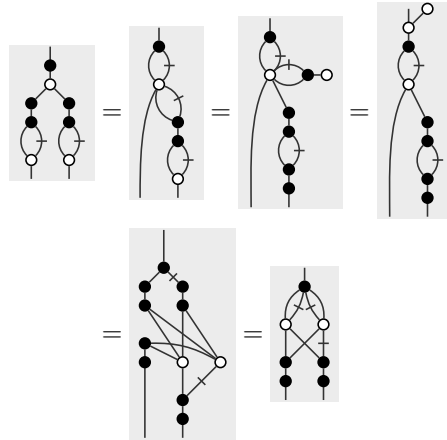


□

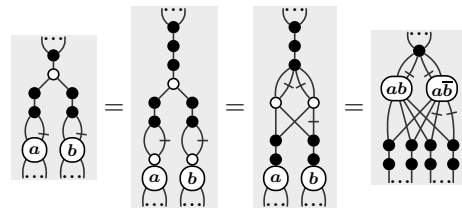
Lemma 14.



Proof of Lemma 14. First:

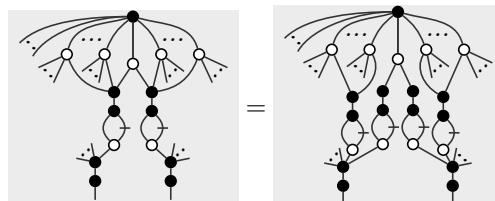


then:

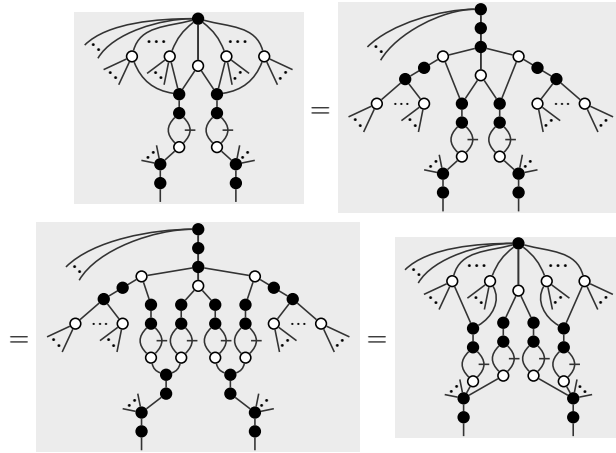


□

Lemma 15.

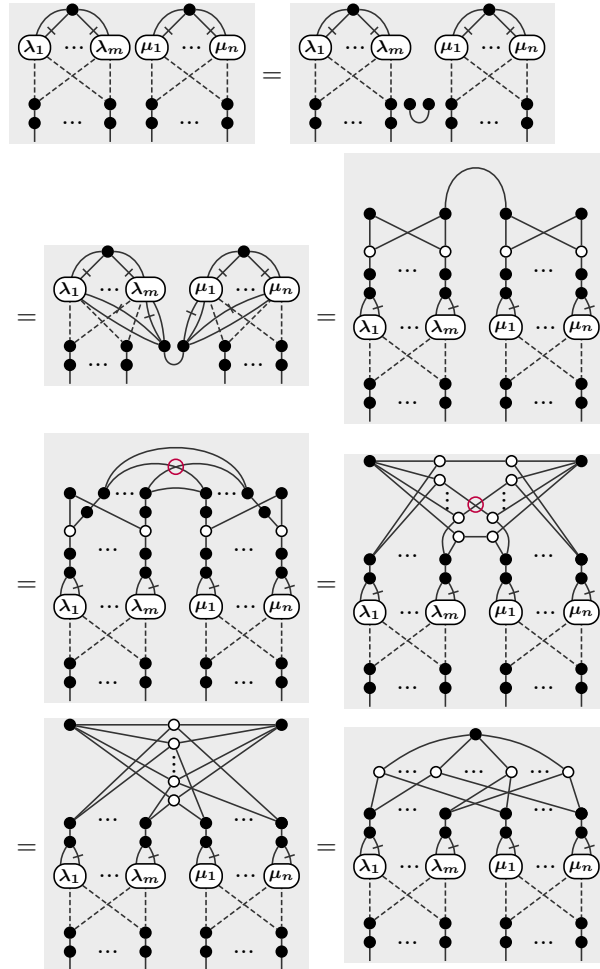


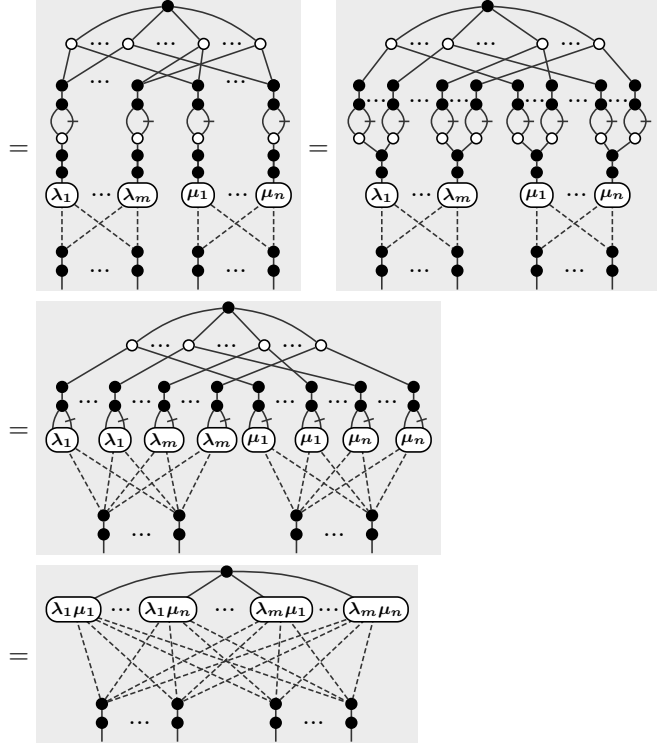
Proof of Lemma 15.



□

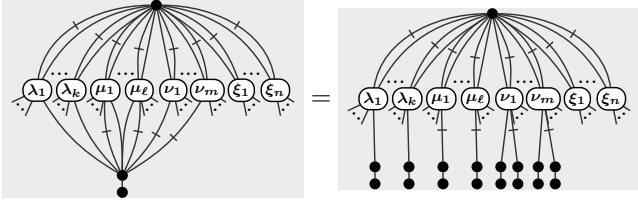
Proof of Proposition 8.





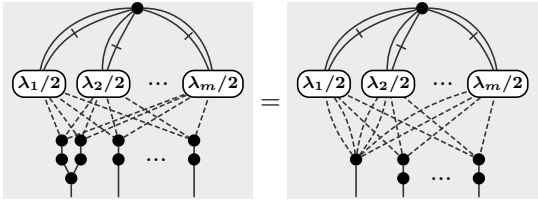
□

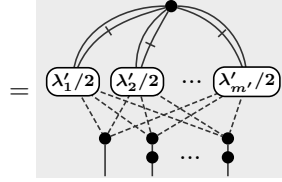
Proof of Lemma 3. Notice first that by the Negation Lemma, if the result is true for \downarrow then it is true for \uparrow . If we look at the case \downarrow , we get:

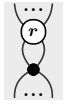
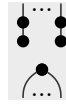
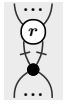
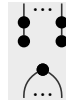



It then remains to apply the copy rule and the W-spider rule as many times as possible. This will remove the white nodes that were initially connected to the output (through \downarrow , \uparrow , or both) and all the \downarrow produced will be absorbed thanks to the spider rule to either the top W-spider or the ones at the outputs. □

Proof of Lemma 4. We have:

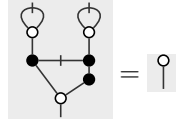




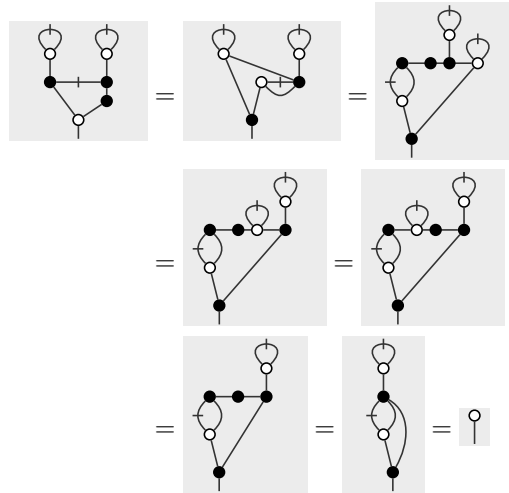
where in the last equality we use equations  =  and  =  to remove parallel edges of the

same type. In doing so, the white node is deleted and again it suffices to merge all occurrences of  and apply Lemma 2 to arrive to the normal form. □

Lemma 16.



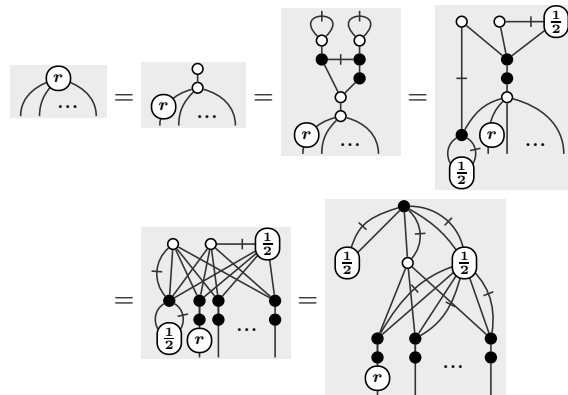
Proof of Lemma 16.

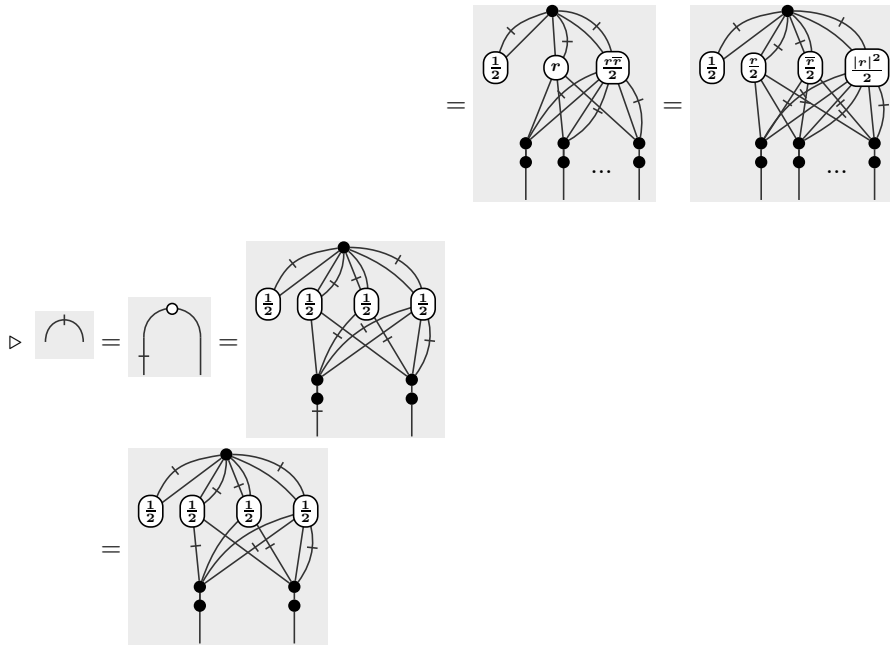


□

Proof of Proposition 10.

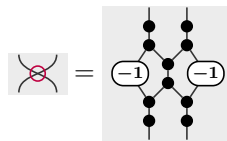
▷ The normal form of the Z-spider can be obtained as:





▷ For the W-spider, let us first look at the first 4 possible degrees: a W-spider of degree 0 is already in normal form. For the degree 1: $\bullet = \cap \bullet$, since the cap can be put in normal form, and thanks to Lemma 3, this generator can be put in normal form. For the degree 2: $\cap \cap = \cap \bullet$, and this time it is Lemma 2 that puts the diagram in normal form. Finally for degree 3: $\cap \cap \cap = \cap \cap \bullet$, that is, the $0 \rightarrow 3$ W-spider can be obtained by applying the $2 \rightarrow 1$ W-spider on the tensor product of a cap with itself. Since the cap can be put in normal form, thanks to Proposition 8 its tensor product with itself can be put in normal form, and by Lemma 4 the application of $\cap \bullet$ yields a diagram that can be put in normal form. W-spiders with a larger degree can then be obtained by compositions of the smaller ones, as: $\cap \dots = \cap \cap \dots \cap$.

▷ By completeness of the **ZW**-Calculus:



i.e. it is a composition of generators we already showed how to put in normal form. It can hence be put in normal form.

□