

Minimality in Finite-Dimensional ZW-Calculi

Marc de Visme, Renaud Vilmart

Université Paris-Saclay, Inria, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, 91190,
Gif-sur-Yvette, France.

marc.de-visme@inria.fr, renaud.vilmart@inria.fr

Abstract. The ZW-calculus is a graphical language capable of representing 2-dimensional quantum systems (qubit) through its diagrams, and manipulating them through its equational theory. We extend the formalism to accommodate finite dimensional Hilbert spaces beyond qubit systems.

First we define a qudit version of the language, where all systems have the same arbitrary finite dimension d , and show that the provided equational theory is both complete – i.e. semantical equivalence is entirely captured by the equations – and minimal – i.e. none of the equations are consequences of the others. We then extend the graphical language further to accommodate all finite dimensional Hilbert spaces at the same time. We again show the completeness of the provided equational theory.

1 Introduction

Graphical languages for quantum computations are a product of the categorical quantum mechanics program [1, 13] devoted to studying the foundations of quantum mechanics through the prism of category theory. These graphical languages come in different flavours, depending on which generators are used to build the diagrams (graphical representations of the quantum operators), and critically, displaying different kinds of interactions between said generators. The ZX-calculus describes the interaction between two complementary bases [11], the ZW-calculus, the interaction between the two “spiders” derived from the “GHZ” and “W” states, the only two fully entangled tripartite states up to SLOCC-equivalence [12], and the ZH-calculus the interaction between the same GHZ-state inferred spider and a spider obtained by generalising the Hadamard gate [3].

The equations that describe these interactions form “equational theories”, that define syntactic equivalence classes of diagrams, that are also semantically equivalent. When the syntactic equivalence matches perfectly the semantical one (i.e. when two diagrams represent the same quantum operator iff they can be turned into one another), we say that the equational theory is *complete*. Complete equational theories have been found for the aforementioned graphical languages, betimes for restrictions of them [2, 4, 9, 10, 18, 20, 23, 24, 33].

As is customary for the computer science part of the community, the focus was largely set onto *qubit* systems, i.e. systems where the base quantum system is 2-dimensional, yet this is enough to get applications in optimisation [5, 25], quantum error correction [6, 22, 32], verification [16, 21], simulation [26–28]... However, physics allow for qudit systems (where the base quantum system is d -dimensional with $d > 2$) and even infinite dimensional systems. Several attempts have hence been made to go beyond the qubit case [7, 19, 34], but it was only recently that a complete equational theory was found for d -dimensional (i.e. qudit) systems [30] and later for finite dimensional systems (so-called “qufinite”, i.e. for the category **FdHilb**) [36]. The results were obtained by generalising both the ZX and the ZW calculi and mixing them together. The W-node in particular allows for a neat intuitive (and unique) normal form for the diagrams. Satisfying the necessary conditions for every diagram to be normalisable then yields a complete equational theory. However, we believe that the ones obtained in [30, 36] are far from being *minimal*, due in particular to the presence the generators from both the ZX and the ZW calculi.

From a foundational perspective, it can be enlightening to know if an equation is a defining property of quantum systems (and hence necessary), or on the contrary if it is derivable from more fundamental properties (see e.g. [14, 15]). The redundancy in the equational theory may also cause issues when trying to transport the completeness result to other diagrammatic languages for qudit systems (every equation has to be proven in the new language, hence the fewer the better); or when trying to generalise further, e.g. to the **FdHilb** setting.

We argue here that the ZW-calculus is enough to get a natural normal form (akin to that of [30]) even in the qudit version, and provide two elegant equational theories, differing primarily in how we deal with the parameters in one of the generators. We show that the equational theories are *complete*, resorting to the normal form instead of transporting the completeness result from [30], for the reason described above. We also show that the first equational theory is minimal, meaning that none of the equations can be derived from the others, hence avoiding the aforementioned redundancy in the presentation.

We then adapt diagrams and the equational theory of the graphical language to accommodate all finite dimensional Hilbert spaces (**FdHilb**), in a way that requires no additional generator and only two new equations. Here again we prove the completeness of the equational theory, by leveraging that of the qudit setting.

The paper is split into two parts, Section 2 and Section 3, devoted respectively to the **Qudit**_{*d*} version, and to the **FdHilb** version. In the **Qudit**_{*d*} version, diagrams and their interpretation are introduced in Section 2.1 and the equational theory is introduced and discussed in Section 2.2. We then show its minimality in Section 2.3 and its completeness in Section 2.4. In the **FdHilb** version, diagrams and their interpretation are introduced in Section 3.1, and the equational theory is introduced and shown to be complete in Section 3.2. All missing proofs are provided in the appendix.

The Dirac Notation

All the upcoming diagrams can be given an interpretation as a linear map, in the appropriate category. In quantum information, it is usual to express such linear maps using the so-called Dirac notation. The current section hence serves as a gentle introduction to this notation.

Let $d \geq 2$. In the d -dimensional Hilbert space \mathbb{C}^d , the canonical basis

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \right\}$$

is usually denoted $\{|0\rangle, |1\rangle, \dots, |d-1\rangle\}$. All 1-qudit systems have states that live in \mathbb{C}^d and that can hence be represented by linear combinations of the elements of this basis: $|\psi\rangle = a_0|0\rangle + a_1|1\rangle + \dots + a_{d-1}|d-1\rangle$ (notice that the “ket” notation $|\cdot\rangle$ is used for states in general, not only basis elements).

To combine systems, we use the tensor product (Kronecker product): $(\cdot \otimes \cdot)$ which is a fairly standard operation on linear maps. In particular, the overall state obtained by composing two 1-qudit systems in respective states $|\psi\rangle$ and $|\varphi\rangle$ is simply $|\psi\rangle \otimes |\varphi\rangle$. Notice that $\{|i\rangle \otimes |j\rangle\}_{0 \leq i < d, 0 \leq j < d'}$ forms a basis of $\mathbb{C}^d \otimes \mathbb{C}^{d'} \simeq \mathbb{C}^{d \times d'}$. It is customary to write $|\psi, \varphi\rangle$ to abbreviate $|\psi\rangle \otimes |\varphi\rangle$.

The “bra” notation $\langle \cdot |$ is used to represent the dagger (the conjugate transpose) of a state, i.e. $\langle \psi | = |\psi\rangle^\dagger = \overline{|\psi\rangle}^\top$. The choice of the “bra-ket” notation is such that composing a bra with ket forms the bracket, the usual inner product in \mathbb{C}^d :

$$\langle \psi | \circ |\varphi\rangle = \langle \psi | \varphi \rangle$$

Linear combinations of kets and bras of the canonical basis can be used to represent any linear map of the correct dimensions, e.g. the 1-qudit identity:

$$id = \sum_{k=0}^{d-1} |k\rangle \langle k|$$

2 ZW-Calculus for Qudit Systems


In this section, we introduce a graphical language for quantum systems that all have the same fixed dimension d ($d \geq 2$): qudit systems.


2.1 Diagrams of ZW_d and their Interpretation


First, we need to introduce the mathematical objects at the heart of the graphical language – the diagrams – what they represent.

The Diagrams As is traditional for graphical languages for finite-dimensional quantum systems, we work with a \dagger -compact prop [29, 31, 37]. Categorically speaking, this is a symmetric, compact closed monoidal category generated by a single object, endowed with a contravariant endofunctor that behaves well with the symmetry and the compact structure. The following explains some of these concepts in more detail.

Let us denote ZW_d the \dagger -compact prop generated by:




▷ the Z-spider  : $n \rightarrow m$, for $r \in \mathbb{C}$


▷ the W-node  : $1 \rightarrow n$

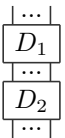
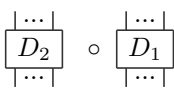
▷ the kets  : $0 \rightarrow 1$, for $1 \leq k < d$


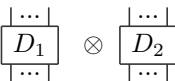
▷ the global scalars $r : 0 \rightarrow 0$, for $r \in \mathbb{C}$

The symmetry and the compact structure of the prop are represented respectively by:

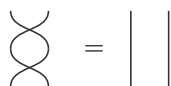
▷  : $2 \rightarrow 2$ and
 ▷  : $0 \rightarrow 2$,  : $2 \rightarrow 0$,

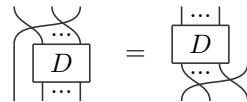
and the identity as  : $1 \rightarrow 1$. All these generators can be composed:

▷ sequentially:  := 

▷ in parallel:  := 

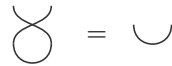
The symmetry satisfies the following identities:

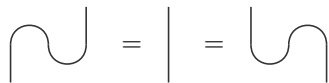





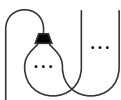
and the compact structure:









This compact structure in particular allows us to define the "upside-down" version of the generators, for instance:

 := 

 := 

The \dagger functor is defined inductively as:

$$\begin{array}{l}
\triangleright (D_2 \circ D_1)^\dagger = D_1^\dagger \circ D_2^\dagger \\
\triangleright (D_1 \otimes D_2)^\dagger = D_1^\dagger \otimes D_2^\dagger \\
\triangleright \left(\begin{array}{c} | \\ | \end{array} \right)^\dagger = \begin{array}{c} | \\ | \end{array} \\
\triangleright \left(\begin{array}{c} \diagup \\ \diagdown \end{array} \right)^\dagger = \begin{array}{c} \diagdown \\ \diagup \end{array} \\
\triangleright \left(\begin{array}{c} \cup \\ \cup \end{array} \right)^\dagger = \begin{array}{c} \cup \\ \cup \end{array} \\
\triangleright \left(\begin{array}{c} \cap \\ \cap \end{array} \right)^\dagger = \begin{array}{c} \cap \\ \cap \end{array}
\end{array}
\qquad
\begin{array}{l}
\triangleright \left(\begin{array}{c} \dots \\ n \\ \dots \\ m \end{array} \right)^\dagger = \begin{array}{c} \dots \\ m \\ \dots \\ n \end{array} \\
\triangleright \left(\begin{array}{c} \bullet \\ \dots \\ n \end{array} \right)^\dagger = \begin{array}{c} \dots \\ \bullet \end{array} \\
\triangleright \left(\begin{array}{c} \bullet \\ | \end{array} \right)^\dagger = \begin{array}{c} | \\ \bullet \end{array} \\
\triangleright (r)^\dagger = \bar{r}
\end{array}$$

Notice that thanks to the identities satisfied by the \dagger -compact prop, the \dagger -functor is involutive.

As will be made clearer in what follows, in \mathbf{ZW}_d , $d \geq 2$ represents the dimension of the "base" quantum system, called qudit. As this d will be fixed in the following, we may forget to specify it. For convenience, we define an empty white node as a parameter 1 Z-spider:

$$\begin{array}{c} \dots \\ n \\ \dots \\ m \end{array} := \begin{array}{c} \textcircled{1} \\ \dots \\ n \\ \dots \\ m \end{array}$$

and give the $1 \rightarrow 0$ W-node a special symbol, akin to the dagger of the kets (as its interpretation, as we will see later, is merely $\langle 0|$):

$$\begin{array}{c} | \\ \bullet \end{array} := \begin{array}{c} | \\ \blacktriangle \end{array}$$

The Interpretation The point of the diagrams of the \mathbf{ZW}_d is to represent quantum operators on multipartite d -dimensional systems. The way those are usually specified is thanks to the category \mathbf{Qudit}_d . This forms again a symmetric \dagger -compact prop, where the base object is $1 := \mathbb{C}^d$, and morphisms $n \rightarrow m$ are linear maps $\mathbb{C}^{d^n} \rightarrow \mathbb{C}^{d^m}$. The symmetry and the compact structure correspond to their counterparts in \mathbf{ZW}_d , they will be stated out in the following, as part of the interpretation of \mathbf{ZW}_d diagrams. The \dagger functor is the usual \dagger of linear maps in \mathbb{C} .

We may hence interpret diagrams of the \mathbf{ZW}_d -calculus thanks to the functor $\llbracket \cdot \rrbracket : \mathbf{ZW}_d \rightarrow \mathbf{Qudit}_d$ inductively defined as follows:

$$\begin{array}{l}
\llbracket D_2 \circ D_1 \rrbracket = \llbracket D_2 \rrbracket \circ \llbracket D_1 \rrbracket \\
\llbracket D_1 \otimes D_2 \rrbracket = \llbracket D_1 \rrbracket \otimes \llbracket D_2 \rrbracket \\
\llbracket \begin{array}{c} | \\ | \end{array} \rrbracket = \sum_k |k\rangle\langle k| \\
\llbracket \begin{array}{c} \diagup \\ \diagdown \end{array} \rrbracket = \sum_{k,\ell} |\ell, k\rangle\langle k, \ell| \\
\llbracket \begin{array}{c} \cup \\ \cup \end{array} \rrbracket = \llbracket \begin{array}{c} \cup \\ \cup \end{array} \rrbracket^\dagger = \sum_k |k, k\rangle \\
\llbracket \begin{array}{c} \cap \\ \cap \end{array} \rrbracket = \llbracket \begin{array}{c} \cap \\ \cap \end{array} \rrbracket^\dagger = \sum_k |k, k\rangle
\end{array}
\qquad
\begin{array}{l}
\llbracket \begin{array}{c} \dots \\ n \\ \dots \\ m \end{array} \rrbracket = \sum_{k=0}^{d-1} r^k \sqrt{k!}^{n+m-2} |k^m\rangle\langle k^n| \\
\llbracket \begin{array}{c} \bullet \\ \dots \\ n \end{array} \rrbracket = \sum_{\substack{k \in \{0, \dots, d-1\} \\ i_1 + \dots + i_n = k}} \sqrt{\binom{k}{i_1, \dots, i_n}} |i_1, \dots, i_n\rangle\langle k| \\
\llbracket \begin{array}{c} \bullet \\ | \end{array} \rrbracket = \sqrt{k!} |k\rangle \\
\llbracket r \rrbracket = r
\end{array}$$

where $\binom{k}{i_1, \dots, i_n} = \frac{k!}{i_1! \dots i_n!}$ is a multinomial coefficient. Notice that the interpretation of the $1 \rightarrow 0$ W-node is simply:

$$\llbracket \begin{array}{c} | \\ \bullet \end{array} \rrbracket = \sum_{\substack{k \in \{0, \dots, d-1\} \\ 0=k}} \sqrt{\binom{k}{0}} \langle k| = \langle 0|$$

The presence of $\sqrt{}$ on the coefficients is not particularly relevant, and is simply an artefact of us maintaining some symmetry between generators and their dagger. Indeed, as explained below, we have $\llbracket \begin{array}{c} \bullet \\ \bullet \end{array} \rrbracket = k!$, meaning that the coefficient $k!$ needs to be split between both nodes, resulting in either an asymmetric presentation or a square root.

Notice also that the interpretation of the Z-spider differs from more usual generalisations of its qubit counterpart, because of the $\sqrt{k^{n+m-2}}$ which depends on the degree of the spider. While it makes the interpretation of the diagrams slightly more complicated, it allows us – as will be stated later – to quite conveniently generalise equations from the qubit ZW-calculus, and hence have a simpler equational theory. It will be shown in the following (Corollary 1), that the above set of generators makes for a universal calculus, i.e. any linear map of \mathbf{Qudit}_d can be represented by a \mathbf{ZW}_d -diagram.

To gain intuition about the upcoming equations between diagrams, it can be useful to *semantically* decompose a diagram into sums of simpler ones¹. To do so, it can be convenient to understand $|k\rangle$ as a bunch of k indistinguishable particles.

The W-node spreads the k “particles” that enter it by a multinomial distribution:

$$\left[\begin{array}{c} k \\ \text{W} \\ \dots \end{array} \right] = \sum_{i_1 + \dots + i_n = k} \binom{k}{i_1, \dots, i_n} \left[\begin{array}{c} i_1 \dots i_n \\ \dots \end{array} \right]$$

The $2 \rightarrow 1$ W-node takes two bunches of particles k and ℓ and regroups them into one, and yields the null state if $k + \ell$ exceeds the “capacity” (i.e. the dimension) of a single wire:

$$\left[\begin{array}{c} k \quad \ell \\ \text{W} \\ \dots \end{array} \right] = \begin{cases} \left[\begin{array}{c} k + \ell \\ \dots \end{array} \right] & \text{if } k + \ell < d \\ \bar{0} & \text{if } k + \ell \geq d \end{cases}$$

This will be proven graphically (Lemma 7) from the upcoming equational theory (Figure 1). When $k + \ell < d$, the fact that there is no additional scalar is due to the rescaling of the k -dots to represent $\sqrt{k!} |k\rangle$. This rescaling also makes the “copy” more natural: The Z-spider $1 \rightarrow n$ copies any bunch of k particles entering it, yielding global scalar r^k in the process:

$$\left[\begin{array}{c} k \\ \text{Z} \\ \dots \\ n \end{array} \right] = r^k \left[\begin{array}{c} k \quad k \\ \dots \\ n \end{array} \right]$$

This will again be proven graphically (Lemma 8) from the equational theory.

The rescaling, however, forces:

$$\left[\begin{array}{c} k \\ \text{b} \\ \ell \end{array} \right] = k! \quad \text{and} \quad \left[\begin{array}{c} k \\ \text{b} \\ \ell \end{array} \right] = 0 \quad \text{if } k \neq \ell$$

Finally, it can be useful to decompose the identity as a linear combination of products of kets and bras:

$$\left[\begin{array}{c} | \\ \dots \\ \langle \end{array} \right] = \sum_{k=0}^{d-1} \frac{1}{k!} \left[\begin{array}{c} k \\ \dots \\ k \end{array} \right]$$

2.2 Equational Theory

With the above interpretation of the \mathbf{ZW}_d , different diagrams may yield the same linear map. All axioms of symmetric \dagger -compact props in particular preserve the interpretation. More generally, we may want to relate together *all* diagrams that have the same semantics. This is done through an equational theory, i.e. a set of equations that can be applied locally in a diagram without changing the semantics of the whole.

¹ Notice that here, such decompositions are merely semantical. The upcoming completeness is only interested in equivalence between single diagrams.

Equations of the ZW_d -Calculus On top of the axioms of symmetric \dagger -compact props, we assume some conventional equations about the topology of the generators, which should align with the symmetries of the symbols used to depict them. The Z-spider does not distinguish between any of its connections: it is “flexsymmetric” [8], meaning that we can interchange any of its legs without changing the semantics. Graphically, for any permutation of wires σ :

On the other hand, the binary W -node is only co-commutative, which, together with the upcoming Equation (a), means that all the outputs of the n -ary W -node can be exchanged, i.e. for any permutation of wire σ :

With all this in place, we can give the core of the equational theory, in Figure 1. When diagram D_1 can be turned into diagram D_2 using the rules of ZW_d , we write $ZW_d \vdash D_1 = D_2$.

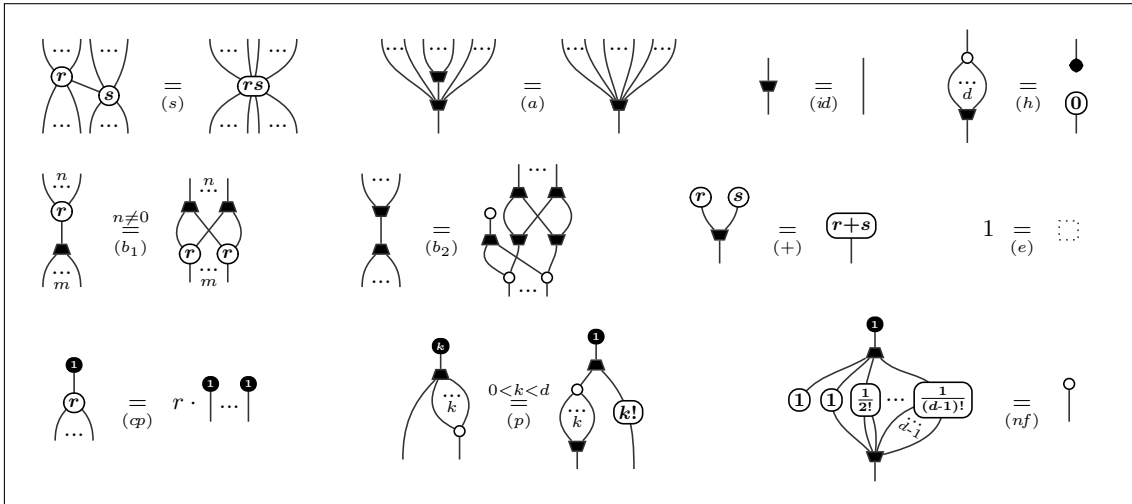


Fig. 1: Equational theory ZW_d for the qudit ZW -calculus.

Remark 1. In this framework, we can tensor global scalars together $r \otimes s$, which graphically could be confused with their product rs . This is actually unambiguous in the equational theory, as:

$$ZW_d \vdash r \otimes s \stackrel{(cp)}{=} \begin{array}{c} \bullet \\ | \\ \textcircled{r} \end{array} \begin{array}{c} \bullet \\ | \\ \textcircled{s} \end{array} \stackrel{(cp)}{=} \begin{array}{c} \bullet \\ | \\ \textcircled{rs} \end{array} \stackrel{(s)}{=} \begin{array}{c} \bullet \\ | \\ \textcircled{rs} \end{array} \stackrel{(cp)}{=} rs$$

Moreover, Equation (e) states that global scalar 1 is the empty diagram. Scalar multiplication is assumed to be automatically applied, and scalar 1 is assumed to be automatically removed in the following.

All rules up to (cp) are fairly standard generalisations of rules of the qubit ZW -calculus (with (b₂) being inspired from [30] to avoid using a *fermionic swap*, as explained below). The non-conventional \sqrt{k}^{n+m-2} coefficients in the interpretation of the Z-spider seem to be necessary for

Equations (s), (b₁) and (+) to all work. Equation (p) relates $|k\rangle$ for $k > 1$ to $|1\rangle$. On the left hand side of the equation, the Z-spider forces the k rightmost outputs from the W-node to carry the same state, hence applying $|k\rangle$ on top of the W-node results in only two terms: one where the k particles all go to the left, and one where they are spread on the k inputs of the Z-spider, resulting in a mere $|1\rangle$ on the right wire. This can be achieved by applying a $|1\rangle$ on the input of a W-node, which results in $|0, 1\rangle + |1, 0\rangle$ and by “multiplying” the number of particles on the left by k . The Z-spider parameter $k!$ is here to deal with the unnormalisation of one of the two terms:

$$\left[\begin{array}{c} \bullet \\ \vdots \\ k \\ \circ \end{array} \right] = \left[\begin{array}{c} \bullet \\ \circ \end{array} \right] + k! \left[\begin{array}{c} \bullet \\ \circ \end{array} \right] = \left[\begin{array}{c} \bullet \\ \vdots \\ k \\ \circ \end{array} \right]$$

Finally, Equation (nf) simply states what the normal form (to be defined later) of the unary Z-spider should be.

Proposition 1. *All equations in ZW_d are sound, i.e.:*

$$ZW_d \vdash D_1 = D_2 \implies \llbracket D_1 \rrbracket = \llbracket D_2 \rrbracket$$

Proof. This is a straightforward verification for most of the equations. They can all be proven using the aforementioned identities in the semantics of the diagrams, especially the decomposition of the identity. Equation (b₂) requires the Vandermonde identity:

$$\sum_{k_1 + \dots + k_p = m} \binom{n_1}{k_1} \dots \binom{n_p}{k_p} = \binom{n_1 + \dots + n_p}{m}.$$

Related Work The first and only result to date of a complete equational theory for a graphical language describing qudit systems comes from the “ZXW-calculus” [30]. There, the authors start from a qudit version of the ZX-Calculus and most probably end up requiring a W-node in the definition of a normal form, and hence in the equational theory leading to completeness. We argue here that we can get a complete equational theory, purely inside the ZW-calculus. Doing so, we end up with fewer generators, and as a consequence, fewer, more intuitive equations in the equational theory.

Focussing on ZW-calculus is not a new idea. The first ever completeness proof for qubit graphical languages was in the (qubit) ZW-calculus, introduced in [12] and tweaked and made complete in [18, 20]. The ZW-calculus noticeably has very nice combinatorial properties different from those of its counterparts, which in particular allows for a very natural notion of normal form. It is hence not surprising that some attempts were made to get a complete equational theory of qudit systems purely in ZW. There have then been tentative generalisations for qudit systems, in particular in [19] where q -arithmetic is used, and in [35] where the W-node is generalised in a different way (and that we encounter in [30]).

The system we are actually the closest with, is **QPath** [17]. Our W-node is merely the “triangle” node of **QPath** that we truncated to a finite dimension, and we generalised their “line weight” to an n -ary Z-spider. The degree-2 Z-spider furthermore has exactly the same interpretation as the line weight. While in **QPath** the triangle nodes satisfy a bialgebra, this is not the case when truncating to finite dimension. Here we could either resort to define a “fermionic swap” that would replace the usual swap in the bialgebra (as in [19] and [35]), or give a context in which the bialgebra works (as is done in [30]). While such a “fermionic swap” exists in our setting, it does not have all the nice properties of the qubit fermionic swap, that in particular allow us to see it as a quasi-proper swap. Instead we went with the latter solution, which as it turns out works in our setting, despite the W-node having a different interpretation from that of [30], and we end up with Equation (b₂).



A final important distinction with [35] and [30] is that we decided that the Z-spider would have a single parameter, instead of a $(d - 1)$ -sized list of them that would describe all but the 0-th weights in the description of the Z-spider as a linear combination of the basis states. This makes the overall presentation leaner, with the downside of requiring Equation (nf) that fully describes the



normal form of the unary Z-spider. In Appendix B, we give a presentation of the qudit ZW-calculus with lists as parameters. This simplifies Equation (nf) at the cost of more complex bookkeeping and operations on Z-spider parameters.

2.3 Minimality

Minimality of an equational theory states that every single equation is necessary: none can be derived from the others. Said otherwise, as soon as we remove one of the equations, some equalities (that were previously provable) become unprovable. Minimality is a fundamental property, as it allows us to pinpoint properties that are necessary to our model, and a contrario those that are consequences of the necessary ones. Notice however that there is usually not a single minimal equational theory, as it often happens that one equation can be replaced by an equivalent one.

In trying to prove minimality, it often happens that two equations fail to be proven necessary individually, but that the pair (i.e. at least one of the two) can be proven necessary. Such cases underline some sort of proximity between the two equations, and the obstacle it poses to minimality can sometimes be circumvented (somewhat artificially) by merging them into a single, potentially

slightly less intuitive, equation. This happened once here: we merged equations  =  and

 = , that we initially had as axioms, into Equation (h). This finally provides us with a minimal equational theory for qudit ZW-calculus.

To prove that an equation is necessary, we define a non-standard interpretation which is preserved by all the equations (including the axioms of †-compact props), except the equation of interest. When such an interpretation is exhibited, we can safely conclude that the equation is necessary, as if it were a consequence of the others, it would also preserve this interpretation. Interpretations like these sometimes simply take the form of a quantity that turns out to be invariant for all equations except the one that is considered.

In the following, we show that the equational theory ZW_d from Figure 1 is minimal, i.e. that none of the equations can be derived from the others. It is to be noted that most of the equations in Figure 1 are schemas, that is they are parametrised, and the equation is assumed for all possible values of the parameters. Our minimality result is “weak” in the sense that for each equation schema, we show that *at least* one of the occurrences cannot be derived from the other equations, but we do not pinpoint for which parameters the equation is necessary or not. Nevertheless:

Theorem 1. *All equations in ZW_d are necessary, hence ZW_d is minimal.*

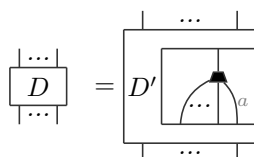
Several arguments in the proof require to distinguish between Z-spiders that have a non-trivial influence on the semantics of the diagrams, from those that essentially have no effect on the semantics. This is captured by the following definition:

Definition 1 (Effective Z-spider). *Let z be a Z-spider in diagram D . For $x \in \mathbb{C}$, we denote by $D[z * = x]$ the diagram obtained by multiplying the parameter of z by x in D . We say that z is effective if:*

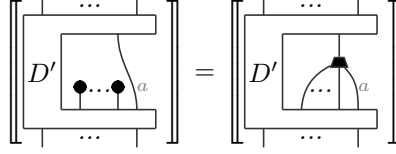
$$\forall x \in \mathbb{C} \setminus \{1\}, \forall \lambda \in \mathbb{C}, \llbracket D[z * = x] \rrbracket \neq \lambda \llbracket D \rrbracket$$

There is a similar notion that we need for determining whether a W-node could essentially just be an identity from its input to one of its outputs, the other outputs merely bearing a $|0\rangle$:

Definition 2 (Trivial W-node and sole effective output). *Let D be a diagram with a W-node. Let's call a one of its outputs:*



We say that edge a is the sole effective output of the W -node if:



If a W -node has a sole effective output, we say that it is trivial.

We can now prove Theorem 1:

Proof. We consider each of these equations individually:

- (s) When applying the transformation that turns all Z -spider parameters and global scalars to their real part ($r \mapsto \text{Re}(r)$), Equation (s) is the only one that is not preserved.
- (a) It is the only equation permitting to create non-trivial W -nodes with arity $> \frac{d^2}{2}$ from a diagram that only has W -nodes with arity $\leq \frac{d^2}{2}$.
- (id) It is the only equation that can create nodes connected to boundaries from a node-free diagram.
- (cp) It is the only equation that creates a non-1 scalar
- (e) It is the only equation that can create generators out of empty diagrams.
- (p) It is the only equation able to introduce generator $\begin{array}{c} \bullet \\ | \end{array}$ for $k > 1$.
- (nf) Take the interpretation that maps $\begin{array}{c} \bullet \\ | \end{array}$ to $\begin{array}{c} \bullet \\ | \end{array}$. All rules hold (up to colinearity) except (nf).
- (+) Take the interpretation that maps all the Z -spider parameters (and the global scalars) to their absolute value ($r \mapsto |r|$). This interpretation preserves all equations except (+).
- (b₁) Consider diagrams as graphs, and define an “effective Z -path” in the diagram as a path 1) that goes from a boundary to another boundary, and 2) that only goes through effective Z -spiders or through trivial W -nodes, through their sole effective output. All equations except (b₁) preserve the existence of effective Z -paths.
- (b₂) Consider diagrams as graphs, and define a “ W -path” in the diagram as a path 1) that goes from a boundary to another boundary, 2) which cannot use two outputs of a W -node (if it goes through a W -node, it has to use the input edge) and 3) that does not go through a Z -spider. All equations, except (b₂), preserve the existence of a W -path. (b₂) is the only equation that can bring the number of W -paths from non-zero to zero (which is done by adding a Z -spider on the path).
- (h) To each wire in a diagram D , we associate a number $0 \leq k < d$ (or more graphically we annotate each wire by some number k). The procedure to do so is as follows:

1. annotate all wires with $d - 1$

2. rewrite the annotations using the following rules, until a fixed point is reached:

$$\begin{array}{l}
 \bullet \quad a \begin{array}{c} \bullet \\ | \end{array} \xrightarrow{a \neq 0} 0 \begin{array}{c} \bullet \\ | \end{array} \quad \text{and} \quad a \begin{array}{c} \bullet \\ | \end{array} \xrightarrow{k \leq a} k \begin{array}{c} \bullet \\ | \end{array} \\
 \\
 \bullet \quad \begin{array}{c} a_0 \dots a_k \\ \text{---} \\ \text{---} \\ \text{---} \\ a_{k+1} \dots a_q \end{array} \rightarrow \begin{array}{c} a \dots a \\ \text{---} \\ \text{---} \\ \text{---} \\ a \dots a \end{array} \quad \text{with } a := \min(a_i) \\
 \\
 \bullet \quad \begin{array}{c} a \\ \text{---} \\ \text{---} \\ \text{---} \\ b_1 \dots b_n \end{array} \rightarrow \begin{array}{c} \min(a, \sum b_i) \\ \text{---} \\ \text{---} \\ \text{---} \\ \min(a, b_1) \dots \min(a, b_n) \end{array}
 \end{array}$$

This simple procedure obviously terminates, as a step is only applied if at least one of the annotations is decreased. By considering inputs and outputs of D (which are the only wires that can be guaranteed to remain during rewrites with ZW_d), we can check that Equation (h) is the only one that can modify the outcome of the procedure.

Some Useful Derivable Equalities The equational theory being minimal, there are several useful and intuitive identities that were left out, as they can be proven to be derivable. The following lemmas will in particular be useful in the upcoming proof of completeness:

Lemma 1.

$$\text{ZW}_d \vdash \begin{array}{c} | \\ \circ \end{array} = \begin{array}{c} | \\ | \end{array}$$

Lemma 2.

$$\text{ZW}_d \vdash \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{c} \circ \\ \circ \end{array} \\ \dots \end{array} = \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{c} \circ \\ \circ \end{array} \\ \dots \end{array}$$

Lemma 3.

$$\text{ZW}_d \vdash \begin{array}{c} \dots \\ \downarrow \\ \circ \\ \bullet \end{array} = \begin{array}{c} \dots \\ \bullet \end{array}$$

Lemma 4.

$$\text{ZW}_d \vdash \begin{array}{c} \textcircled{0} \\ | \\ \bullet \end{array} = \begin{array}{c} \bullet \end{array}$$

Lemma 5.

$$\text{ZW}_d \vdash \begin{array}{c} \textcircled{r} \quad \textcircled{s} \\ \downarrow \quad \downarrow \\ \dots \end{array} = \begin{array}{c} \textcircled{r+s} \\ \dots \end{array}$$

Lemma 6.

$$\text{ZW}_d \vdash \begin{array}{c} | \\ \circ \\ \downarrow \\ \bullet \end{array} = \begin{array}{c} \circ \\ \bullet \end{array}$$

Lemma 7.

$$\text{ZW}_d \vdash \begin{array}{c} \textcircled{k} \quad \textcircled{\ell} \\ \downarrow \\ \bullet \end{array} = \begin{cases} \textcircled{k+\ell} & \text{if } k + \ell < d \\ \textcircled{0} \bullet & \text{if } k + \ell \geq d \end{cases}$$

Lemma 8. If $n > 0$:

$$\text{ZW}_d \vdash \begin{array}{c} \textcircled{k} \\ \downarrow \\ \textcircled{r} \\ \dots \\ \textcircled{n} \end{array} = r^k \cdot \begin{array}{c} \textcircled{k} \\ \dots \\ \textcircled{n} \end{array}$$

Lemma 9.

$$\text{ZW}_d \vdash \begin{array}{c} \textcircled{1} \\ \downarrow \\ \circ \end{array} = \begin{array}{c} \textcircled{1} \quad \textcircled{1} \\ \downarrow \quad \downarrow \\ \circ \end{array}$$

and

$$\text{ZW}_d \vdash \begin{array}{c} \textcircled{1} \\ \downarrow \\ \circ \end{array} = \begin{array}{c} \textcircled{1} \\ \downarrow \\ \bullet \end{array}$$

2.4 Completeness

Completeness of an equational theory with respect to a semantics is the fundamental property that ensures that semantical equivalence of diagrams is entirely captured by the equational theory. Minimality is worthless without some form of completeness, as it is extremely simple to design minimal, but not complete, equational theories. For instance, the empty equational theory (that contains no axiom), is minimal but clearly not complete for the qudit ZW-diagrams. We hence show in this section that we indeed have completeness.

Normal Form and Universality The usual way to prove completeness is to show that any diagram can be put in a normal form, and that this normal form is unique and similar for all equivalent diagrams. As is customary in a category that is compact-closed, we can focus on states, as there is an isomorphism between operators and states [1]:

$$\begin{array}{c} \dots \\ \boxed{D} \\ \dots \end{array} = \begin{array}{c} \dots \\ \boxed{D} \end{array} \begin{array}{c} \dots \\ \dots \end{array} =: \begin{array}{c} \dots \\ \boxed{\ulcorner D \urcorner} \\ \dots \end{array}$$

Although this section is one of the lengthiest parts of the paper, it does not show in the body as all proofs are offset to Appendix A.

Definition 3. We define $\mathcal{N} : \text{Qudit}_d \rightarrow \text{ZW}_d$ as the functor such that, for any n -qudit state $|\psi\rangle = r_0 |0\dots 0\rangle + r_1 |0\dots 01\rangle + \dots + r_i |x_1^i \dots x_n^i\rangle + \dots$:

$$\mathcal{N}(|\psi\rangle) = \begin{array}{c} \textcircled{1} \\ \downarrow \\ \textcircled{r_0} \quad \textcircled{r_1} \quad \dots \quad \textcircled{\frac{r_i}{\sqrt{\alpha_1^i! \dots \alpha_n^i!}}} \quad \dots \\ \downarrow \quad \downarrow \quad \dots \quad \downarrow \\ \dots \end{array}$$

We say of any diagram in the image of \mathcal{N} that it is in normal form.

This functor \mathcal{N} creates a diagram whose interpretation is the starting state:

Proposition 2. $\forall |\psi\rangle \in \mathbf{Qudit}_d[0, n], \llbracket \mathcal{N}(|\psi\rangle) \rrbracket = |\psi\rangle$.

As a simple consequence of this proposition, any qudit operator can be represented by a diagram of \mathbf{ZW} :

Corollary 1 (Universality).

$$\forall f \in \mathbf{Qudit}_d[n, m], \exists D_f \in \mathbf{ZW}_d[n, m], \llbracket D_f \rrbracket = f$$

As we defined the normal form as the image of a map from the semantics, any diagram can only be associated to a unique normal form.

Compositions of Normal Forms Our goal now is to show that any diagram can be put in normal form. To do so, we show that all generators can be put in normal form, and that all compositions of diagrams in normal form can be put in normal form.

We start by showing the latter for the tensor product:

Proposition 3. *The spatial composition of diagrams in normal form can be put in normal form, i.e. $\mathbf{ZW}_d \vdash \mathcal{N}(v_1) \otimes \mathcal{N}(v_2) = \mathcal{N}(v_1 \otimes v_2)$.*

When turning arbitrary operators into states, the sequential composition turns into the application of cups \cup onto pairs of outputs of the state, as:

Proposition 4. *The diagram obtained by applying a cup \cup to two outputs of a diagram in normal form can be put in normal form.*

Normal Forms of the Generators Then we move on to showing that all the generators can be put in normal form.

Proposition 5. *The Z-spider can be put in normal form.*

Proposition 6. *The diagram obtained by applying ∇ to two outputs of a normal form can be put in normal form.*

Corollary 2. *The W-node can be put in normal form.*

Proposition 7. *Generators $\overset{\bullet}{\uparrow}$ can be put in normal form.*

Theorem 2 (Completeness for Qudit Systems). *The language is complete: for any two diagrams D_1 and D_2 of the \mathbf{ZW}_d -calculus:*

$$\llbracket D_1 \rrbracket = \llbracket D_2 \rrbracket \iff \mathbf{ZW}_d \vdash D_1 = D_2$$

Proof. By Propositions 5 and 7, and Corollary 2, any generator of the language can be put in normal form. Thanks to Propositions 3 and 4, compositions of diagrams in normal form can be put in normal form. As a consequence, any diagram can be put in normal form. By uniqueness of this normal form, if two diagrams share the same semantics, they can be rewritten into the same diagram. This proves completeness.

3 Finite Dimensional Hilbert Spaces

In the previous setting, all systems are required to be d -dimensional for some fixed d . Here we relax that constraint, which allows us to represent morphisms of **FdHilb**

FdHilb [1] is the strict symmetric monoidal \dagger -compact category of finite dimensional Hilbert spaces. Its objects are tensor products of finite dimensional Hilbert spaces $\mathbb{C}^d (d \in \mathbb{N} \setminus \{0\})$, and its morphisms are linear maps between them. The symmetry and the compact structure are naturally extended from that of **Qudit** $_d$.

In this new setting, we will be able to represent *all* morphisms of **FdHilb**, at the cost of annotating the wires of the diagrams to keep track of their dimensions. Instead of the dimension itself, we rather annotate the wire with its dimension -1 , i.e. with the largest k such that $|k\rangle$ is allowed on the wire. We call such k the *capacity* of the wire. This makes the bookkeeping a little bit less tedious.

3.1 Diagrams and Interpretation

We also require the following constraints for the capacities around each generator:

- ▷ All capacities around a Z-spider are the same
- ▷ The input capacity of the W-node must be larger than each of its outputs

The first constraint follows from the fact that Z-spiders in ZW can be seen as a generalisation of graph edges (more precisely they can be seen as hyperedges). Hence the whole hyperedge should have a single capacity. The second constraint simply comes from the fact that a larger capacity on the outputs of a W-node will never be used, so we might as well prevent it. When considering $1 \rightarrow 1$ W-nodes, which represent projections, this restriction allows us to see at a glance which side has the largest dimension.

The first restriction further allows us to put the capacity annotation on the Z-spider rather than on all its legs, making annotating diagrams less cumbersome.

We now work with a \dagger -compact symmetric monoidal category, but which is not a prop anymore. Our base objects are \mathbb{C}^d for $d \in \mathbb{N} \setminus \{0\}$. Every pair of objects can be composed with \otimes to form a third object, with \otimes being associative, and with the tensor unit I being $I := \mathbb{C}^1$. We work with a strict monoidal category, so we consider $I \otimes \mathbb{C}^d = \mathbb{C}^d = \mathbb{C}^d \otimes I$. To simplify notations, we represent objects $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2} \otimes \dots \otimes \mathbb{C}^{d_n}$ by a list of capacities $\langle d_1 - 1, d_2 - 1, \dots, d_n - 1 \rangle$ (the tensor product simply becomes the concatenation of lists). The tensor unit is represented by $\langle \rangle$.

In this new setting, the generators are generalised as follows:

$$\triangleright \text{Z-spiders } \begin{array}{c} \langle \dots \rangle \\ \textcircled{r} \\ \langle \dots \rangle \end{array} : \langle \overbrace{a, \dots, a}^n \rangle \rightarrow \langle \overbrace{a, \dots, a}^m \rangle \text{ with } r \in \mathbb{C}$$

$$\triangleright \text{W-nodes } \begin{array}{c} \textcircled{a} \\ \textcircled{b_1} \dots \textcircled{b_n} \end{array} : \langle a \rangle \rightarrow \langle b_1, \dots, b_n \rangle$$

$$\text{with } a \geq \max_{1 \leq i \leq n} (b_i)$$

$$\triangleright \text{kets } \begin{array}{c} \textcircled{k} \\ \textcircled{a} \end{array} : \langle \rangle \rightarrow \langle a \rangle \text{ with } k \leq a$$

$$\triangleright \text{global scalars } r : \langle \rangle \rightarrow \langle \rangle \text{ with } r \in \mathbb{C}$$

with the symmetry and the compact structure being generalised to

$$\triangleright \begin{array}{c} \textcircled{a} \quad \textcircled{b} \\ \diagdown \quad \diagup \\ \textcircled{a} \quad \textcircled{b} \end{array} : \langle a, b \rangle \rightarrow \langle b, a \rangle \text{ and}$$

$$\triangleright \left(\begin{array}{c} \textcircled{a} \\ \textcircled{a} \end{array} : \langle \rangle \rightarrow \langle a, a \rangle, \begin{array}{c} \textcircled{a} \\ \textcircled{a} \end{array} : \langle a, a \rangle \rightarrow \langle \rangle \right),$$

and the identity as $\begin{array}{c} | \\ \text{---} \\ a \end{array} : \langle a \rangle \rightarrow \langle a \rangle$.

Diagrams can still be composed together both sequentially and in parallel. The composition of categories prevents us from composing diagrams with unmatched objects (e.g. two Z-spiders with different capacities in sequence), but the conditions on the capacities of the W-nodes still have to be checked for a diagram to be valid. Such valid diagrams are called **ZW_f**-diagrams, and are graphical representations of the morphisms of the †-compact symmetric monoidal category **FdHilb**.

The compact structure still allows us to define upside-down versions of the W-node and the kets. Again, we give the $1 \rightarrow 0$ W-node a special symbol:

$$\begin{array}{c} | \\ \text{---} \\ \bullet \end{array} := \begin{array}{c} | \\ \text{---} \\ \blacktriangle \end{array}$$

The interpretation of these diagrams is now a monoidal functor $\llbracket \cdot \rrbracket : \mathbf{ZW}_f \rightarrow \mathbf{FdHilb}$ inductively defined as:

$$\begin{aligned} \llbracket D_2 \circ D_1 \rrbracket &= \llbracket D_2 \rrbracket \circ \llbracket D_1 \rrbracket & \llbracket \begin{array}{c} a \\ \text{---} \\ \text{---} \\ b \end{array} \rrbracket &= \sum_{k=0}^a \sum_{\ell=0}^b |\ell, k\rangle \langle k, \ell| \\ \llbracket D_1 \otimes D_2 \rrbracket &= \llbracket D_1 \rrbracket \otimes \llbracket D_2 \rrbracket & \llbracket \begin{array}{c} \dots \\ \text{---} \\ \text{---} \\ \dots \\ a \\ \text{---} \\ \text{---} \\ m \end{array} \rrbracket &= \sum_{k=0}^a r^k \sqrt{k!}^{n+m-2} |k^m\rangle \langle k^n| \\ \llbracket \begin{array}{c} | \\ \text{---} \\ a \end{array} \rrbracket &= \sum_{k=0}^a |k\rangle \langle k| & \llbracket \begin{array}{c} \dots \\ \text{---} \\ \text{---} \\ \dots \\ \bullet \\ \text{---} \\ a \end{array} \rrbracket &= \begin{cases} \sqrt{k!} |k\rangle & \text{if } 0 < k \leq a \\ \vec{0} & \text{otherwise} \end{cases} \\ \llbracket \begin{array}{c} \text{---} \\ \text{---} \\ a \end{array} \rrbracket &= \llbracket \begin{array}{c} \text{---} \\ \text{---} \\ a \end{array} \rrbracket^\dagger = \sum_{k=0}^a |k, k\rangle & \llbracket \begin{array}{c} a \\ \text{---} \\ \text{---} \\ b_1 \dots b_n \end{array} \rrbracket &= \sum_{\substack{0 \leq k_i \leq b_i \\ k_1 + \dots + k_n \leq a}} \sqrt{\binom{k_1 + \dots + k_n}{k_1, \dots, k_n}} |k_1, \dots, k_n\rangle \langle k_1 + \dots + k_n| \\ \llbracket r \rrbracket &= r \end{aligned}$$

Notice that we use the same notation for the interpretation of **ZW_d**-diagrams, and for the interpretation of the **ZW_f**-diagrams. Which interpretation we are referring to should be clear from the context.

3.2 Complete Equational Theory

We once again equip the language with an equational theory **ZW_f**, defined in Figure 2. This equational theory only slightly differs from the one for qudit systems in Figure 1. It is interesting to notice that 1) the associativity of the W-node is broken down into two equations (a) and (a'), whose choice depends on the capacities involved, 2) the W-bialgebra equation (b₂) does not need a context anymore, but instead side conditions on the capacities, 3) the equation that allows for a normal form is greatly simplified, and 4) we now have a version of the copy of a white node through a W-node (xp).

The category **Qudit_d** is a full subcategory of **FdHilb**, and as such there is an obvious inclusion functor **Qudit_d** $\xrightarrow{i_d}$ **FdHilb**. This inclusion transports to the ZW-calculi: we can turn any **ZW_d**-diagram into a **ZW_f**-diagram through ι_d in such a way that the following diagram commutes:

$$\begin{array}{ccc} \mathbf{ZW}_d & \xleftarrow{\iota_d} & \mathbf{ZW}_f \\ \llbracket \cdot \rrbracket \downarrow & & \downarrow \llbracket \cdot \rrbracket \\ \mathbf{Qudit}_d & \xleftarrow{i_d} & \mathbf{FdHilb} \end{array}$$

The functor ι_d simply takes a **ZW_d**-diagram and annotates all its wires with $d - 1$.

We show that the present equational theory is complete. To do so, we need to adapt the notion of normal form from qudit systems (and we again use the map/state duality to focus on states rather than arbitrary morphisms):

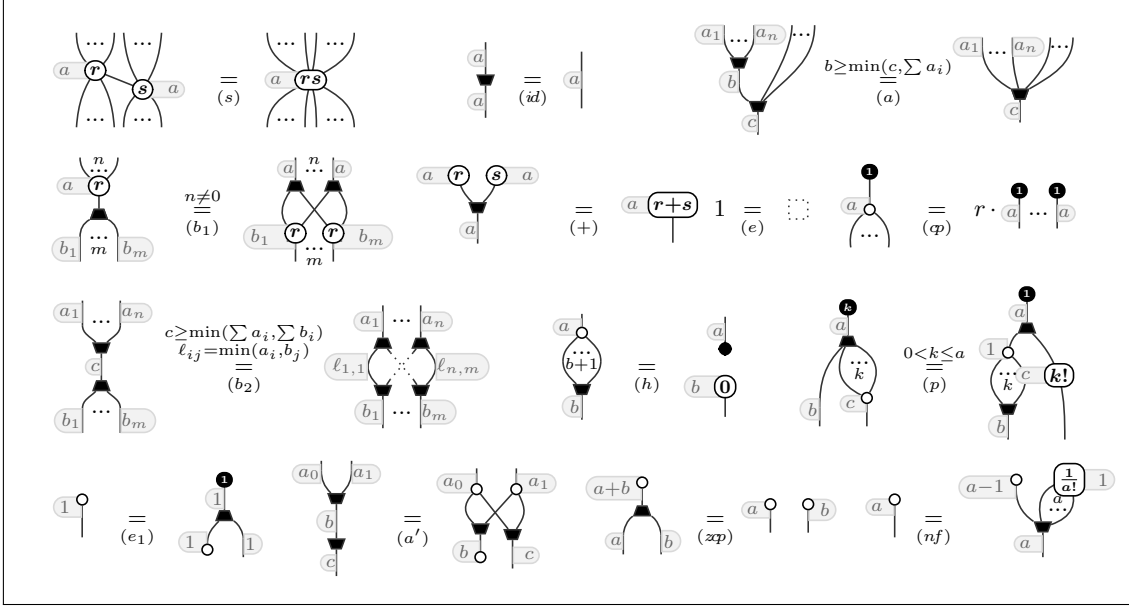
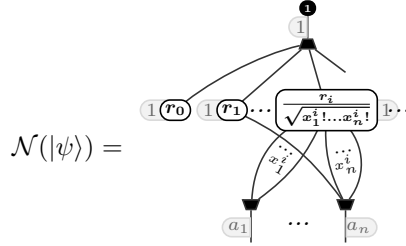


Fig. 2: Equational theory \mathbf{ZW}_f for the finite-dimensional \mathbf{ZW} -calculus.

Definition 4. We define $\mathcal{N} : \mathbf{FdHilb} \rightarrow \mathbf{ZW}_f$ as the functor such that, for any n -ary state $|\psi\rangle = r_0 |0\dots 0\rangle + r_1 |0\dots 01\rangle + \dots + r_i |x^1\dots x_n^i\rangle + \dots \in \mathbf{FdHilb}[\langle \cdot \rangle, \langle a_1, \dots, a_n \rangle]$:



We say of any diagram in the image of \mathcal{N} that it is in normal form. Notice that $x_j^i \leq a_j$ for all $\langle i \rangle \leq i < a_1\dots a_n$ and $1 \leq j \leq n$.

We can once again show that \mathcal{N} builds a diagram that represents $|\psi\rangle$:

Lemma 10.

$$\forall |\psi\rangle \in \mathbf{FdHilb}[0, \langle a_1, \dots, a_n \rangle], \llbracket \mathcal{N}(|\psi\rangle) \rrbracket = |\psi\rangle$$

We hence get universality of the language as a direct consequence:

Corollary 3.

$$\begin{aligned} \forall f \in \mathbf{FdHilb}[\langle a_1, \dots, a_n \rangle, \langle b_1, \dots, b_m \rangle], \\ \exists D_f \in \mathbf{ZW}_f[\langle a_1, \dots, a_n \rangle, \langle b_1, \dots, b_m \rangle], \llbracket D_f \rrbracket = f \end{aligned}$$

Using the normal form, we can then leverage the completeness from the qudit \mathbf{ZW} -calculus to get the similar result in the current setting:

Theorem 3 (Completeness for Finite Dimensional Systems). *The language is complete: for any two diagrams D_1 and D_2 of the \mathbf{ZW}_f -calculus:*

$$\llbracket D_1 \rrbracket = \llbracket D_2 \rrbracket \iff \mathbf{ZW}_f \vdash D_1 = D_2$$

Proof. The proof, in its entirety in Appendix C, uses the completeness of \mathbf{ZW}_d for qudit systems. We once again use the map / state duality to restrict the proof to states. Then, we proceed as follows:

1. We show that all equations of ZW_d are derivable from ZW_f , i.e. that $ZW_d \vdash D_1 = D_2 \implies ZW_f \vdash \iota_d(D_1) = \iota_d(D_2)$ for all $d \geq 2$
2. We show that any ZW_f -diagram D , with maximum capacity $\leq d - 1$ can be turned into

$$ZW_f \vdash \begin{array}{c} \boxed{D} \\ \hline a_1 \quad \dots \quad a_n \end{array} = \begin{array}{c} \boxed{\iota_d(D_d)} \\ \hline \begin{array}{c} d \quad \dots \quad d \\ \uparrow \quad \dots \quad \uparrow \\ a_1 \quad \dots \quad a_n \end{array} \end{array}$$

for some ZW_d -diagram D_d

3. Using rules of ZW_d , we can turn D_d in (qudit) normal form
4. We show that $\iota_d(D)$ with D in ZW_d normal form can be turned in ZW_f normal form. Consequently, any ZW_f -diagram can be put in ZW_f normal form
5. We conclude thanks to the uniqueness of the ZW_f normal form.

Related Work Another complete presentation of a graphical language for **FdHilb** was recently announced [36]. This one builds upon the aforementioned ZXW-calculus, and introduces a new generator that takes two systems, of dimensions a and b , and builds a system of dimension $a \times b$. Our approach builds upon ZW_d , the qudit version of the ZW-calculus from Section 2 and hence starts with fewer generators and equations. As a consequence, the graphical language for **FdHilb** we end up with has fewer equations as well.

Moreover, we did not require a new generator, and simply promoted the qudit W-node to work with any mix of dimensions in a natural manner, which was enough to provide us with universality. This way we deal with dimensions is less constraining, as the generator used in [36] only allows to multiply dimensions together. For instance, a (non-trivial) $\langle 2 \rangle \rightarrow \langle 3 \rangle$ diagram would require at least going through a 6-dimensional system, while in our setting one can directly go from a 2-dimensional system to a 3-dimensional one.

4 Conclusion

In this paper, we explored the potential for a minimal yet complete diagrammatic language for quantum mechanics beyond qubit systems. This starts with a well-chosen generalisation of the generators of the ZW-calculus, allowing us to have few and (mostly) intuitive equations. For both qudit systems and finite dimensional systems, we showed that the diagrams are universal, and that the equational theories are complete for their respective interpretation. In the case of qudit systems, we further showed that the equational theory was minimal. As we used it to build the equational theory for finite dimensional systems, we conjecture the latter is also minimal, or very close to being so. It turns out that transporting the minimality theorem from the first setting to the second one is challenging, as some minimality arguments break when adding or modifying equations. Proving minimality in the second setting hence requires a fair amount of work, and is hence left as an open question.

References

1. S. Abramsky & B. Coecke (2004): *A categorical semantics of quantum protocols*. In: *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, pp. 415–425, doi:10.1109/LICS.2004.1319636.
2. Miriam Backens (2014): *The ZX-Calculus is Complete for Stabilizer Quantum Mechanics*. In: *New Journal of Physics*, 16, IOP Publishing, p. 093021, doi:10.1088/1367-2630/16/9/093021. Available at <https://doi.org/10.1088/1367-2630/16/9/093021>.
3. Miriam Backens & Aleks Kissinger (2019): *ZH: A Complete Graphical Calculus for Quantum Computations Involving Classical Non-linearity*. In Peter Selinger & Giulio Chiribella, editors: *Proceedings of the 15th International Conference on Quantum Physics and Logic, Halifax, Canada, 3-7th June 2018, Electronic Proceedings in Theoretical Computer Science* 287, pp. 23–42, doi:10.4204/EPTCS.287.2.
4. Miriam Backens, Aleks Kissinger, Hector Miller-Bakewell, John van de Wetering & Sal Wolfs (2023): *Completeness of the ZH-calculus*. *Compositionality* 5, doi:10.32408/compositionality-5-5. Available at <https://doi.org/10.32408/compositionality-5-5>.

5. Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski & John van de Wetering (2020): *There and back again: A circuit extraction tale*. arXiv: Quantum Physics. Available at <https://arxiv.org/abs/2003.01664>.
6. Niel de Beaudrap & Dominic Horsman (2020): *The ZX calculus is a language for surface code lattice surgery*. *Quantum* 4, p. 218, doi:10.22331/q-2020-01-09-218. Available at <https://doi.org/10.22331/q-2020-01-09-218>.
7. Robert I. Booth & Titouan Carette (2022): *Complete ZX-Calculi for the Stabiliser Fragment in Odd Prime Dimensions*. In Stefan Szeider, Robert Ganian & Alexandra Silva, editors: *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 241, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 24:1–24:15, doi:10.4230/LIPIcs.MFCS.2022.24. Available at <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.MFCS.2022.24>.
8. Titouan Carette (2021): *Wielding the ZX-calculus, Flexsymmetry, Mixed States, and Scalable Notations*. (Manier le ZX-calcul, flexsymétrie, systèmes ouverts et limandes). Ph.D. thesis, University of Lorraine, Nancy, France. Available at <https://tel.archives-ouvertes.fr/tel-03468027>.
9. Titouan Carette, Emmanuel Jeandel, Simon Perdrix & Renaud Vilmart (2019): *Completeness of Graphical Languages for Mixed States Quantum Mechanics*. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini & Stefano Leonardi, editors: *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 132, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 108:1–108:15, doi:10.4230/LIPIcs.ICALP.2019.108. Available at <http://drops.dagstuhl.de/opus/volltexte/2019/10684>.
10. Titouan Carette, Etienne Moutot, Thomas Perez & Renaud Vilmart (2023): *Compositionality of Planar Perfect Matchings: A Universal and Complete Fragment of ZW-Calculus*. In Kousha Etesami, Uriel Feige & Gabriele Puppis, editors: *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 261, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 120:1–120:17, doi:10.4230/LIPIcs.ICALP.2023.120. Available at <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2023.120>.
11. Bob Coecke & Ross Duncan (2011): *Interacting Quantum Observables: Categorical Algebra and Diagrammatics*. *New Journal of Physics* 13(4), p. 043016, doi:10.1088/1367-2630/13/4/043016. Available at <https://doi.org/10.1088/1367-2630/13/4/043016>.
12. Bob Coecke & Aleks Kissinger (2010): *The Compositional Structure of Multipartite Quantum Entanglement*. In: *Automata, Languages and Programming*, Springer Berlin Heidelberg, pp. 297–308, doi:10.1007/978-3-642-14162-1_25. Available at https://doi.org/10.1007/978-3-642-14162-1_25.
13. Bob Coecke & Aleks Kissinger (2017): *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, Cambridge, doi:10.1017/9781316219317.
14. Joseph Collins & Ross Duncan (2020): *Hopf-Frobenius Algebras and a Simpler Drinfeld Double*. *Electronic Proceedings in Theoretical Computer Science* 318, p. 150–180, doi:10.4204/eptcs.318.10. Available at <http://dx.doi.org/10.4204/EPTCS.318.10>.
15. Ross Duncan & Kevin Dunne (2016): *Interacting Frobenius Algebras are Hopf*. In: *2016 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 1–10.
16. Ross Duncan & Maxime Lucas (2014): *Verifying the Steane code with Quantomatic*. In Bob Coecke & Matty Hoban, editors: *Proceedings of the 10th International Workshop on Quantum Physics and Logic, Castelldefels (Barcelona), Spain, 17th to 19th July 2013*, *Electronic Proceedings in Theoretical Computer Science* 171, Open Publishing Association, pp. 33–49, doi:10.4204/EPTCS.171.4.
17. Giovanni de Felice & Bob Coecke (2022): *Quantum Linear Optics via String Diagrams*.
18. Amar Hadzihasanovic (2015): *A Diagrammatic Axiomatisation for Qubit Entanglement*. In: *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pp. 573–584, doi:10.1109/LICS.2015.59.
19. Amar Hadzihasanovic (2017): *The Algebra of Entanglement and the Geometry of Composition*. Ph.D. thesis, University of Oxford. Available at <https://arxiv.org/abs/1403.7828>.
20. Amar Hadzihasanovic, Kang Feng Ng & Quanlong Wang (2018): *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, ACM, New York, NY, USA, pp. 502–511, doi:10.1145/3209108.3209128. Available at <http://doi.acm.org/10.1145/3209108.3209128>.
21. Anne Hillebrand (2011): *Quantum Protocols involving Multiparticle Entanglement and their Representations*. Master's thesis, University of Oxford. Available at <https://www.cs.ox.ac.uk/people/bob.coecke/Anne.pdf>.

22. Jiaxin Huang, Sarah Meng Li, Lia Yeh, Aleks Kissinger, Michele Mosca & Michael Vasmer (2023): *Graphical CSS Code Transformation Using ZX Calculus*. *Electronic Proceedings in Theoretical Computer Science* 384, p. 1–19, doi:10.4204/eptcs.384.1. Available at <http://dx.doi.org/10.4204/EPTCS.384.1>.
23. Emmanuel Jeandel, Simon Perdrix & Renaud Vilmart (2018): *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*, ACM, New York, NY, USA, pp. 559–568, doi:10.1145/3209108.3209131. Available at <http://doi.acm.org/10.1145/3209108.3209131>.
24. Emmanuel Jeandel, Simon Perdrix & Renaud Vilmart (2020): *Completeness of the ZX-Calculus*. *Logical Methods in Computer Science* Volume 16, Issue 2, doi:10.23638/LMCS-16(2:11)2020. Available at <https://lmcs.episciences.org/6532>.
25. Aleks Kissinger & John van de Wetering (2020): *Reducing the number of non-Clifford gates in quantum circuits*. *Phys. Rev. A* 102, p. 022406, doi:10.1103/PhysRevA.102.022406. Available at <https://link.aps.org/doi/10.1103/PhysRevA.102.022406>.
26. Aleks Kissinger & John van de Wetering (2022): *Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions*. *Quantum Science and Technology* 7(4), p. 044001, doi:10.1088/2058-9565/ac5d20. Available at <https://dx.doi.org/10.1088/2058-9565/ac5d20>.
27. Aleks Kissinger, John van de Wetering & Renaud Vilmart (2022): *Classical Simulation of Quantum Circuits with Partial and Graphical Stabiliser Decompositions*. In François Le Gall & Tomoyuki Morimae, editors: *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 232, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 5:1–5:13, doi:10.4230/LIPIcs.TQC.2022.5. Available at <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TQC.2022.5>.
28. Mark Koch, Richie Yeung & Quanlong Wang (2023): *Speedy Contraction of ZX Diagrams with Triangles via Stabiliser Decompositions*.
29. Stephen Lack (2004): *Composing PROPs*. In: *Theory and Applications of Categories*, 13, pp. 147–163. Available at <http://www.tac.mta.ca/tac/volumes/13/9/13-09abs.html>.
30. Boldizsár Poór, Quanlong Wang, Razin A. Shaikh, Lia Yeh, Richie Yeung & Bob Coecke (2023): *Completeness for arbitrary finite dimensions of ZXW-calculus, a unifying calculus*. In: *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 1–14, doi:10.1109/LICS56636.2023.10175672.
31. Peter Selinger (2010): *A Survey of Graphical Languages for Monoidal Categories*. In: *New Structures for Physics*, Springer, pp. 289–355.
32. Alex Townsend-Teague, Julio Magdalena de la Fuente & Markus Kesselring (2023): *Floquetifying the Colour Code*. *Electronic Proceedings in Theoretical Computer Science* 384, p. 265–303, doi:10.4204/eptcs.384.14. Available at <http://dx.doi.org/10.4204/EPTCS.384.14>.
33. Renaud Vilmart (2019): *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)*, pp. 1–10, doi:10.1109/LICS.2019.8785765.
34. Quanlong Wang (2018): *Qutrit ZX-calculus is Complete for Stabilizer Quantum Mechanics*. In Bob Coecke & Aleks Kissinger, editors: *Proceedings 14th International Conference on Quantum Physics and Logic, Nijmegen, The Netherlands, 3-7 July 2017*, *Electronic Proceedings in Theoretical Computer Science* 266, pp. 58–70, doi:10.4204/EPTCS.266.3.
35. Quanlong Wang (2021): *A non-anyonic qudit ZW-calculus*.
36. Quanlong Wang & Boldizsár Poór (2023): *Completeness of qufinite ZXW calculus, a graphical language for mixed-dimensional quantum computing*.
37. Fabio Zanasi (2015): *Interacting Hopf Algebras – the theory of linear systems*. Ph.D. thesis, Université de Lyon. Available at <http://www.zanasi.com/fabio/#/publications.html>.

A Auxiliary Lemmas and Proofs For Qudit

In this section, we prove all the lemmas that were left unproven in the body of the article. To do so, we introduce the following auxiliary lemmas:

Lemma 11. *If $0 < k < d$:*

$$\text{ZW}_d \vdash k! \begin{array}{c} \bullet \\ | \\ \bullet \end{array} = \begin{array}{c} \bullet \\ \vdots \\ \bullet \\ \circ \\ \vdots \\ \bullet \\ \circ \end{array}$$

Lemma 12.

$$\text{ZW}_d \vdash \begin{array}{c} \bullet \\ | \\ \bullet \\ \circ \end{array} = \begin{array}{c} \bullet \\ | \\ \bullet \end{array} \begin{array}{c} \bullet \\ | \\ \bullet \end{array}$$

Lemma 13.

$$\text{ZW}_d \vdash \begin{array}{c} \bullet \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} = \begin{array}{c} \bullet \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & & \uparrow \\ \bullet & & \bullet \end{array} \end{array}$$

Lemma 17. *If $0 \leq k < d$, we have:*

$$\text{ZW}_d \vdash \begin{array}{c} \bullet \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} = \begin{array}{c} \bullet \\ \downarrow \\ \begin{array}{c} \circ \\ \uparrow \\ \bullet \end{array} \end{array}$$

Lemma 14. *If $k \neq \ell$:*

$$\text{ZW}_d \vdash \begin{array}{c} \bullet \\ \downarrow \\ \begin{array}{c} \circ \\ \uparrow \\ \bullet \end{array} \end{array} = k! \quad \text{and} \quad \text{ZW}_d \vdash \begin{array}{c} \bullet \\ \downarrow \\ \begin{array}{c} \circ \\ \uparrow \\ \bullet \end{array} \end{array} = 0$$

Lemma 18. *The bialgebra between W -nodes can be used in the following context, if $0 \leq k_i < d$ for all i :*

Lemma 15.

$$\text{ZW}_d \vdash \begin{array}{c} \bullet \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} = r \cdot \begin{array}{c} \bullet \\ \downarrow \\ \begin{array}{c} \circ \\ \uparrow \\ \bullet \end{array} \end{array}$$

$$\text{ZW}_d \vdash \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} = \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array}$$

Lemma 16. *If $0 \leq k < d$:*

$$\text{ZW}_d \vdash \begin{array}{c} \bullet \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} = \begin{array}{c} \bullet \\ \downarrow \\ \begin{array}{c} \circ \\ \uparrow \\ \bullet \end{array} \end{array}$$

Now, since the lemmas' ordering does not follow their dependencies any more, we provide in Figure 3 a small (automatically generated) graph of all dependencies between them, so as to convince the reader that there is no circular proof. Equations from the equational theories are not displayed in the graph, they are assumed to be available at any point.

Proof (Proof of Lemma 1).

$$\text{ZW}_d \vdash \begin{array}{c} \bullet \\ \downarrow \\ \circ \end{array} \stackrel{(s)}{=} \begin{array}{c} \bullet \\ \downarrow \\ \circ \end{array} \stackrel{(id)}{=} \begin{array}{c} \bullet \\ \downarrow \\ \circ \end{array} \stackrel{(b_2)}{=} \begin{array}{c} \bullet \\ \downarrow \\ \circ \end{array} \stackrel{(id)}{=} \begin{array}{c} \bullet \\ \downarrow \\ \circ \end{array}$$

Proof (Proof of Lemma 2).

$$\text{ZW}_d \vdash \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} \stackrel{(b_2)}{=} \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} \stackrel{(s)}{=} \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} \stackrel{(b_1)}{=} \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} \stackrel{(s)}{=} \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array}$$

Proof (Proof of Lemma 3).

$$\text{ZW}_d \vdash \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} \stackrel{(cp)}{=} \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} \stackrel{(id)}{=} \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} \stackrel{(cp)}{=} \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array} \stackrel{(id)}{=} \begin{array}{c} \dots \\ \downarrow \\ \begin{array}{ccc} \circ & & \circ \\ \uparrow & \uparrow & \uparrow \\ \bullet & & \bullet \end{array} \end{array}$$

Proof (Proof of Lemma 4).

$$\text{ZW}_d \vdash \begin{array}{c} \bullet \\ \downarrow \\ \circ \end{array} \stackrel{(0)}{=} \begin{array}{c} \bullet \\ \downarrow \\ \circ \end{array} \stackrel{(h)}{=} \begin{array}{c} \bullet \\ \downarrow \\ \circ \end{array} \stackrel{(b_1)}{=} \begin{array}{c} \bullet \\ \downarrow \\ \circ \end{array} \stackrel{(a)}{=} \begin{array}{c} \bullet \\ \downarrow \\ \circ \end{array}$$

Proof (Proof of Lemma 5).

$$\text{ZW}_d \vdash \begin{array}{c} \begin{array}{cc} \circ & \circ \\ \uparrow & \uparrow \\ \bullet & \bullet \end{array} \\ \downarrow \\ \dots \end{array} \stackrel{(s)}{=} \begin{array}{c} \begin{array}{cc} \circ & \circ \\ \uparrow & \uparrow \\ \bullet & \bullet \end{array} \\ \downarrow \\ \dots \end{array} \stackrel{(+)}{=} \begin{array}{c} \begin{array}{cc} \circ & \circ \\ \uparrow & \uparrow \\ \bullet & \bullet \end{array} \\ \downarrow \\ \dots \end{array} \stackrel{(s)}{=} \begin{array}{c} \begin{array}{cc} \circ & \circ \\ \uparrow & \uparrow \\ \bullet & \bullet \end{array} \\ \downarrow \\ \dots \end{array}$$

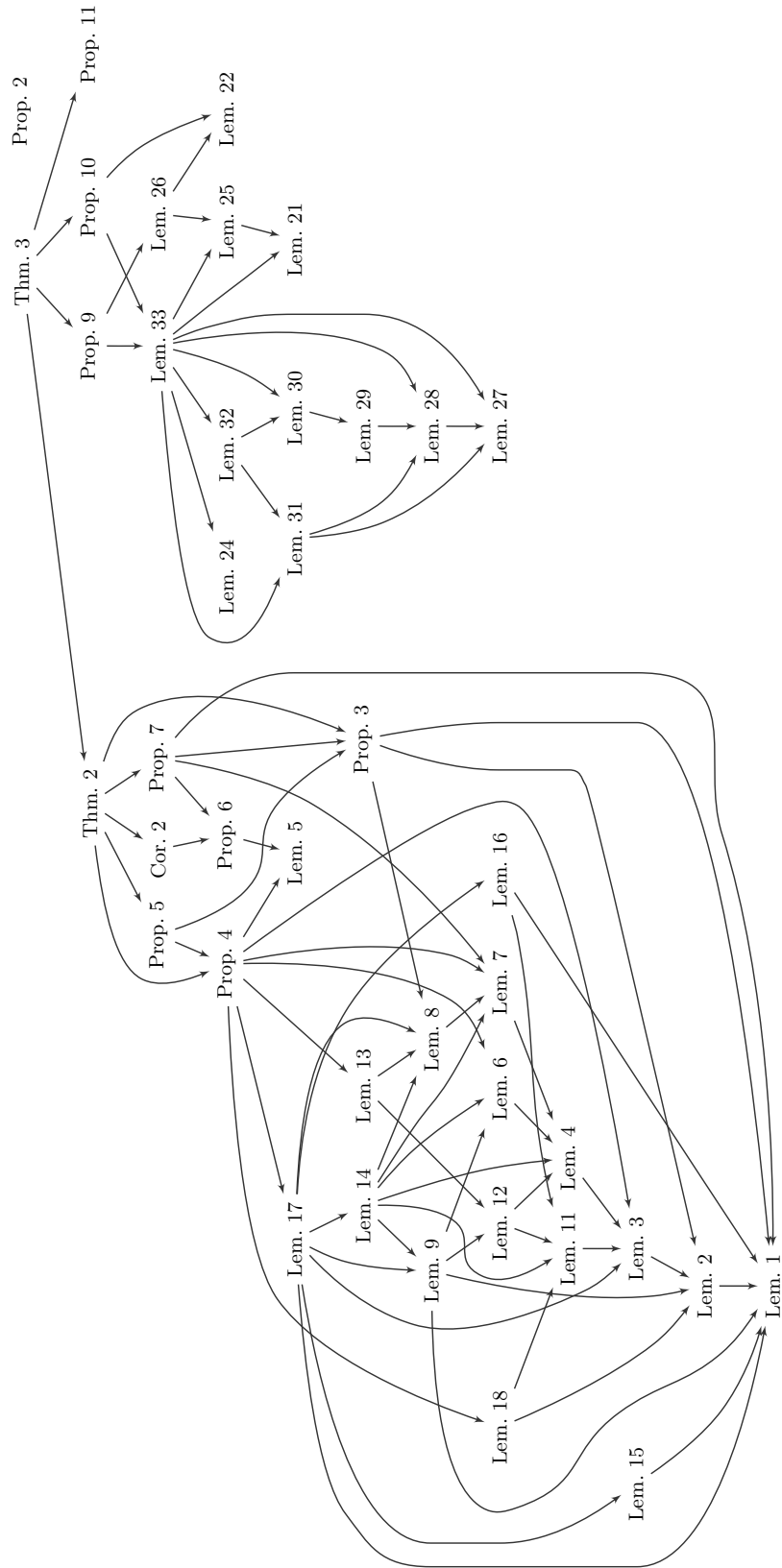
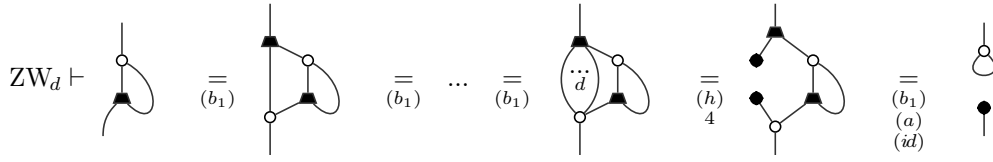
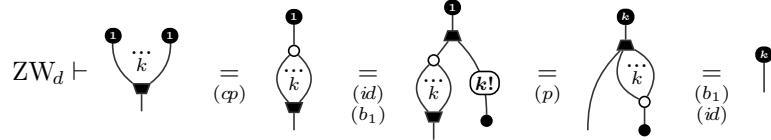


Fig. 3: Dependencies between lemmas, corollaries, propositions and theorems.

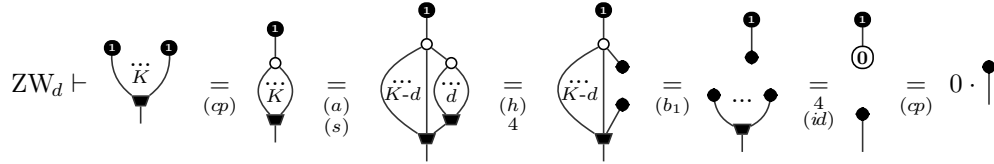
Proof (Proof of Lemma 6).



Proof (Proof of Lemma 7). First, notice that if $k = 0$ or $\ell = 0$, the equation is obvious by (id) . Then, if $k < d$:



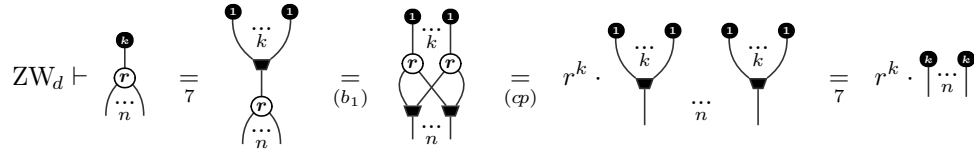
Otherwise, if $K \geq d$:



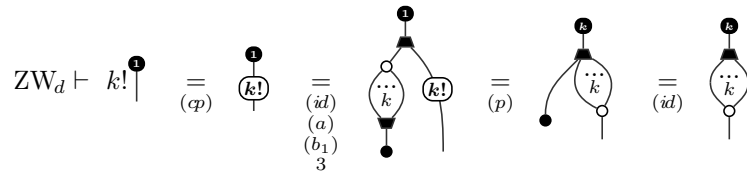
Hence:

$$ZW_d \vdash \begin{array}{c} \bullet \\ \vdots \\ \bullet \end{array} = \begin{array}{c} \bullet \\ \vdots \\ \bullet \end{array} = \begin{cases} \begin{array}{c} \bullet \\ \vdots \\ \bullet \end{array} & \text{if } k + \ell < d \\ \vec{0} & \text{if } k + \ell \geq d \end{cases}$$

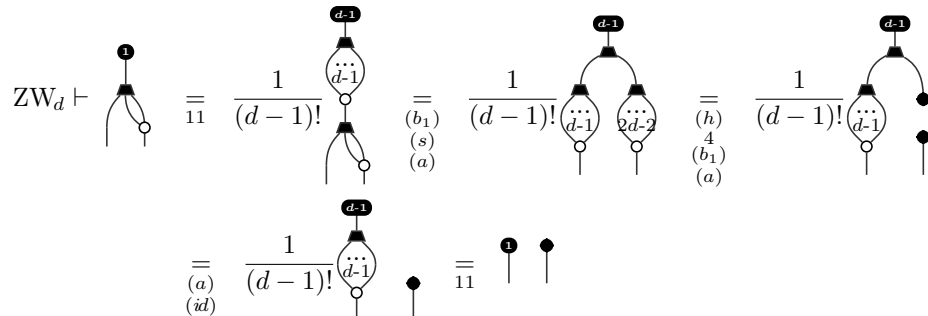
Proof (Proof of Lemma 8).



Proof (Proof of Lemma 11).



Proof (Proof of Lemma 12).



Proof (Proof of Lemma 9). The proof goes in three steps:

$$\begin{array}{c}
 \text{ZW}_d \vdash \begin{array}{c} \text{Diagram 1} \\ \text{Diagram 2} \\ \text{Diagram 3} \\ \text{Diagram 4} \\ \text{Diagram 5} \end{array} \stackrel{(cp)}{=} \begin{array}{c} \text{Diagram 6} \\ \text{Diagram 7} \\ \text{Diagram 8} \\ \text{Diagram 9} \\ \text{Diagram 10} \end{array} \stackrel{(b_1)}{=} \begin{array}{c} \text{Diagram 11} \\ \text{Diagram 12} \\ \text{Diagram 13} \\ \text{Diagram 14} \\ \text{Diagram 15} \end{array} \stackrel{2}{=} \begin{array}{c} \text{Diagram 16} \\ \text{Diagram 17} \\ \text{Diagram 18} \\ \text{Diagram 19} \\ \text{Diagram 20} \end{array} \stackrel{(s)}{=} \begin{array}{c} \text{Diagram 21} \\ \text{Diagram 22} \\ \text{Diagram 23} \\ \text{Diagram 24} \\ \text{Diagram 25} \end{array} \stackrel{(b_1)}{=} \begin{array}{c} \text{Diagram 26} \\ \text{Diagram 27} \\ \text{Diagram 28} \\ \text{Diagram 29} \\ \text{Diagram 30} \end{array} \stackrel{(a)}{=} \begin{array}{c} \text{Diagram 31} \\ \text{Diagram 32} \\ \text{Diagram 33} \\ \text{Diagram 34} \\ \text{Diagram 35} \end{array} \\
 \end{array} \tag{1}$$

$$\begin{array}{c}
 \text{ZW}_d \vdash \begin{array}{c} \text{Diagram 36} \\ \text{Diagram 37} \\ \text{Diagram 38} \\ \text{Diagram 39} \\ \text{Diagram 40} \end{array} \stackrel{(s)}{=} \begin{array}{c} \text{Diagram 41} \\ \text{Diagram 42} \\ \text{Diagram 43} \\ \text{Diagram 44} \\ \text{Diagram 45} \end{array} \stackrel{(1)}{=} \begin{array}{c} \text{Diagram 46} \\ \text{Diagram 47} \\ \text{Diagram 48} \\ \text{Diagram 49} \\ \text{Diagram 50} \end{array} \stackrel{(b_1)}{=} \begin{array}{c} \text{Diagram 51} \\ \text{Diagram 52} \\ \text{Diagram 53} \\ \text{Diagram 54} \\ \text{Diagram 55} \end{array} \stackrel{(a)}{=} \begin{array}{c} \text{Diagram 56} \\ \text{Diagram 57} \\ \text{Diagram 58} \\ \text{Diagram 59} \\ \text{Diagram 60} \end{array} \stackrel{(s)}{=} \begin{array}{c} \text{Diagram 61} \\ \text{Diagram 62} \\ \text{Diagram 63} \\ \text{Diagram 64} \\ \text{Diagram 65} \end{array} \stackrel{(nf)}{=} \begin{array}{c} \text{Diagram 66} \\ \text{Diagram 67} \\ \text{Diagram 68} \\ \text{Diagram 69} \\ \text{Diagram 70} \end{array} \stackrel{(s)}{=} \begin{array}{c} \text{Diagram 71} \\ \text{Diagram 72} \\ \text{Diagram 73} \\ \text{Diagram 74} \\ \text{Diagram 75} \end{array} \\
 \end{array} \tag{2}$$

$$\text{ZW}_d \vdash \begin{array}{c} \text{Diagram 76} \\ \text{Diagram 77} \\ \text{Diagram 78} \\ \text{Diagram 79} \\ \text{Diagram 80} \end{array} \stackrel{(1)}{=} \begin{array}{c} \text{Diagram 81} \\ \text{Diagram 82} \\ \text{Diagram 83} \\ \text{Diagram 84} \\ \text{Diagram 85} \end{array} \stackrel{(2)}{=} \begin{array}{c} \text{Diagram 86} \\ \text{Diagram 87} \\ \text{Diagram 88} \\ \text{Diagram 89} \\ \text{Diagram 90} \end{array}$$

Proof (Proof of Lemma 13).

$$\text{ZW}_d \vdash \begin{array}{c} \text{Diagram 91} \\ \text{Diagram 92} \\ \text{Diagram 93} \\ \text{Diagram 94} \\ \text{Diagram 95} \end{array} \stackrel{(b_1)}{=} \begin{array}{c} \text{Diagram 96} \\ \text{Diagram 97} \\ \text{Diagram 98} \\ \text{Diagram 99} \\ \text{Diagram 100} \end{array} \stackrel{12}{=} \begin{array}{c} \text{Diagram 101} \\ \text{Diagram 102} \\ \text{Diagram 103} \\ \text{Diagram 104} \\ \text{Diagram 105} \end{array} \stackrel{8}{=} \begin{array}{c} \text{Diagram 106} \\ \text{Diagram 107} \\ \text{Diagram 108} \\ \text{Diagram 109} \\ \text{Diagram 110} \end{array} \stackrel{(id)}{=} \begin{array}{c} \text{Diagram 111} \\ \text{Diagram 112} \\ \text{Diagram 113} \\ \text{Diagram 114} \\ \text{Diagram 115} \end{array}$$

Proof (Proof of Lemma 14). First:

$$\text{ZW}_d \vdash \begin{array}{c} \text{Diagram 116} \\ \text{Diagram 117} \\ \text{Diagram 118} \\ \text{Diagram 119} \\ \text{Diagram 120} \end{array} \stackrel{(cp)}{=} \begin{array}{c} \text{Diagram 121} \\ \text{Diagram 122} \\ \text{Diagram 123} \\ \text{Diagram 124} \\ \text{Diagram 125} \end{array} \stackrel{(id)}{=} \begin{array}{c} \text{Diagram 126} \\ \text{Diagram 127} \\ \text{Diagram 128} \\ \text{Diagram 129} \\ \text{Diagram 130} \end{array} \stackrel{(a)}{=} \begin{array}{c} \text{Diagram 131} \\ \text{Diagram 132} \\ \text{Diagram 133} \\ \text{Diagram 134} \\ \text{Diagram 135} \end{array} \stackrel{9}{=} \begin{array}{c} \text{Diagram 136} \\ \text{Diagram 137} \\ \text{Diagram 138} \\ \text{Diagram 139} \\ \text{Diagram 140} \end{array} \stackrel{(id)}{=} \begin{array}{c} \text{Diagram 141} \\ \text{Diagram 142} \\ \text{Diagram 143} \\ \text{Diagram 144} \\ \text{Diagram 145} \end{array} \stackrel{(cp)}{=} \begin{array}{c} \text{Diagram 146} \\ \text{Diagram 147} \\ \text{Diagram 148} \\ \text{Diagram 149} \\ \text{Diagram 150} \end{array} \stackrel{1}{=} \begin{array}{c} \text{Diagram 151} \\ \text{Diagram 152} \\ \text{Diagram 153} \\ \text{Diagram 154} \\ \text{Diagram 155} \end{array}$$

Then:

$$\text{ZW}_d \vdash \begin{array}{c} \text{Diagram 156} \\ \text{Diagram 157} \\ \text{Diagram 158} \\ \text{Diagram 159} \\ \text{Diagram 160} \end{array} \stackrel{7}{=} \begin{array}{c} \text{Diagram 161} \\ \text{Diagram 162} \\ \text{Diagram 163} \\ \text{Diagram 164} \\ \text{Diagram 165} \end{array} \stackrel{(cp)}{=} \begin{array}{c} \text{Diagram 166} \\ \text{Diagram 167} \\ \text{Diagram 168} \\ \text{Diagram 169} \\ \text{Diagram 170} \end{array} \stackrel{11}{=} \begin{array}{c} \text{Diagram 171} \\ \text{Diagram 172} \\ \text{Diagram 173} \\ \text{Diagram 174} \\ \text{Diagram 175} \end{array} \stackrel{1}{=} k!$$

Finally:

$$\text{ZW}_d \vdash \begin{array}{c} \text{Diagram 176} \\ \text{Diagram 177} \\ \text{Diagram 178} \\ \text{Diagram 179} \\ \text{Diagram 180} \end{array} \stackrel{7}{=} \begin{array}{c} \text{Diagram 181} \\ \text{Diagram 182} \\ \text{Diagram 183} \\ \text{Diagram 184} \\ \text{Diagram 185} \end{array} \stackrel{8}{=} \begin{array}{c} \text{Diagram 186} \\ \text{Diagram 187} \\ \text{Diagram 188} \\ \text{Diagram 189} \\ \text{Diagram 190} \end{array} \stackrel{6}{=} \begin{array}{c} \text{Diagram 191} \\ \text{Diagram 192} \\ \text{Diagram 193} \\ \text{Diagram 194} \\ \text{Diagram 195} \end{array} \stackrel{(id)}{=} \begin{array}{c} \text{Diagram 196} \\ \text{Diagram 197} \\ \text{Diagram 198} \\ \text{Diagram 199} \\ \text{Diagram 200} \end{array} \stackrel{(b_1)}{=} \begin{array}{c} \text{Diagram 201} \\ \text{Diagram 202} \\ \text{Diagram 203} \\ \text{Diagram 204} \\ \text{Diagram 205} \end{array} \stackrel{11}{=} (k-l)! \begin{array}{c} \text{Diagram 206} \\ \text{Diagram 207} \\ \text{Diagram 208} \\ \text{Diagram 209} \\ \text{Diagram 210} \end{array}$$

$$\stackrel{=}{4} (k-l)! \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(\varphi)} 0$$

Proof (Proof of Lemma 15).

$$\text{ZW}_d \vdash \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(s)} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(b_1)} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(\varphi)} r \cdot \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array}$$

Proof (Proof of Lemma 16). First notice that the equation derives from (b_1) when $k = 0$. Then, for $0 < k < d$:

$$\begin{array}{c} \text{ZW}_d \vdash \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{11} \frac{1}{k!} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(b_1) (s) (a)} \frac{1}{k!} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(s) (b_1)} \frac{1}{k!} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(s) (a)} \frac{1}{k!} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \\ \stackrel{=}{(b_1) (s) (a)} \frac{1}{k!} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{11} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \end{array}$$

Proof (Proof of Lemma 17). If $k = 0$, the equation derives from Lemma 8 and Lemma 3. Else, if $k < d$:

$$\begin{array}{c} \text{ZW}_d \vdash \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(b_1) (s)} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{9} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(s) (p)} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(b_1) (s) (a)} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \\ \stackrel{=}{(b_1) (a)} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{8} \frac{1}{k!} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(s) (p)} \frac{1}{k!} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{16} \frac{1}{k!} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{9} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(s)} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \end{array}$$

Proof (Proof of Lemma 18). To apply (b_2) , we would need to have k_i connections between the leftmost W -node and the i -th W -node of the bottom of the bialgebra, but so far we only have one. We can get more, in the following way:

$$\text{ZW}_d \vdash \begin{array}{c} \dots \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{11} \frac{1}{(d-1)!} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{(b_1) (s) (a)} \frac{1}{(d-1)!} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array}$$

as $k_i < d$, we now have enough connections to apply (b_2) . Doing so and undoing the transformations on the left part, we get:

$$\text{ZW}_d \vdash \frac{1}{(d-1)!} \begin{array}{c} \dots \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \stackrel{=}{2} \frac{1}{(d-1)!} \begin{array}{c} \dots \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} = \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array} \begin{array}{c} \bullet \\ \text{---} \\ \circ \end{array}$$

Proof (Proof of Proposition 2). First notice that:

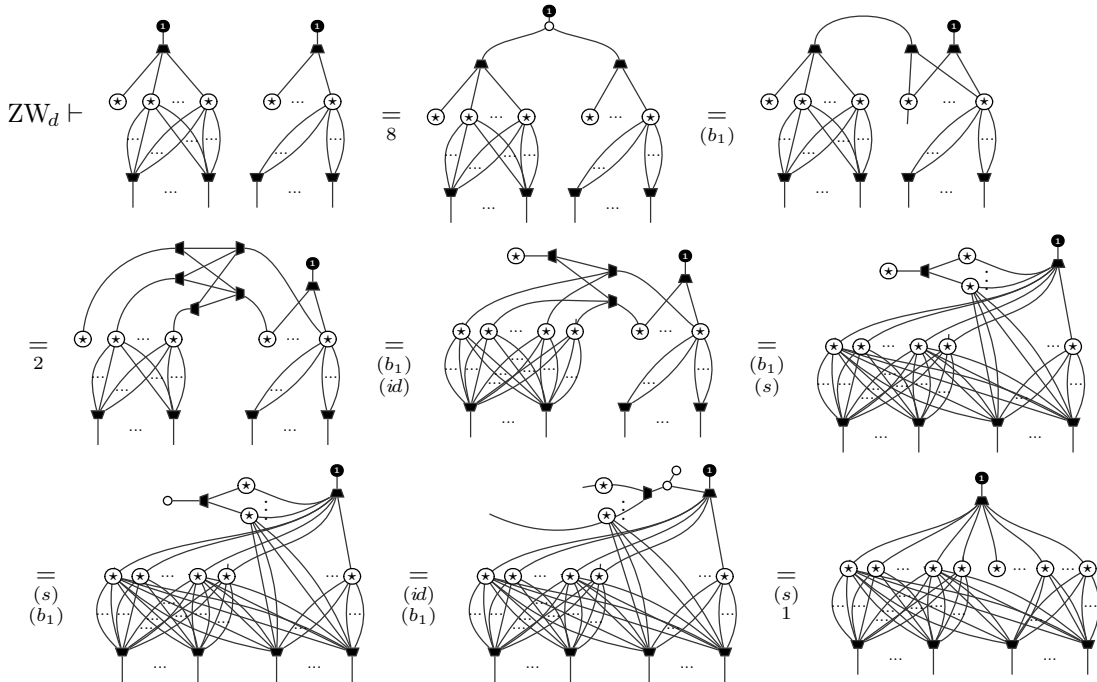
$$\begin{aligned} \left[\begin{array}{c} \bullet \\ \swarrow \quad \downarrow \quad \searrow \\ \dots \quad \dots \end{array} \right] &= |1, 0, \dots, 0\rangle + |0, 1, \dots, 0\rangle + \dots + |0, 0, \dots, 1\rangle \\ &= \left[\begin{array}{c} \bullet \\ | \\ \bullet \end{array} \right] + \left[\begin{array}{c} \bullet \\ | \\ \bullet \end{array} \right] + \dots + \left[\begin{array}{c} \bullet \\ | \\ \bullet \end{array} \right] \end{aligned}$$

In a given term, each $|0\rangle$ will merely cancel the Z-spider connected to it, and bring no contribution to the resulting state. Every contribution is brought by the $|1\rangle$. The number of parallel edges dictates to which basis state the contribution will go to. Consider for instance the i th term:

$$\begin{aligned} \left[\begin{array}{c} \bullet \\ \swarrow \quad \downarrow \quad \searrow \\ r_0 \quad r_1 \quad \dots \quad \frac{r_i}{\sqrt{x_1^i! \dots x_n^i!}} \quad \dots \\ \swarrow \quad \downarrow \quad \searrow \\ \dots \end{array} \right] &= \left[\begin{array}{c} \bullet \\ \swarrow \quad \downarrow \quad \searrow \\ \frac{r_i}{\sqrt{x_1^i! \dots x_n^i!}} \quad \dots \\ \swarrow \quad \downarrow \quad \searrow \\ \dots \end{array} \right] = \left[\begin{array}{c} \bullet \\ \swarrow \quad \downarrow \quad \searrow \\ \frac{r_i}{\sqrt{x_1^i! \dots x_n^i!}} \quad \dots \\ \swarrow \quad \downarrow \quad \searrow \\ \dots \end{array} \right] \\ &= \frac{r_i}{\sqrt{x_1^i! \dots x_n^i!}} \left[\begin{array}{c} \bullet \\ \swarrow \quad \downarrow \quad \searrow \\ \bullet \quad \bullet \quad \dots \quad \bullet \\ \swarrow \quad \downarrow \quad \searrow \\ \dots \end{array} \right] = \frac{r_i}{\sqrt{x_1^i! \dots x_n^i!}} \left[\begin{array}{c} \bullet \\ | \\ \bullet \end{array} \right] = r_i |x_1^i, \dots, x_n^i\rangle \end{aligned}$$

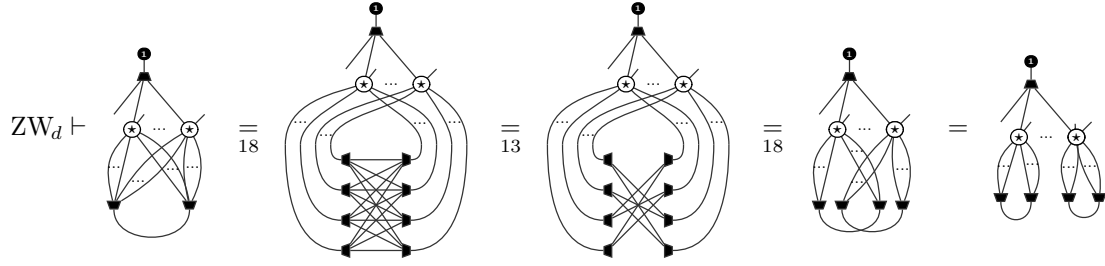
We do indeed recover the i th term in $|\psi\rangle$. We have something similar for all terms, so in the end $[\mathcal{N}(|\psi\rangle)] = |\psi\rangle$.

Proof (Proof of Proposition 3). Putting two diagrams in normal form side by side, we get:

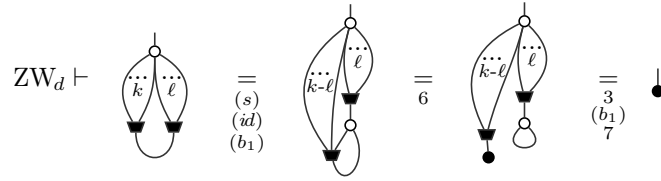


The diagram obtained at the end of this process is indeed in normal form.

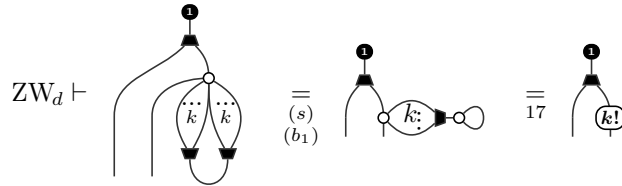
Proof (Proof of Proposition 4). First, we can show that the cup can “distribute” to each white node in the normal form:



We can then treat each white node independently. If a white node had k connections with the first output, and ℓ with the second one, with $k \neq \ell$, then it can simply be removed as:

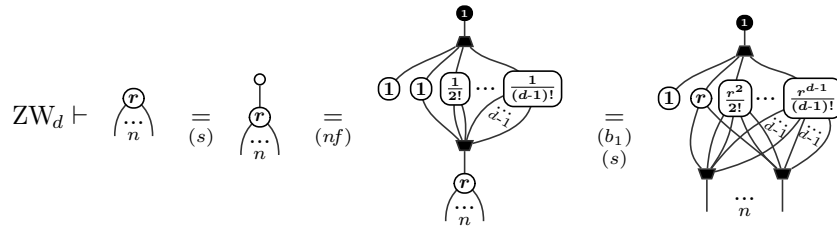


This \bullet then copies through the white node (with (b_1)) and gets absorbed by the top and bottom W-nodes of the normal form (by (id)). If $k = \ell$, however the “cup gadget” disappears and the white node gets a new parameter, as:



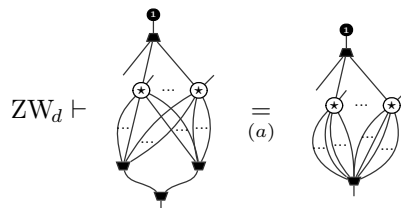
After the last two simplifications, it is possible that two white nodes end up with exactly the same connections. It is then possible to merge them (performing the sum of their parameters), using Lemma 5. After doing all these simplifications, the diagram is again in normal form.

Proof (Proof of Proposition 5). First, if $n > 0$:



If $n = 0$, then (r) can be obtained by compositions: $(r) = (r) \circ$. It can hence be turned in normal form by Propositions 3 and 4.

Proof (Proof of Proposition 6). By simply using (a), we get:



then, if a white node ends up with more than $d - 1$ connections with the resulting W-node, we can remove it by (h) (again the \downarrow generated by the rule copies through the white node and gets absorbed by the top and bottom W-nodes of the normal form, and white nodes with the same connections can be merged with Lemma 5). If two nodes end up with the same connections, they can be merged using Lemma 5. We end up with a diagram in normal form.

Proof (Proof of Corollary 2). Thanks to:

$$\text{ZW}_d \vdash \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array}$$

the binary W-node can be seen as applying $\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array}$ to 2 outputs of the normal form of $\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array}$ (notice that the diagram obtained by swapping 2 outputs of a normal form is directly in normal form). Thanks to 6, the obtained diagram can be put in normal form. This can obviously be generalised to the n -ary W-node thanks to (a).

Proof (Proof of Proposition 7). The cases where $k = 0$ and $k = 1$ are fairly trivial:

$$\text{ZW}_d \vdash \begin{array}{c} \bullet \\ | \\ \text{---} \end{array} \stackrel{\substack{(cp) \\ (id)}}{=} \begin{array}{c} \bullet \\ | \\ \text{---} \end{array} \quad \text{and} \quad \text{ZW}_d \vdash \begin{array}{c} \bullet \\ | \\ \text{---} \end{array} \stackrel{1}{=} \begin{array}{c} \bullet \\ | \\ \text{---} \end{array}$$

When $k > 1$, it can be obtained by composition, thanks to $\begin{array}{c} \bullet \\ | \\ \text{---} \end{array} \stackrel{7}{=} \begin{array}{c} \bullet \\ \vdots \\ \text{---} \end{array}$, hence by Propositions 3 and 6, it can be put in normal form.

B Alternative Presentation

Equation (nf) is certainly the least elegant of the equational theory, as its justification is simply: “we need it for the completeness proof”. Part of the difficulty in breaking it down to simpler ones is the fact that a single parameter describes the behaviour of the unary Z-spider on the whole d -dimensional space. If we instead had a list of parameters as in [30, 35], we could replace (nf) by a single simple equation, at the cost of a more complicated handling of the parameters, especially in Equation (+). We provide here a presentation in such a system.

We redefine the Z-spider as having $\vec{r} := (r_1, \dots, r_{d-1}) \in \mathbb{C}^{d-1}$ as parameter, with interpretation:

$$\left[\left[\begin{array}{c} \dots \\ \text{---} \\ \vec{r} \\ \text{---} \\ \dots \\ m \end{array} \right] \right] = |0^m\rangle\langle 0^n| + \sum_{k=1}^{d-1} r_k \sqrt{k!}^{n+m-2} |k^m\rangle\langle k^n|.$$

When a single parameter is used, we want to get the previous definition back, so we denote:

$$\mathbf{r} := (r, r^2, \dots, r^{d-1})$$

The two main operations that will be done on the lists of complex numbers are the following:

▷ The Hadamard product, or term-by-term product:

$$(r_1, \dots, r_{d-1}) \odot (s_1, \dots, s_{d-1}) := (r_1 s_1, \dots, r_{d-1} s_{d-1})$$

▷ A discrete binomial convolution:

$$(r_1, \dots, r_{d-1}) \star (s_1, \dots, s_{d-1}) := \left(r_1 + s_1, \dots, \sum_{i=0}^k \binom{k}{i} r_i s_{k-i}, \dots, \sum_{i=0}^{d-1} \binom{d-1}{i} r_i s_{d-1-i} \right)$$

taking $r_0 = s_0 := 1$.

Notice that in particular:

$$(r, r^2, \dots, r^{d-1}) \star (s, s^2, \dots, s^{d-1}) = ((r+s), (r+s)^2, \dots, (r+s)^{d-1}).$$

These two operations redefine the effects of Equations (s) and (+) on the parameters of the Z-spiders. The rest of the equational theory remains the same, except (nf) can be replaced by (e₁). The obtained equational theory is given in Figure 4. Equation (e₁) gives the normal form of a unary Z-spider, but with the simplest non-trivial support. It is then possible to derive Equation (nf) by decomposing arbitrary lists of parameters into convolutions of this simple one.

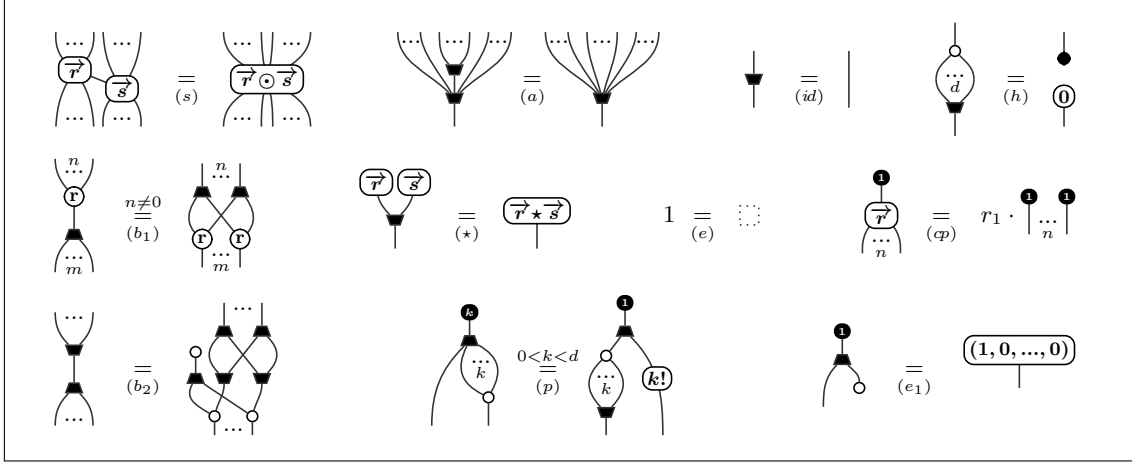


Fig. 4: Alternative equational theory ZW'_d for the qudit ZW -calculus.

To prove that the equational theory is complete, it suffices to prove that ZW_d is derivable with ZW'_d . Since all equations of ZW_d but (nf) are either exactly in ZW'_d or are special cases of equations of ZW'_d , it suffices to show that $ZW'_d \vdash (nf)$. Notice also that all lemmas and propositions derivable in ZW_d remain derivable in ZW'_d if they do not use (nf).

We denote the elementary lists $\vec{e}_k := (\overbrace{0, \dots, 0}^{k-1}, 1, 0, \dots)$. These can be shown to be decomposable as follows:

Lemma 19.

$$ZW'_d \vdash \begin{array}{c} \vec{e}_k \\ | \\ \circlearrowleft \end{array} = \begin{array}{c} \vec{e}_1 \quad \vec{e}_1 \quad \dots \quad \vec{e}_1 \\ | \quad | \quad \dots \quad | \\ 1 \quad e^{\frac{2i\pi}{k}} \quad \dots \quad e^{\frac{2i(k-1)\pi}{k}} \\ | \quad | \quad \dots \quad | \\ \circlearrowleft \quad \circlearrowleft \quad \dots \quad \circlearrowleft \\ | \\ \circlearrowleft \\ | \\ x_k \end{array}$$

where $x_k = (k!)^{\frac{1}{k}}$ if k is odd and $x_k = (k!)^{\frac{1}{k}} e^{\frac{i\pi}{k}}$ if k is even.

Proof. Proving this equality is more a matter of proving a non-trivial equation involving our convolution (\star) than a diagrammatic one. The operation that is performed is essentially:

$$r_0 \vec{e}_1 \star \dots \star r_{k-1} \vec{e}_1 = \left(1! \sum_i r_i, 2! \sum_{i_1, i_2} r_{i_1} r_{i_2}, 3! \sum_{i_1, i_2, i_3} r_{i_1} r_{i_2} r_{i_3}, \dots, k! r_0 \dots r_{k-1}, 0, \dots, 0 \right)$$

where we are looking for r_i s such that the above is equal to \vec{e}_k . Taking the r_i s to be the k th roots of unity makes all $k-1$ first terms equal to 0, and the k th equal to $(-1)^{k+1} k!$. The $(k!)^{\frac{1}{k}}$ and $(k!)^{\frac{1}{k}} e^{\frac{i\pi}{k}}$ then simply correct for the $(-1)^{k+1} k!$ that is left.

We can then use those to decompose arbitrary unary Z-spiders:

Lemma 20. *For any $\vec{r} \in \mathbb{C}^{d-1}$, there exist s_1, \dots, s_{d-1} such that:*

$$ZW'_d \vdash \begin{array}{c} \boxed{\vec{r}} \\ | \end{array} = \begin{array}{c} \boxed{\vec{e}_1} \quad \boxed{\vec{e}_2} \quad \dots \quad \boxed{\vec{e}_{d-1}} \\ \boxed{s_1} \quad \boxed{s_2} \quad \dots \quad \boxed{s_{d-1}} \\ \downarrow \quad \downarrow \quad \dots \quad \downarrow \\ \text{Spider} \end{array}$$

Proof. Finding such s_i s are just a matter of solving a linear system of equations in variables $s_1, s_2^2, \dots, s_{d-1}^{d-1}$.

We can finally use the above two lemmas to derive the normal form of arbitrary unary Z-spiders:

Proposition 8. *For any $\vec{r} \in \mathbb{C}^{d-1}$, the unary Z-spider with parameter $\vec{r}: \begin{array}{c} \boxed{\vec{r}} \\ | \end{array}$ can be put in normal form using ZW'_d .*

Proof. First of all, notice that tensor products of diagrams in normal form can be put in normal form, as Proposition 3 does not use (nf) at all. Same goes for Proposition 6 which states that the diagram obtained by applying $\begin{array}{c} \downarrow \\ \text{Spider} \end{array}$ to outputs of a normal form can be put in normal form. We can hence easily show that $\begin{array}{c} \boxed{\vec{e}_k} \\ | \end{array}$ can be put in normal form:

$$\begin{array}{l} ZW'_d \vdash \begin{array}{c} \boxed{\vec{e}_k} \\ | \end{array} =_{19} \begin{array}{c} \boxed{\vec{e}_1} \quad \boxed{\vec{e}_1} \quad \dots \quad \boxed{\vec{e}_1} \\ \textcircled{1} \quad e \quad \frac{2i\pi}{k} \quad e \quad \frac{2i(k-1)\pi}{k} \\ \downarrow \quad \downarrow \quad \dots \quad \downarrow \\ \text{Spider} \\ \downarrow \\ \boxed{x_k} \end{array} =_{(b_1)} \begin{array}{c} \boxed{\vec{e}_1} \quad \boxed{\vec{e}_1} \quad \dots \quad \boxed{\vec{e}_1} \\ \boxed{x_k} \quad \boxed{x_k e \frac{2i\pi}{k}} \quad \dots \quad \boxed{x_k e \frac{2i(k-1)\pi}{k}} \\ \downarrow \quad \downarrow \quad \dots \quad \downarrow \\ \text{Spider} \end{array} =_{(e_1)} \begin{array}{c} \textcircled{1} \quad \textcircled{1} \quad \dots \quad \textcircled{1} \\ \downarrow \quad \downarrow \quad \dots \quad \downarrow \\ \text{Spider} \\ \downarrow \\ \boxed{x_k} \quad \boxed{x_k e \frac{2i\pi}{k}} \quad \dots \quad \boxed{x_k e \frac{2i(k-1)\pi}{k}} \end{array} \\ = \begin{array}{c} \text{NF} \quad \text{NF} \quad \dots \quad \text{NF} \\ \downarrow \quad \downarrow \quad \dots \quad \downarrow \\ \text{Spider} \end{array} =_3 \begin{array}{c} \text{NF} \\ \downarrow \\ \text{Spider} \end{array} =_6 \begin{array}{c} \text{NF} \\ | \end{array} \end{array}$$

where the “NF” boxes represent diagrams in normal form, the exact shape of which does not matter for the proof, as they are uniquely defined from the semantics of the diagram they correspond to.

Similarly, we can further show that any $\begin{array}{c} \boxed{\vec{r}} \\ | \end{array}$ can be put in normal form:

$$\begin{array}{l} ZW'_d \vdash \begin{array}{c} \boxed{\vec{r}} \\ | \end{array} =_{20} \begin{array}{c} \boxed{\vec{e}_1} \quad \boxed{\vec{e}_2} \quad \dots \quad \boxed{\vec{e}_{d-1}} \\ \boxed{s_1} \quad \boxed{s_2} \quad \dots \quad \boxed{s_{d-1}} \\ \downarrow \quad \downarrow \quad \dots \quad \downarrow \\ \text{Spider} \end{array} = \begin{array}{c} \text{NF} \quad \text{NF} \quad \dots \quad \text{NF} \\ \boxed{s_1} \quad \boxed{s_2} \quad \dots \quad \boxed{s_{d-1}} \\ \downarrow \quad \downarrow \quad \dots \quad \downarrow \\ \text{Spider} \end{array} =_{(b_1)} \begin{array}{c} \text{NF} \quad \text{NF} \quad \dots \quad \text{NF} \\ \downarrow \quad \downarrow \quad \dots \quad \downarrow \\ \text{Spider} \end{array} \\ =_3 \begin{array}{c} \text{NF} \\ \downarrow \\ \text{Spider} \end{array} =_6 \begin{array}{c} \text{NF} \\ | \end{array} \end{array}$$

Theorem 4 (Completeness). *The language is complete with ZW'_d : for any two diagrams D_1 and D_2 of the ZW_d -calculus:*

$$[[D_1]] = [[D_2]] \iff ZW'_d \vdash D_1 = D_2$$

Proof. By Proposition 8, we can derive the normal form of $\begin{array}{c} \boxed{\vec{r}} \\ | \end{array}$ by taking $\vec{r} \leftarrow \mathbf{1}$. In other words $ZW'_d \vdash (nf)$. As a consequence $ZW'_d \vdash ZW_d$, so by Theorem 2, the equational theory ZW'_d makes the language complete.

C Lemmas and Proofs for FdHilb

In this section, for ease of notation, we will omit the box over the capacity annotations.

Lemma 21.

$$\text{ZW}_f \vdash a \circlearrowleft = a \mid$$

Proof.

$$\text{ZW}_f \vdash a \circlearrowleft \stackrel{(s)}{=} \begin{array}{c} a \\ \circlearrowleft \\ \circlearrowleft \end{array} \stackrel{(id)}{=} \begin{array}{c} a \\ \circlearrowleft \\ \blacktriangledown \\ a \end{array} \stackrel{(a)}{=} \begin{array}{c} a \\ \circlearrowleft \\ \blacktriangledown \\ a \end{array} \stackrel{(b_1)}{=} \begin{array}{c} a \\ \circlearrowleft \\ \blacktriangledown \\ a \end{array} \stackrel{(a')}{=} \begin{array}{c} a \\ \circlearrowleft \\ \blacktriangledown \\ a \end{array} \stackrel{(a)}{=} \begin{array}{c} a \\ \circlearrowleft \\ \blacktriangledown \\ a \end{array} \stackrel{(id)}{=} a \mid$$

Lemma 22. *If $b \leq a$:*

$$\text{ZW}_f \vdash \begin{array}{c} a \\ \blacktriangledown \\ b \\ \blacktriangledown \\ a \end{array} \stackrel{(b_2)}{=} \begin{array}{c} a \\ \blacktriangledown \\ a \\ \blacktriangledown \\ a \end{array} \stackrel{(id)}{=} a \mid$$

Lemma 23.

$$\text{ZW}_f \vdash \begin{array}{c} \bullet \\ \blacktriangledown \\ 0 \\ \blacktriangledown \\ a \end{array} \stackrel{(a)}{=} \begin{array}{c} \blacktriangledown \\ a \end{array} = \begin{array}{c} \bullet \\ a \end{array}$$

Lemma 24. *Equation (xp) can be generalised to:*

$$\text{ZW}_f \vdash \begin{array}{c} \sum_{i=1}^n a_i \\ \bullet \\ \blacktriangledown \\ a_1 \dots a_n \end{array} = \begin{array}{c} \circlearrowleft \\ a_1 \dots a_n \end{array}$$

Proof.

$$\text{ZW}_f \vdash \begin{array}{c} \sum_{i=1}^n a_i \\ \bullet \\ \blacktriangledown \\ a_1 \dots a_n \end{array} \stackrel{(a)}{=} \begin{array}{c} \sum_{i=1}^n a_i \\ \bullet \\ \blacktriangledown \\ a_1 \dots a_{n-1} \end{array} \stackrel{(xp)}{=} \begin{array}{c} \sum_{i=1}^{n-1} a_i \\ \bullet \\ \blacktriangledown \\ a_1 \dots a_{n-1} \end{array} \circlearrowleft \stackrel{ind.}{=} \begin{array}{c} \circlearrowleft \\ a_1 \dots a_{n-1} \end{array} \circlearrowleft$$

Lemma 25. *Equation (a') can be generalised to:*

$$\text{ZW}_f \vdash \begin{array}{c} a_0 \dots a_n \\ \bullet \\ \blacktriangledown \\ b \\ \blacktriangledown \\ c \end{array} = \begin{array}{c} a_0 \dots a_n \\ \bullet \\ \blacktriangledown \\ b \\ \blacktriangledown \\ c \end{array}$$

Proof. By induction on $n + 1$ the number of wires labelled a_0, \dots, a_n . When $n + 1 = 2$, the case is directly dealt with rule (a'). Else, assuming the result is proven for n :

$$\text{ZW}_f \vdash \begin{array}{c} a_0 \dots a_n \\ \bullet \\ \blacktriangledown \\ b \\ \blacktriangledown \\ c \end{array} \stackrel{(a)}{=} \begin{array}{c} a_0 \ a_1 \dots a_n \\ \bullet \\ \blacktriangledown \\ b \\ \blacktriangledown \\ c \end{array} \stackrel{(a')}{=} \begin{array}{c} a_0 \ a_1 \dots a_n \\ \bullet \\ \blacktriangledown \\ b \\ \blacktriangledown \\ c \end{array} \stackrel{(b_1)}{=} \begin{array}{c} a_0 \ a_1 \dots a_n \\ \bullet \\ \blacktriangledown \\ b \\ \blacktriangledown \\ c \end{array} \stackrel{(s)}{=} \begin{array}{c} a_0 \ a_1 \dots a_n \\ \bullet \\ \blacktriangledown \\ b \\ \blacktriangledown \\ c \end{array} \stackrel{(b_1)}{=} \begin{array}{c} a_0 \ a_1 \dots a_n \\ \bullet \\ \blacktriangledown \\ b \\ \blacktriangledown \\ c \end{array} \stackrel{(s)}{=} \begin{array}{c} a_0 \ a_1 \dots a_n \\ \bullet \\ \blacktriangledown \\ b \\ \blacktriangledown \\ c \end{array} \stackrel{(21)}{=} \begin{array}{c} a_0 \ a_1 \dots a_n \\ \bullet \\ \blacktriangledown \\ b \\ \blacktriangledown \\ c \end{array}$$

Lemma 26.

$$\text{ZW}_f \vdash \begin{array}{c} a \quad a \\ \dots \\ \downarrow \\ a \\ \dots \\ a \quad a \end{array} = \begin{array}{c} a \quad a \\ \dots \\ \downarrow \\ a \quad a \\ \dots \\ a \quad a \end{array}$$

Proof.

$$\begin{array}{c} \text{ZW}_f \vdash \begin{array}{c} a \quad a \\ \dots \\ \downarrow \\ a \\ \dots \\ a \quad a \end{array} \end{array} \stackrel{22}{=} \begin{array}{c} a \quad a \\ \dots \\ \downarrow \\ a \quad a \\ \dots \\ a \quad a \end{array} \stackrel{25}{=} \begin{array}{c} a \quad a \\ \dots \\ \downarrow \\ a \quad a \\ \dots \\ a \quad a \end{array} \stackrel{(b_2)}{=} \begin{array}{c} a \quad a \\ \dots \\ \downarrow \\ a \quad a \\ \dots \\ a \quad a \end{array} \\ \stackrel{(b_1) \quad (a) \quad (s)}{=} \begin{array}{c} a \quad a \\ \dots \\ \downarrow \\ a \quad a \\ \dots \\ a \quad a \end{array} \stackrel{(b_1) \quad (s)}{=} \begin{array}{c} a \quad a \\ \dots \\ \downarrow \\ a \quad a \\ \dots \\ a \quad a \end{array} \stackrel{(b_1) \quad (a)}{=} \begin{array}{c} a \quad a \\ \dots \\ \downarrow \\ a \quad a \\ \dots \\ a \quad a \end{array}$$

Lemma 27.

$$\text{ZW}_f \vdash \begin{array}{c} \textcircled{0} \\ | \\ a \end{array} = \begin{array}{c} \bullet \\ | \\ a \end{array}$$

Proof.

$$\text{ZW}_f \vdash \begin{array}{c} \textcircled{0} \\ | \\ a \end{array} \stackrel{(b_2)}{=} \begin{array}{c} \bullet \\ | \\ \textcircled{0} \\ | \\ a \end{array} \stackrel{(h)}{=} \begin{array}{c} \bullet \\ | \\ \textcircled{0} \\ | \\ a \end{array} \stackrel{(b_1)}{=} \begin{array}{c} \bullet \\ | \\ \textcircled{0} \\ | \\ a \end{array} \stackrel{(a)}{=} \begin{array}{c} \bullet \\ | \\ a \end{array}$$

Lemma 28.

$$\text{ZW}_f \vdash \begin{array}{c} \bullet \\ | \\ \textcircled{r} \\ | \\ a \end{array} = \begin{array}{c} \bullet \\ | \\ \dots \\ | \\ a \end{array}$$

Proof. The case where there are no outputs is dealt with

$$\text{ZW}_f \vdash \begin{array}{c} \bullet \\ | \\ \textcircled{r} \\ | \\ a \end{array} \stackrel{27}{=} \begin{array}{c} \textcircled{0} \\ | \\ \textcircled{r} \\ | \\ a \end{array} \stackrel{(s)}{=} a \textcircled{0} \stackrel{(s)}{=} \begin{array}{c} \textcircled{0} \\ | \\ \textcircled{0} \\ | \\ a \end{array} \stackrel{27}{=} \begin{array}{c} \bullet \\ | \\ a \end{array} \stackrel{(b_2)}{=} 1$$

The general case is a direct application of (b₁) with m = 0.

Lemma 29.

$$\text{ZW}_f \vdash \begin{array}{c} \bullet \\ | \\ \textcircled{a} \\ | \\ a \end{array} = \begin{array}{c} \bullet \\ | \\ \textcircled{a} \\ | \\ a \end{array}$$

Proof.

$$\text{ZW}_f \vdash \begin{array}{c} \bullet \\ | \\ \textcircled{a} \\ | \\ a \end{array} \stackrel{(id) \quad 28}{=} \begin{array}{c} \bullet \\ | \\ \textcircled{a} \\ | \\ a \end{array} \stackrel{(p)}{=} \begin{array}{c} \bullet \\ | \\ \textcircled{a} \\ | \\ a \end{array} \stackrel{(id) \quad 28}{=} \begin{array}{c} \bullet \\ | \\ \textcircled{a} \\ | \\ a \end{array} \stackrel{(cp)}{=} \begin{array}{c} \bullet \\ | \\ \textcircled{a} \\ | \\ a \end{array}$$

Lemma 30. If $b \geq a \geq k \geq b_i$:

$$\text{ZW}_f \vdash \begin{array}{c} \bullet \\ | \\ \textcircled{a} \\ | \\ b \end{array} = \begin{array}{c} \bullet \\ | \\ b \end{array} \quad \text{and} \quad \text{ZW}_f \vdash \begin{array}{c} \bullet \\ | \\ \textcircled{a} \\ | \\ b_1 \quad b_n \end{array} = \begin{array}{c} \bullet \\ | \\ \textcircled{a} \\ | \\ b_1 \quad b_n \end{array}$$

Proof.

$$\begin{aligned}
 \text{ZW}_f \vdash \begin{array}{c} \bullet^a \\ \downarrow \\ b \end{array} & \stackrel{29}{=} \begin{array}{c} 1 \quad \dots \quad 1 \\ \uparrow \quad \dots \quad \uparrow \\ \bullet^k \\ \downarrow \\ a \\ \downarrow \\ b \end{array} \stackrel{(a')}{=} \begin{array}{c} 1 \quad \dots \quad 1 \\ \uparrow \quad \dots \quad \uparrow \\ \bullet^k \\ \downarrow \\ b \end{array} \stackrel{29}{=} \begin{array}{c} \bullet^k \\ \downarrow \\ b \end{array} \\
 \text{ZW}_f \vdash \begin{array}{c} \bullet^a \\ \downarrow \\ \begin{array}{c} \dots \\ b_1 \quad \dots \quad b_n \end{array} \end{array} & = \begin{array}{c} \bullet^k \\ \downarrow \\ a \\ \downarrow \\ \begin{array}{c} \dots \\ b_1 \quad \dots \quad b_n \end{array} \end{array} \stackrel{(b_2)}{=} \begin{array}{c} \bullet^k \\ \downarrow \\ \begin{array}{c} \dots \\ b_1 \quad \dots \quad b_n \end{array} \end{array} \stackrel{(id)}{=} \begin{array}{c} \bullet^k \\ \downarrow \\ \begin{array}{c} \dots \\ b_1 \quad \dots \quad b_n \end{array} \end{array}
 \end{aligned}$$

Lemma 31. *Let $a \geq b$. Then:*

$$\text{ZW}_f \vdash \begin{array}{c} \circ^a \\ \uparrow \\ b \end{array} = \begin{array}{c} \circ \\ \downarrow \\ b \end{array}$$

Proof. We work by induction on $a \geq b$. If $a = b$, the result is a direct application of Equation (id). Else:

$$\begin{aligned}
 \text{ZW}_f \vdash \begin{array}{c} \circ^a \\ \uparrow \\ b \end{array} & \stackrel{(nf)}{=} \begin{array}{c} \circ^{a-1} \\ \uparrow \\ \begin{array}{c} \circ \\ \downarrow \\ a \\ \downarrow \\ b \end{array} \end{array} \stackrel{(b_2)}{=} \begin{array}{c} \circ^{a-1} \\ \uparrow \\ \begin{array}{c} \circ \\ \downarrow \\ \begin{array}{c} \dots \\ b \end{array} \end{array} \end{array} \stackrel{ind. (id)}{=} \begin{array}{c} \circ^{a-1} \\ \uparrow \\ b \end{array} \stackrel{(a) (s)}{=} \begin{array}{c} \circ^{a-1} \\ \uparrow \\ \begin{array}{c} \circ \\ \downarrow \\ a \\ \downarrow \\ b \end{array} \end{array} \\
 & \stackrel{(h) 27}{=} \begin{array}{c} \circ^{a-1} \\ \uparrow \\ \begin{array}{c} \bullet \\ \downarrow \\ b \end{array} \end{array} \stackrel{(id) 28}{=} \begin{array}{c} \circ \\ \downarrow \\ b \end{array}
 \end{aligned}$$

Definition 5. *With $d \geq a$, we define the “ a -restricted Z-spider as:*

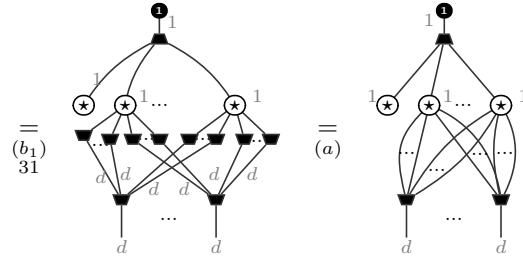
$$\begin{array}{c} \left(\begin{array}{c} n \\ \dots \\ a \\ \dots \\ m \end{array} \right) \\ d \end{array} \quad := \quad \begin{array}{c} \left(\begin{array}{c} n \\ \dots \\ a \\ \dots \\ m \end{array} \right) \\ d \end{array} \begin{array}{c} \circ^a \\ \uparrow \\ \bullet \end{array}$$

Lemma 32.

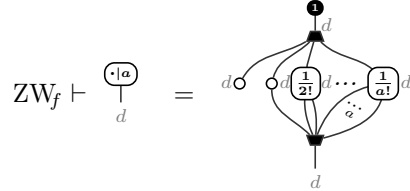
$$\text{ZW}_f \vdash \begin{array}{c} \bullet^d \\ \downarrow \\ \begin{array}{c} \circ^* \\ \downarrow \\ \dots \\ d \end{array} \end{array} = \begin{array}{c} \bullet^1 \\ \downarrow \\ \begin{array}{c} \circ^* \\ \downarrow \\ \dots \\ d \end{array} \end{array}$$

Proof.

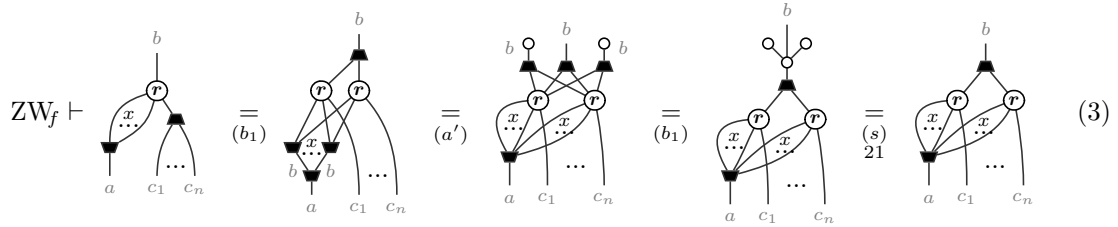
$$\begin{aligned}
 \text{ZW}_f \vdash \begin{array}{c} \bullet^d \\ \downarrow \\ \begin{array}{c} \circ^* \\ \downarrow \\ \dots \\ d \end{array} \end{array} & \stackrel{30}{=} \begin{array}{c} \bullet^1 \\ \downarrow \\ \begin{array}{c} \circ^* \\ \downarrow \\ \dots \\ d \end{array} \end{array} \stackrel{(b_2)}{=} \begin{array}{c} \bullet^1 \\ \downarrow \\ \begin{array}{c} \circ^* \\ \downarrow \\ \dots \\ d \end{array} \end{array}
 \end{aligned}$$



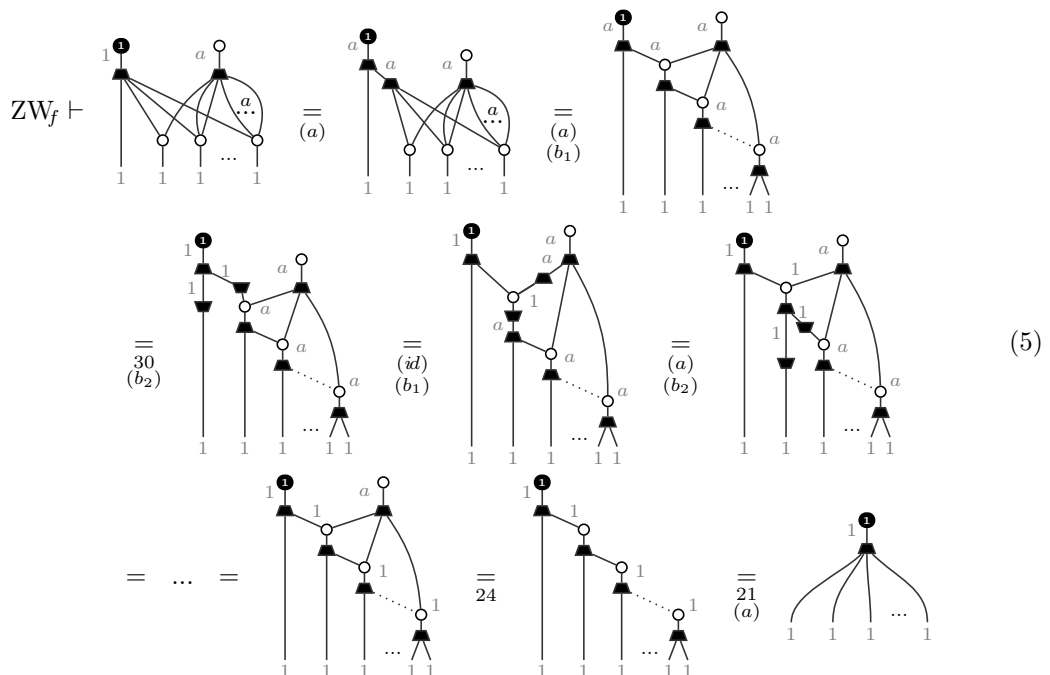
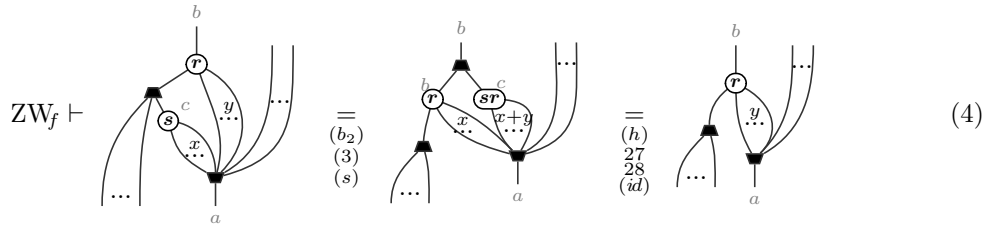
Lemma 33.



Proof. We need a few intermediary derivations first:



If $x + y > a$:



We then reason by induction on a . If $a = 0$, we have:

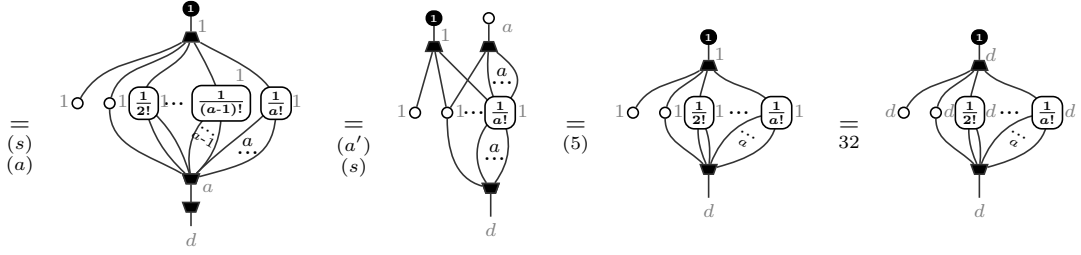
$$\begin{aligned}
 \text{ZW}_f \vdash \frac{\circlearrowleft \cdot 0}{d} &= \frac{\circlearrowleft 0}{d} \\
 &\stackrel{(id)}{=} \frac{\circlearrowleft 0}{d} \\
 &\stackrel{(h)}{=} \frac{\bullet 0}{d} \\
 &\stackrel{(28)}{=} \frac{\bullet 1}{d} \\
 &\stackrel{(q)}{=} \frac{\bullet 1}{d} \\
 &\stackrel{(id)}{=} \frac{\bullet 1}{d} \\
 &\stackrel{(a)}{=} \frac{\bullet 1}{d}
 \end{aligned}$$

The case $a = 1$ is obtained as:

$$\begin{aligned}
 \text{ZW}_f \vdash \frac{\circlearrowleft \cdot 1}{d} &= \frac{\circlearrowleft 1}{d} \\
 &\stackrel{(\epsilon_1)}{=} \frac{\bullet 1}{d}
 \end{aligned}$$

Otherwise, if we suppose the result is proven for 1 and for $a - 1$, then we can prove:

$$\begin{aligned}
 \text{ZW}_f \vdash \frac{\circlearrowleft \cdot a}{d} &= \frac{\circlearrowleft a}{d} \\
 &\stackrel{(nf)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{ind.}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{(s)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{ind. (5)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{30}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{(q)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{(b_1)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{30 (b_2)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{(3)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{(4)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{(s)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{(s)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{21}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{(s)}{=} \frac{\circlearrowleft a}{d} \\
 &\stackrel{21 (b_1)}{=} \frac{\circlearrowleft a}{d}
 \end{aligned}$$



Proposition 9. All equations of ZW_d are derivable from ZW_f , i.e.

$$\forall d \geq 2, ZW_d \vdash D_1 = D_2 \implies ZW_f \vdash \iota_d(D_1) = \iota_d(D_2)$$

Proof. First, notice that the result is obviously true for all axioms of compact-closed props. Then, it is enough to show the result for the equations in Figure 1, as all equations provable with ZW_d derive from them. Equations (s) , (id) , (a) , (b_1) , $(+)$, (cp) , (h) and (p) are directly translated to an equation of ZW_f through ι_d . Then, (b_2) is exactly Lemma 26, and (nf) is Lemma 33 with $a \leftarrow d$.

Proposition 10. Let $D \in \mathbf{ZW}_f$ be a diagram (state) with its largest capacity being $d - 1$. Then, there exists $D_d \in \mathbf{ZW}_d$ such that:

$$ZW_f \vdash \boxed{D} = \boxed{\iota_d(D_d)}$$

Proof. Let us write $\delta := d - 1$ for the maximum capacity in D . Using Lemma 22, we can “force” capacity δ on all wires:

$$ZW_f \vdash a \Big| \stackrel{22}{=} \begin{array}{c} a \\ \downarrow \\ \delta \\ \downarrow \\ a \end{array}$$

Every original generator in D is then surrounded by W-nodes (connected to them by their output). These can be turned into purely qudit systems as follows:

$$ZW_f \vdash \begin{array}{c} \delta \quad \delta \\ \downarrow \quad \downarrow \\ \text{---} \text{---} \\ \downarrow \quad \downarrow \\ a \text{---} r \\ \downarrow \quad \downarrow \\ \delta \quad \delta \end{array} \stackrel{(s)}{=} \begin{array}{c} \delta \quad \delta \\ \downarrow \quad \downarrow \\ \text{---} \text{---} \\ \downarrow \quad \downarrow \\ a \text{---} r \\ \downarrow \quad \downarrow \\ \delta \quad \delta \end{array} \stackrel{(b_1)}{=} \begin{array}{c} \delta \\ \downarrow \\ \text{---} \text{---} \\ \downarrow \\ \delta \end{array} \stackrel{(a)}{=} \begin{array}{c} \delta \\ \downarrow \\ \text{---} \text{---} \\ \downarrow \\ \delta \end{array}$$

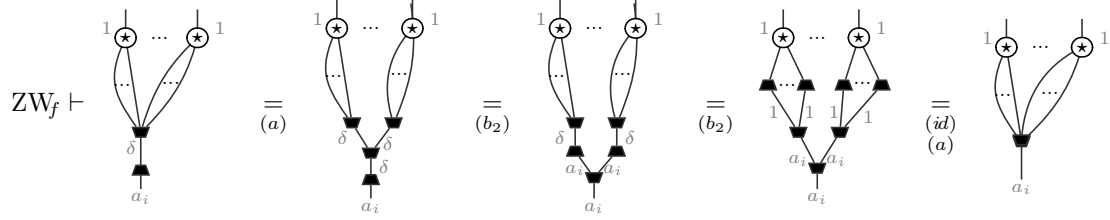
$$ZW_f \vdash \begin{array}{c} \delta \\ \downarrow \\ a \\ \downarrow \quad \downarrow \\ b \quad c \\ \downarrow \quad \downarrow \\ \delta \quad \delta \end{array} \stackrel{(a)}{=} \begin{array}{c} \delta \\ \downarrow \\ a \\ \downarrow \quad \downarrow \\ b \quad c \\ \downarrow \quad \downarrow \\ \delta \quad \delta \end{array} \stackrel{(b_2)}{=} \begin{array}{c} \delta \\ \downarrow \\ a \\ \downarrow \\ b \quad c \\ \downarrow \quad \downarrow \\ \delta \quad \delta \end{array} \stackrel{(a)}{=} \begin{array}{c} \delta \\ \downarrow \\ a \\ \downarrow \quad \downarrow \\ b \quad c \\ \downarrow \quad \downarrow \\ \delta \quad \delta \end{array} = \begin{array}{c} \delta \\ \downarrow \\ \text{---} \text{---} \\ \downarrow \\ \delta \end{array}$$

Finally, using (s) with 33, all “ a -restricted” Z-spiders $\delta \begin{pmatrix} n \\ r | a \\ m \end{pmatrix}$ can be turned into diagrams that only use capacity δ . Doing so for all generators, we consume all the W-nodes created at the beginning of the proof through Lemma 22, except the ones that are connected to the outputs of the diagram.

We hence end up in the form $\boxed{\iota_d(D_d)}$.

Proposition 11. Let D_d be a \mathbf{ZW}_d -state in (qudit) normal form. Then $\boxed{\iota_d(D_d)}$ can be put in (qufinite) normal form.

Proof. First, we may use Lemma 32 to turn all “internal” capacities in $\iota_d(D_d)$ into 1. $\iota_d(D_d)$ is hence technically in qufinite normal form. It remains to remove the W-nodes $\langle \delta \rangle \rightarrow \langle a_i \rangle$ at the bottom of the diagram. This can be done as follows, considering each output individually:



Finally, we may remove Z-spiders connected k times to output a_i when $k > a_i$ using Equation (h) followed by Lemma 28 and Equation (id).

Proof (Proof of Theorem 3). We can now show that any \mathbf{ZW}_f -state D can be put in normal form. First, use Proposition 10 to turn all but the outputs of the diagram into a qudit diagram. Using Proposition 9 and Theorem 2, we can turn the qudit-part of that diagram into qudit normal form. Finally, using Proposition 11, the whole diagram can be put in qufinite normal form.

If two \mathbf{ZW}_f -states D_1 and D_2 are semantically equivalent ($\llbracket D_1 \rrbracket = \llbracket D_2 \rrbracket$), then they can both be turned into normal form. By uniqueness of the normal form, the two diagrams are equal, i.e. $\llbracket D_1 \rrbracket = \llbracket D_2 \rrbracket \implies \mathbf{ZW}_f \vdash D_1 = D_2$.