Completeness of graphical languages for finite dimensional Hilbert spaces (Extended abstract)

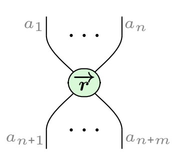
Razin A. Shaikh[1,2] Boldizsár Poór[1] Quanlong Wang[1]

[1]Quantinuum, 17 Beaumont Street, Oxford, OX1 2NA, United Kingdom
[2]University of Oxford, Oxford, United Kingdom

The ZXW completeness is based on the preprint (https://arxiv.org/abs/2309.13014) and the ZX completeness is based on a paper that is attached at the end of this abstract.

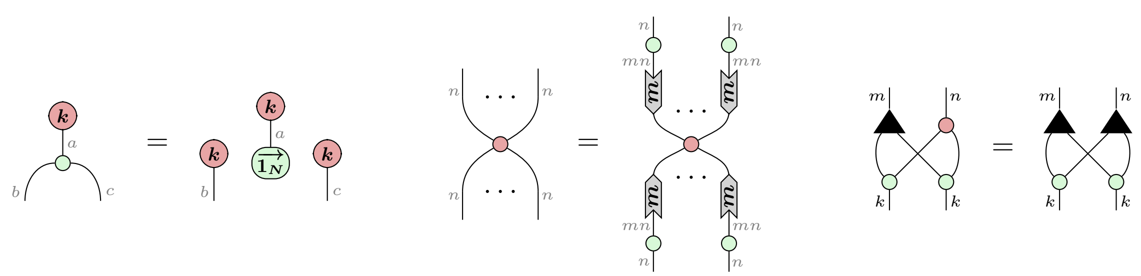
Finite-dimensional quantum theory serves as the theoretical foundation for quantum information and computation. Mathematically, it is formalized in the category **FHilb**, comprising all finite-dimensional Hilbert spaces and linear maps between them. The ZX calculus is a graphical language for reasoning about quantum computing and quantum theory. Recently, it was combined with another graphical language, the ZW calculus, to obtain the ZXW calculus. Both of these languages have been established for fragments of **FHilb** such as $\mathbf{FHilb}_2$ (qubits), and $\mathbf{FHilb}_d$ (qudits). As complete graphical language in fragments of **FHilb**, they incorporate set of axioms rich enough to derive any equation of the underlying formalism.

In this talk, we present the generalizations of the ZXW calculus and the ZX calculus that are universal and complete for **FHilb**. This is enabled by the generalized Z spiders whose legs can be of different dimensions:



$$\xmapsto{[\![\cdot]\!]} \quad \sum_{j=0}^{\min\{a_i\}_i-1} r_j \, |j,\cdots,j\rangle \langle j,\cdots,j| \,,$$

where $\overrightarrow{r} = (r_0, \cdots, r_{\min\{d_i\}_i-1})$. Some new axioms involving these mixed-dimensional spiders are shown below:



Theorem 1 (Completeness)**.** *The ZX and ZXW calculi are complete for* **FHilb***; that is, any equation derivable in multilinear algebra can be derived using the rewrite rules of the calculus.*

We prove this result for ZXW by demonstrating that any ZXW diagram can be rewritten into a unique normal form as: (1) all generators can be rewritten into their normal forms, (2) the tensor product of any two normal forms can be rewritten into a single normal form, and (3) a

Submitted to:
ACT 2024

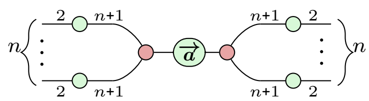
partially traced normal form can be rewritten into a normal form. On the other hand, we prove the completeness of the ZX calculus by direct translation to the finite-dimensional ZW calculus.

Building on the completeness result, we prove the following:

Theorem 2. *The category **ZX** and **ZXW** are monoidally equivalent to the category* **FHilb***.*

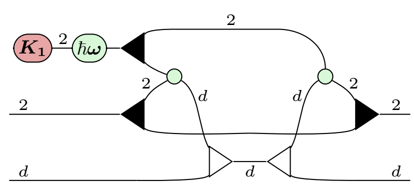
This result implies that the diagrammatic formalisms of **ZX** and **ZXW** have the same reasoning power as **FHilb**. Therefore, any computation of **FHilb** now can be done solely with diagrammatic rewriting.

With such a calculus, we can address problems in a wide range of domains based on finite-dimensional quantum physics. We show that the calculus can be suitable for studying angular momentum coupling and spin networks by presenting a compact diagram for irreducible representations of $SU(2)$. For example, the following diagram represents the symmetrizer of spin-$\frac{n}{2}$ for any $n \in \mathbb{N}$:

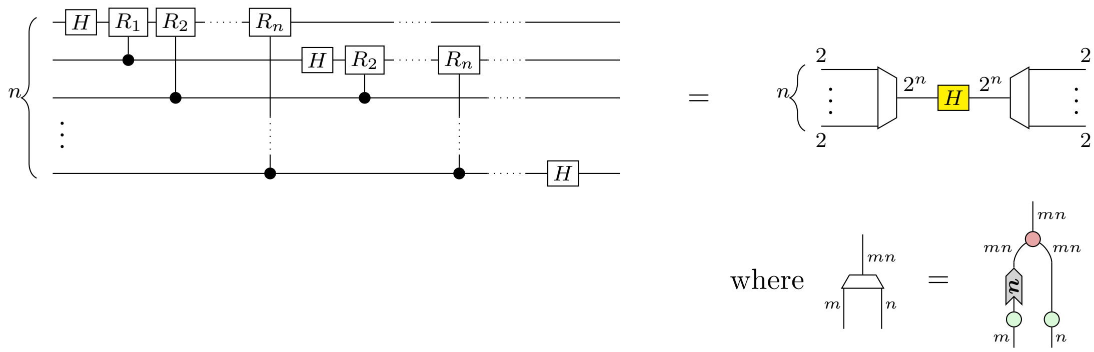


, where $\overrightarrow{a} = \left(\frac{1}{\binom{n}{1}}, \cdots, \frac{1}{\binom{n}{k}}, \cdots, \frac{1}{\binom{n}{n}} \right)$.

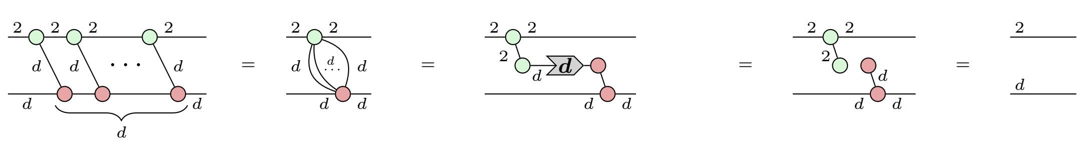
Furthermore, the calculus can be a valuable toolbox for reasoning about the Hamiltonians of interacting systems in quantum chemistry; we illustrate this by representing the Jaynes-Cummings model diagrammatically.



Finally, using the Quantum Fourier Transformation as an example, we demonstrate that the calculus can express complicated quantum computations concisely, and could potentially serve as a high-level language for quantum algorithms.



Quantum circuits with mixed-dimensional qudits have advantages in certain applications. The ZX calculus now allows us to leverage the strength of diagrammatic compilation and optimization techniques to the mixed-dimensional case. As an example, we can show that the qubit-controlled CNOT, when applied d times, is equivalent to the identity:



ZX-calculus is Complete for Finite-Dimensional Hilbert Spaces

Boldizsár Poór[1] Razin A. Shaikh[1,2] Quanlong Wang[1]

[1]Quantinuum, 17 Beaumont Street, Oxford, OX1 2NA, United Kingdom
[2]University of Oxford, United Kingdom

The ZX-calculus is a graphical language for reasoning about quantum computing and quantum information theory. As a complete graphical language, it incorporates a set of axioms rich enough to derive any equation of the underlying formalism. While completeness of the ZX-calculus has been established for qubits and the Clifford fragment of prime-dimensional qudits, universal completeness beyond two-level systems has remained unproven until now. In this paper, we present a proof establishing the completeness of finite-dimensional ZX-calculus, incorporating only the mixed-dimensional Z-spider and the qudit X-spider as generators. Our approach builds on the completeness of another graphical language, the finite-dimensional ZW-calculus, with direct translations between these two calculi. By proving its completeness, we lay a solid foundation for the ZX-calculus as a versatile tool not only for quantum computation but also for various fields within finite-dimensional quantum theory.

1 Introduction

The ZX-calculus [16, 17] is a graphical language for reasoning about quantum computation and quantum information theory. Thanks to its model-independent representation of quantum computation and set of intuitive graphical rewrite rules, the ZX-calculus has found applications across various domains of quantum technologies [56]. These domains include quantum error correction [25, 39, 36, 54, 24], quantum circuit optimization [31, 6, 40, 58], measurement-based quantum computing [32, 4, 43], fusion-based quantum computing [7, 45, 28], compilation [53], classical simulation [15, 11], and education [20, 19, 33].

While the ZX-calculus has primarily been studied for reasoning about qubit quantum computation, various extensions or modifications to the language enable it to address different aspects of quantum computation and theory. From the perspective of the ZX-calculus, there are two approaches to modify the language: changing the set of generators and/or giving them an alternative interpretation. Languages with a different set of generators include the ZW-calculus [21, 35], ZH-calculus [3, 42], and ZXW-calculus [52, 63]. Alternatively, a different interpretation may involve extending the language to higher-dimensional systems such as qutrits [60, 55, 57], quopits [9, 47, 22, 10], qudits [49, 48, 23], or the finite-dimensional setting [61, 62, 30].

While languages like the ZXW- and ZW-calculus enable complete reasoning for both qudits and finite-dimensional Hilbert spaces, defining the ZX-calculus with the same generality can open up avenues for interesting applications. This is because languages with different sets of generators exhibit distinct weaknesses and strengths. For instance, the ZW-calculus [18] provides valuable insights for studying multi-partite entanglement [18], and linear optical quantum computing [27, 29]. However, it is less effective in understanding circuit-based quantum protocols. Similarly, while the ZXW-calculus is a powerful analytical tool capable of expressing Hamiltonians [52, 29], performing differentiation and integration of ZX-diagrams [63], classical simulation [41], and interactions in quantum field theory [51], extracting circuits expressed by these methods remains a hard problem [26]. On the other hand, when working purely with the ZX-calculus, we can restrict ourselves to apply rewrites that keep the diagrams easily extractable [31]. Therefore, this approach has the potential to yield results directly applicable to quantum circuits.

When designing graphical languages, there are three crucial properties that we aim to prove to ensure the language can be effectively used. These properties are (1) *soundness*, which ensures that the rules of the calculus are correct, (2) *universality*, ensuring that the language can express any element of the underlying formalism, and (3) *completeness*, enabling the derivation of any equality within the underlying formalism. Of these, establishing a proof of completeness poses the most difficulty; nevertheless, it ensures that the language can be effectively used to graphically derive equalities and proofs.

Indeed, there is a rich history of results investigating the completeness and incompleteness of various calculi. While the qubit ZX-calculus was first formulated in 2007 [16], the crucial establishment of completeness for qubit quantum computing was not proved until 2017 [44, 38]. This unfolded in several stages, progressively increasing the fragment for which completeness had been proved. The first completeness proof was for the Clifford fragment [1], followed by a proof of incompleteness for the universal fragment [50]. Moving forward, completeness for single qubit Clifford+T quantum mechanics was established in [2], and universal completeness of the ZW-calculus was presented in [34]. Further study of the ZX-calculus revealed the necessity of the 'supplementarity' rule [46]. Transferring the completeness proof of the ZW-calculus [34, 35] led to a surge of completeness results. First, for the Clifford+T fragment in [37], and then for the universal fragment in [44, 38]. Further improvements to the qubit ZX-calculus focused on minimizing its set of axioms, first in [5] and then in [59]. This process was highly non-trivial, but now each rule is concise and intuitive.

Further to qubit calculi, the completeness of qudit calculi has also been extensively studied over the years. The first such completeness result was that of the stabilizer fragment for qutrit ZX-calculus [60]. The stabilizer fragment of the ZX-calculus for all odd prime dimensions was shown to be complete in [9], and its rule-set has since been further reduced in [47]. The first universal completeness of a graphical language beyond qubits was established in [48] for the arbitrary finite-dimensions of ZXW-calculus. Recently, the completeness of the qudit ZW-calculus has also been demonstrated in [30], with a significantly reduced set of axioms. Alongside the qudit ZW-calculus, the paper also introduces and proves the completeness of the finite-dimensional ZW-calculus [30]. This result came shortly after the establishment of the qufinite ZXW-calculus and its completeness for finite dimensional Hilbert spaces [62]. However, the universal completeness of a higher-dimensional ZX-calculus has remained unproven.

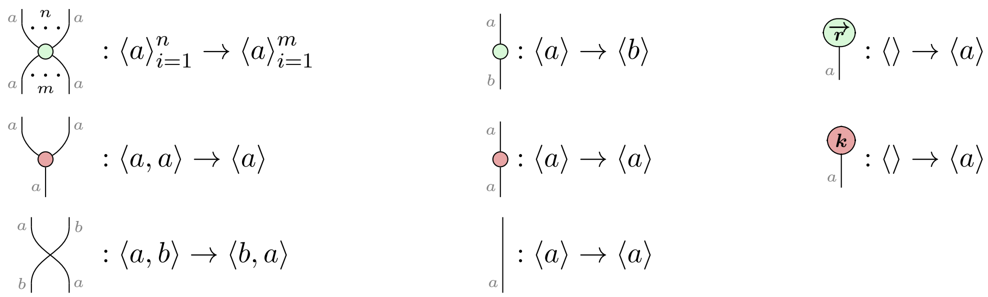
In this paper, we tackle the challenge of establishing the finite-dimensional ZX-calculus and proving its completeness. We begin by establishing the calculus generated by the mixed-dimensional Z-spider [62] and the flexsymmetric qudit X-spider. Next, we recapitulate the definition of the finite-dimensional ZW-calculus in accordance with [30]. We then move on to prove the main result of the paper — the completeness of the finite-dimensional ZX-calculus. It is achieved by translating the generators of our language to the finite-dimensional ZW-calculus [30] and proving the invertibility of this translation. This technique of translating between calculi to prove completeness has been employed before, first in [37], and later in [44, 38]. By establishing its completeness, we lay a solid foundation for the ZX-calculus as a versatile tool not only for quantum computation but also for various fields within finite-dimensional quantum theory.

2 Finite dimensional ZX calculus

In this section, we define our calculus starting with the generators, their interpretation, and lastly the axiomatization that we later show to be complete. Our calculus is constructed in order to satisfy the property of flexsymmetry, proposed in [12, 13], and allowing the recovery of the Only Connectivity Matters (OCM) meta-rule. OCM is a desirable feature when designing of a graphical language as it can both simplify the manipulation of diagrams and on paper and ease implementation in software.

2.1 Generators

We define the symmetric monoidal category $\mathbf{ZX_f}$ with objects as lists of dimensions $\langle d_i\rangle_{i=1}^n$ where $d_i \in \mathbb{N}$ for all $0 < i \leq n$, and morphisms generated by the following diagrams, for any $a, b, n, m \in N$, $0 \leq k < a$, and $\overrightarrow{r} \in \mathbb{C}^a$:

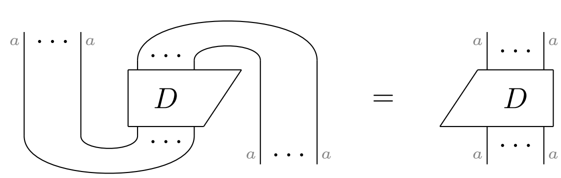


Diagrams in our framework can be composed in two ways, sequentially, by connecting input and output wires, and in parallel, by placing diagrams side-by-side.

We extend our language with the standard *qudit Z-spider* and the *mixed-dimensional Z-spider* [62], given by the following compositions, respectively:

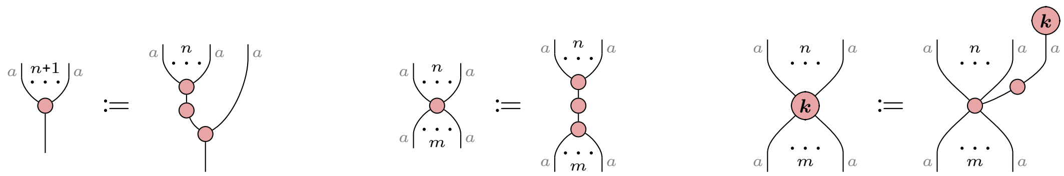
where $a = \min_{i=0}^{n+m} a_i$. Furthermore, the phase-free Z-spider with arbitrary legs can express both the cap and cup as follows:

Caps and cups then allow us to construct the transposition of any diagram:



In particular, we obtain the transposition of all the generators listed.

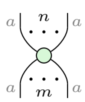
With these elements defined, we can define the *X-spider* inductively, for any $m, n \in \mathbb{N}$, as follows:

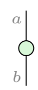


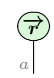
The requirement of the $1 \to 1$ red vertices (called *antipodes*) is the consequence of using the flexsymmetric X-spider in our calculus to recover the OCM meta-rule mentioned above.

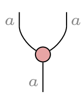
2.2 Interpretation

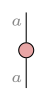
The standard interpretation of a diagram in $\mathbf{ZX_f}$ is a symmetric monoidal functor $[\![\cdot]\!] : \mathbf{ZX_f} \to \mathbf{FHilb}$ where **FHilb** is the category of finite-dimensional Hilbert spaces. On objects, it is defined as $[\![\langle a_i \rangle_{i=1}^n]\!] = \mathbb{C}^A$ where $A = \prod_{i=1}^n a_i$, and on morphisms, it is given as follows:

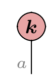
 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k=0}^{a-1} |k\rangle^{\otimes m} \langle k|^{\otimes n}$

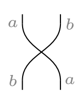
 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k=0}^{\min\{a,b\}-1} |k\rangle \langle k|$

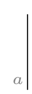
 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k=0}^{a-1} r_k |k\rangle$

 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k,\ell=0}^{a-1} |{-(k+\ell)} \bmod a\rangle \langle k, \ell|$

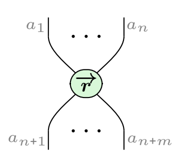
 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k=0}^{a-1} |a-k\rangle \langle k|$

 $\overset{[\![\cdot]\!]}{\longmapsto} |k\rangle$

 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k=0}^{a-1} \sum_{l=0}^{b-1} |\ell, k\rangle \langle k, \ell|$

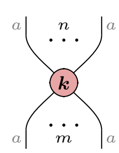
 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k=0}^{a-1} |k\rangle \langle k|$

By the functoriality of the standard interpretation, the general spiders have the following correspondence in **FHilb**:



$$\overset{[\![\cdot]\!]}{\longmapsto} \quad \sum_{j=0}^{\min\{a_i\}_i - 1} r_j |j, \cdots, j\rangle \langle j, \cdots, j| ,$$

where $\overrightarrow{r} = (r_0, \cdots, r_{\min\{d_i\}_i - 1})$, and



$$\overset{[\![\cdot]\!]}{\longmapsto} \quad \sum_{\substack{i_1+\cdots+i_m \\ +j_1+\cdots+j_n \\ \equiv k \ (\mathrm{mod}\ a)}} |i_1, \cdots, i_m\rangle \langle j_1, \cdots, j_n| ,$$

where $0 \le i_p, j_q < a$.

There are a couple of features in this interpretation that should be noted. First and foremost, the mixed-dimensional Z-spider can have legs of varying dimensions. In the qudit setting, a Z-spider behaves as the generalized Kronecker delta: it ensures that the same basis state is present on each of its legs. In our mixed-dimensional calculus, this behaviour is preserved by selecting the k-th standard basis on each leg for any k less than the minimal dimension. As such behaviour cannot be defined for the remaining basis states, we set their coefficients to zero. In other words, these basis states are not included in the sum.

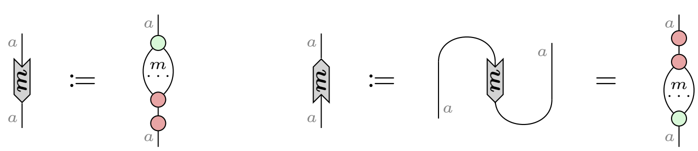
Second, all coefficients of our Z-spiders can be set to an arbitrary complex number. This is similar to the Z-box generator of [62, 48, 61], where the box notation is used for arbitrary complex coefficients, and the oval Z spider is used as a special case where all the coefficients are of the form $e^{i\alpha_j}$. As these exponential phases do not have a special role in this work, we do not distinguish between the two and simply label the oval Z-spider with any complex vector. A small difference between the coefficients of the Z-box from [48] and our Z-spider is that the former defines the first coefficient to be 1 ($a_0 = 1$). We relax this requirement as it provides more flexibility and cleaner rules.

Lastly, the interpretation of the X-spider matches that of [9, 47] but differs from [48, 62]. This is in order to make the X-spider flexsymmetric and thus recover the OCM meta-rule. This allows us to treat diagrams based solely on their connectivity and disregard relative positions. However, with this interpretation, X-spiders do not only sum basis states, but also map the resulting summation to its additive inverse based on the dimension. In particular, this means that the red $1 \to 1$ spider is not the identity, rather, it maps any input X basis state to its additive inverse. Further to these points, an additional advantage of the flexsymmetric interpretation is that k labelled X-spiders with input or output now exactly correspond to a computational basis costates/states:

This is different to the conventions in [48, 62], where the K_k state corresponds to $|d - k\rangle$.

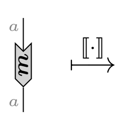
2.3 Notations

- A *multiplier* [8, 14, 9] labelled by m corresponds to a Z- and X-spider connected with m wires. Unlike in the qubit case, a green and a red spider can be connected with more than one wire. In fact, the Hopf law generalizes to d connections in dimension d, implying that the multiplier may be labelled modulo d.



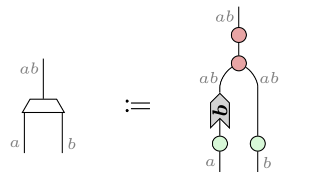
(MU)

Then, the interpretation of the multiplier is given as follows:



$$\sum_{i=0}^{a-1} |m \cdot i\rangle \langle i|$$

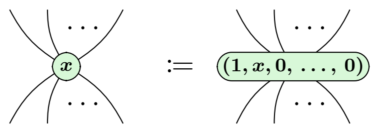
- We can define the *dimension splitter* of the qufinite ZX calculus [61] as follows:



$$\overset{[\![\cdot]\!]}{\longmapsto} \quad \sum_{i=0}^{a-1}\sum_{j=0}^{b-1} |i,j\rangle\langle ib+j| \qquad \text{(DD)}$$

  Note that this definition matches that of [62, Axiom (DD)], up to the extra antipode which is due to flexsymmetric X-spider.

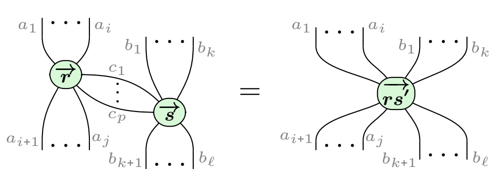
- Throughout this paper, we extensively use the vector $N = (0, 1, \cdots, a-1)$, where the dimension is a, and elementwise functions on this vector. For example, $\sqrt{N!}$ corresponds to the vector $(\sqrt{0!}, \sqrt{1!}, \cdots, \sqrt{(a-1)!})$ and r^N refers to $(r^0, r^1, \cdots, r^{a-1})$.
- When two vectors are multiplied or added in the parameter of a Z-spider, we refer to elementwise multiplication or addition of the vectors.
- When the first two elements of a Z-spider are 1 and x, we use the following shorthand:



  where $x \in \mathbb{C}$.

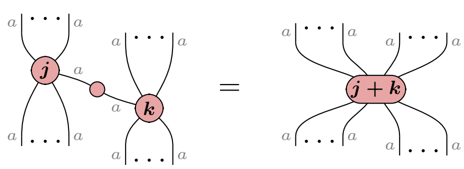
2.4 Axiomatization

In this section, we give a set of graphical rewrite rules to perform purely diagrammatic reasoning.

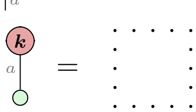


(S1)

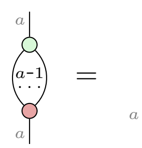
where $A = \{a_t\}_{t=0}^{j}$, $B = \{b_t\}_{t=0}^{\ell}$, $C = \{c_t\}_{t=0}^{p}$, $M = \min(A \cup B \cup C)$, $m = \min(A \cup C)$, $n = \min(B \cup C)$, $\overrightarrow{r} = (r_0, \ldots, r_{m-1})$, $\overrightarrow{s} = (s_0, \ldots, s_{n-1})$, and $\overrightarrow{rs}' = (r_0 s_0, \ldots, r_{M-1} s_{M-1})$.



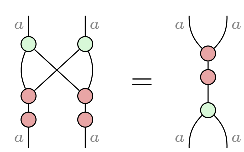
(S4)



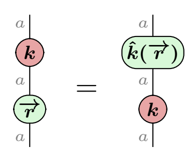
(Ept)



(Special)

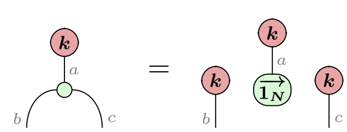


(B2)



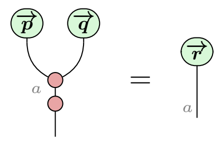
(K2)

where $\hat{k}(\overrightarrow{r}) = (r_{d-k}, \ldots, r_{1-k})$.



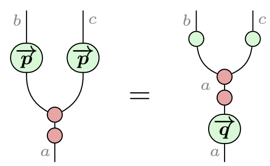
(K0)

where $N = \min\{a, b, c\}$.



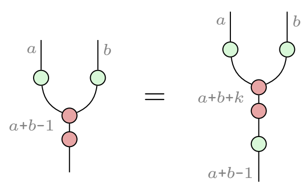
(PA)

where the k^{th} element of $\overrightarrow{r}$ is $r_k = \sum_{i=0}^{a-1} p_i q_{k\text{-}i \text{ (mod } a)}$.



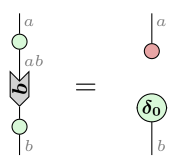
(PC)

where $\overrightarrow{q}$ is a vector such that for all $0 \leq i < b$ and $0 \leq j < c$ we have $p_i p_j = q_{i+j \text{ (mod } a)}$.



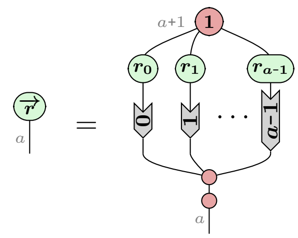
(DA)

where $k \in \mathbb{N}_0$.

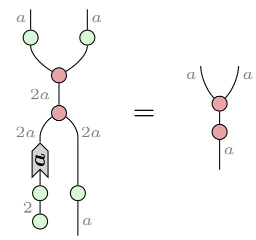


(HP)

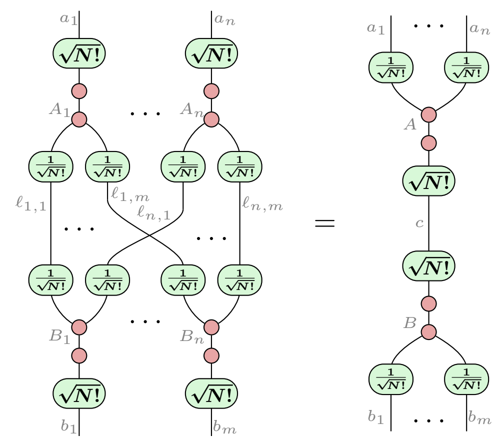
where $\delta_0 = (1, 0, \ldots, 0)$



(ZNF)



(XM)



(WW)

where $c \geq \min(\sum a_i, \sum b_i)$, $\ell_{ij} = \min(a_i, b_j)$, $A_i = \sum_k \ell_{i,k}$, and $B_i = \sum_k \ell_{k,i}$.

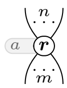
3 Finite dimensional ZW calculus [30]

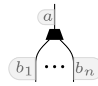
This section recapitulates the finite-dimensional ZW-calculus, in accordance with [27], discussing its generators with their interpretations and presenting the complete axiomatization of the calculus.

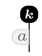
3.1 Generators

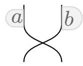
We define the symmetric monoidal category $\mathbf{ZW_f}$ with objects as lists of dimensions $\langle d_i \rangle_{i=1}^n$ where $d_i \in \mathbb{N}$ for all $0 < i \leq n$, and morphisms generated by the following diagrams, for any

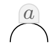
$a, b, b_j, n, m \in N$, $0 < j \leq n$, $0 < k \leq a$, and $r \in \mathbb{C}$:

 $: \langle a \rangle_{i=1}^{n} \to \langle a \rangle_{i=1}^{m}$

 $: \langle a \rangle \to \langle b_i \rangle_{i=1}^{n}$

 $: \langle\rangle \to \langle a \rangle$

 $: \langle a, b \rangle \to \langle b, a \rangle$

 $: \langle\rangle \to \langle a, a \rangle$

 $: \langle a, a \rangle \to \langle\rangle$

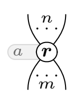
$r : \langle\rangle \to \langle\rangle$

 $: \langle a \rangle \to \langle a \rangle$

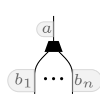
Compositions are given in the usual way.

3.2 Interpretation

The interpretation of a diagram in $\mathbf{ZW_f}$ is a symmetric monoidal functor $[\![\cdot]\!] : \mathbf{ZW_f} \to \mathbf{FHilb}$. On objects, it is defined as $[\![\langle a_i \rangle_{i=1}^{n}]\!] = \mathbb{C}^A$ where $A = \prod_{i=1}^{n} a_i$. The interpretation of the Z-spider is as follows, for any $r \in \mathbb{C}$:

 $\overset{[\![\cdot]\!]}{\longmapsto} \quad \sum_{k=0}^{a} r^k \sqrt{k!}^{\,n+m-2} \, |k^m\rangle\langle k^n|$

The W-node and its interpretation are given as follows:

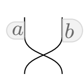
 $\overset{[\![\cdot]\!]}{\longmapsto} \quad \sum_{\substack{0 \leq k_i \leq b_i \\ k_1+...+k_n \leq a}} \sqrt{\binom{k_1+...+k_n}{k_1, ..., k_n}} \, |k_1, ..., k_n\rangle\langle k_1+...+k_n|$

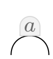
where $a \geq \max_{i=1}^{n} b_i$. The interpretation of the remaining generators are given as follows:

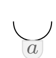
 $\overset{[\![\cdot]\!]}{\longmapsto} \begin{cases} \sqrt{k!}\,|k\rangle & \text{if } 0 < k \leq a \\ \vec{0} & \text{otherwise} \end{cases}$

$r \overset{[\![\cdot]\!]}{\longmapsto} r$

 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k=0}^{a} |k\rangle \langle k|$

 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k=0}^{a} \sum_{l=0}^{b} |\ell, k\rangle \langle k, \ell|$

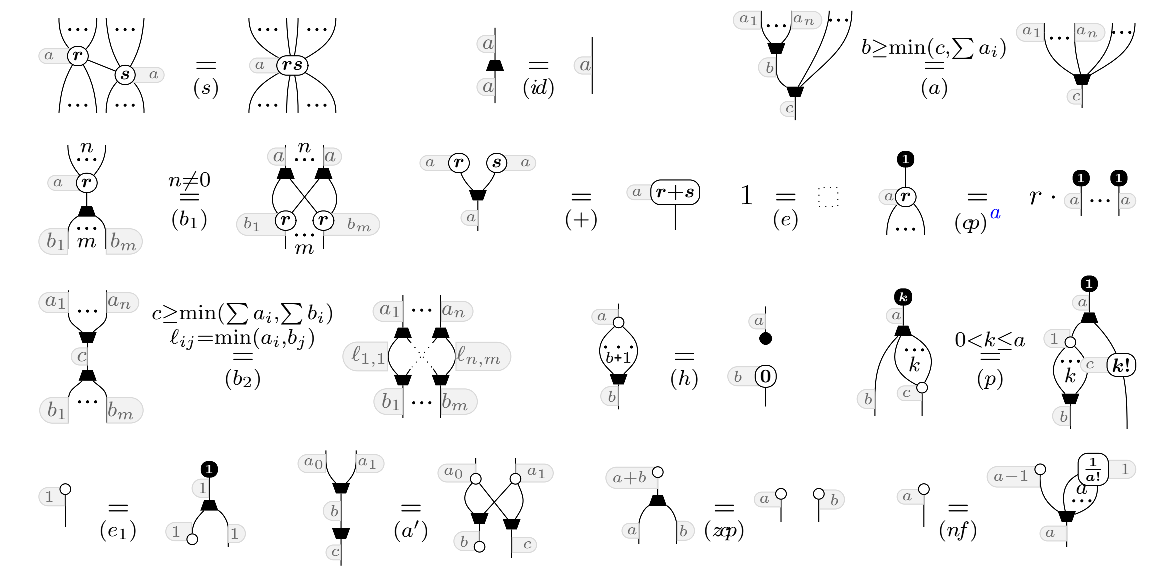
 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k=0}^{a} |k, k\rangle$

 $\overset{[\![\cdot]\!]}{\longmapsto} \sum_{k=0}^{a} \langle k, k|$

It is worth pointing out that wire dimensions in $\mathbf{ZX_f}$ and $\mathbf{ZW_f}$ do not match. In the ZX-calculus we use d to indicate that the wire carries a d-dimensional qudit while in case of the ZW-calculus it means a $(d+1)$-dimensional qudit.

3.3 Axioms

The equational theory $\mathbf{ZW_f}$ for the finite-dimensional ZW-calculus is given as follows:



[a] A missing r is added to the left hand side.

Proposition 1 (Completeness of ZW-calculus)**.** *For any two* $\mathbf{ZW_f}$*-diagrams* D_1 *and* D_2*,*

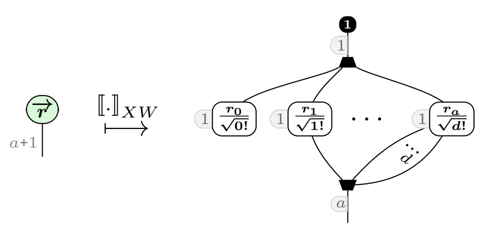
$$[\![D_1]\!] = [\![D_2]\!] \iff \mathbf{ZW_f} \vdash D_1 = D_2$$

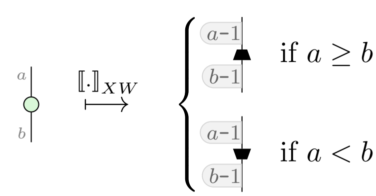
4 Completeness from translation

In this section, we prove the completeness of the finite-dimensional ZX-calculus. Our proof strategy bases on that outlined in [37, 44]. The idea of the proof is to define a translation from and to a complete language — the finite-dimensional ZW-calculus in our case. Then, if one can prove certain properties of these translation functors, such as soundness and invertability, then we obtain completeness.

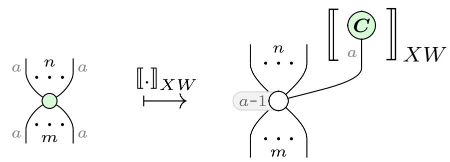
4.1 ZX-to-ZW translation

Here, we define the *ZX-to-ZW translation functor*, $[\![\cdot]\!]_{XW} : \mathbf{ZX_f} \to \mathbf{ZW_f}$, and show how it maps the generators of the finite-dimensional ZX-calculus. We first show how the generators defining a general mixed-dimensional Z-spider are translated. We can translate the Z spider with a single output and the embedding as follows:

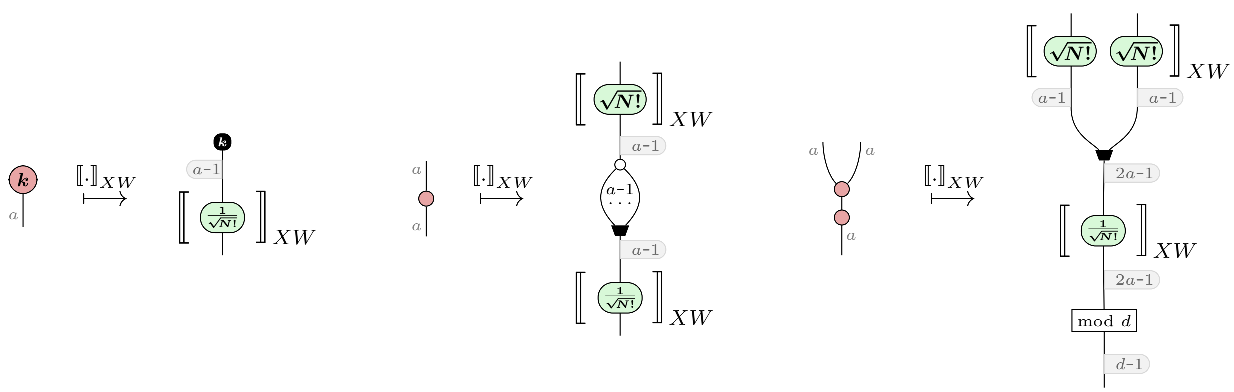




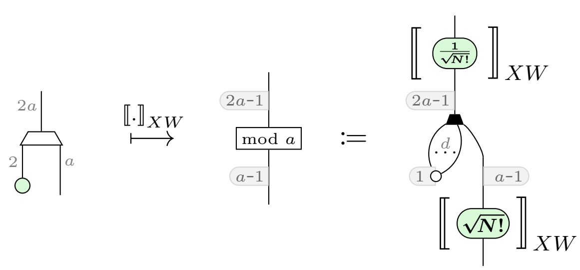
The former interpretation is based on the normal form of the qudit ZW calculus [30, Definition 3], and the embedding follows from comparing the interpretations. Since the Z-spiders of the two calculi match up to a $C = \left(\frac{1}{\sqrt{N!}}\right)^{m+n-1}$ vector parameter, the translation is not difficult:



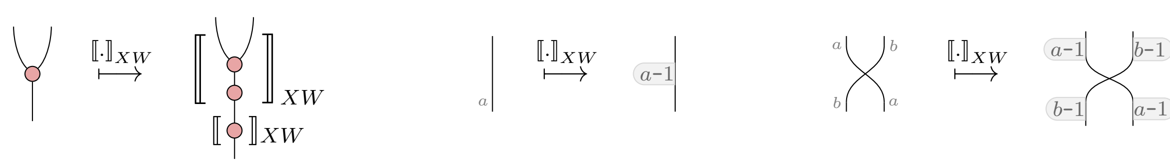
Now, we focus on expressing generators that include X-spiders. The translation of computational basis states and the antipode follow easily from the interpretations and Axiom (**??**). To express the X-spider in the finite dimensional ZW calculus, we first point out that the interpretation of the W- and (non-flexsymmetric) X-spider are—up to scalars—closely related. The only difference is that the W-spider results in sums of the inputs that are smaller than the output dimension ($|a+b\rangle\langle a,b|$ for $a,b \leq d$ and $a+b \leq d$) while the (non-flexsymmetric) X-spider sums elements modulo d ($|a+b \mod d\rangle\langle a,b|$ for $a,b \leq d$). Setting the output dimension of the W-spider to $2d$ and applying a modulo d gate afterwards ($|k \mod d\rangle\langle k|a+b\rangle\langle a,b|$ for $a,b \leq d$ and $a+b \leq 2d$, which is equivalent to $|a+b \mod d\rangle\langle a,b|$ for $a,b \leq d$), we can obtain the translation of the (non-flexsymmetric) X spider.



where the modulo d gadget is implemented with a ZW-diagram, but it is also expressible in the ZX-calculus:



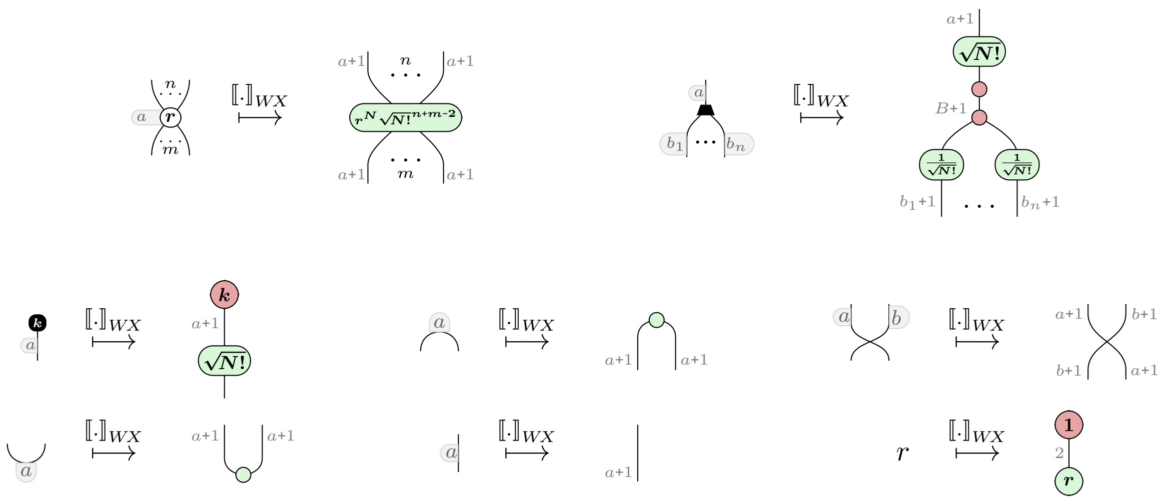
Given the above translations, mapping the remaining generators—the flexsymmetric X-spider, the identity, and the swap—can be tackled easily:



Lemma 1. *Suppose D is an arbitrary ZX-diagram.*
Then the above-defined ZX-to-ZW translation functor is sound, that is, $[\![[\![D]\!]_{XW}]\!] = [\![D]\!]$.

4.2 ZW-to-ZX translation

Here, we define the *ZW-to-ZX translation functor*, $[\![.]\!]_{WX} : \mathbf{ZW_f} \to \mathbf{ZX_f}$, and show how it maps the generators of the finite-dimensional ZW-calculus. For $B = \sum_{i=1}^{n} b_i$,



Lemma 2. *Suppose D is an arbitrary ZW-diagram.*
Then the above-defined ZW-to-ZX translation functor is sound, that is, $[\![[\![D]\!]_{WX}]\!] = [\![D]\!]$.

4.3 Proof

Lemma 3. *For an arbitrary ZX-diagram, D, we can prove* $[\![[\![D]\!]_{WX}]\!]_{XW} = D$.

Appendix A.2 is devoted to proving this by showing that the above lemma holds for all generators of $\mathbf{ZX_f}$.

Proposition 2. *If* $\mathbf{ZW_f} \vdash D_1 = D_2$ *then* $\mathbf{ZX_f} \vdash [\![D_1]\!]_{WX} = [\![D_2]\!]_{WX}$.

To prove this proposition, we need to show that all rewrite rules of $\mathbf{ZW_f}$ are derivable in $\mathbf{ZX_f}$. These proofs are the content of Appendix A.3.

Theorem 1 (Completeness)**.** *The finite-dimensional ZX-calculus is universally complete for finite-dimensional Hilbert spaces: For any ZX-diagrams D_1 and D_2, if* $[\![D_1]\!] = [\![D_2]\!]$, *then* $\mathbf{ZX_f} \vdash D_1 = D_2$.

Proof. Suppose $D_1, D_2 \in \mathbf{ZX_f}$ such that $[\![D_1]\!] = [\![D_2]\!]$. By Lemma 1, $[\![[\![D_1]\!]_{XW}]\!] = [\![D_1]\!] = [\![D_2]\!] = [\![[\![D_2]\!]_{XW}]\!]$. By the completeness of ZW-calculus, $\mathbf{ZW_f} \vdash [\![D_1]\!]_{XW} = [\![D_2]\!]_{XW}$. Now by Proposition 2, $\mathbf{ZX_f} \vdash [\![[\![D_1]\!]_{WX}]\!]_{XW} = [\![[\![D_2]\!]_{WX}]\!]_{XW}$. Finally, by Lemma 3, D1 = D2. □

Acknowledgements

We would like to Alexander Cowtan and Lia Yeh for their detailed feedback and several suggestions for improvement. RS is supported by the Clarendon Fund Scholarship.

References

[1] Miriam Backens (2014): *The ZX-calculus Is Complete for Stabilizer Quantum Mechanics.* *New Journal of Physics* 16(9), p. 093021, doi:10.1088/1367-2630/16/9/093021.

[2] Miriam Backens (2014): *The ZX-calculus Is Complete for the Single-Qubit Clifford+T Group. Electronic Proceedings in Theoretical Computer Science* 172, pp. 293–303, doi:10.4204/EPTCS.172.21.

[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*, *Electronic Proceedings in Theoretical Computer Science* 287, Open Publishing Association, Halifax, Canada, pp. 23–42, doi:10.4204/EPTCS.287.2.

[4] Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski & John van de Wetering (2021): *There and Back Again: A Circuit Extraction Tale.* *Quantum* 5, p. 421, doi:10.22331/q-2021-03-25-421.

[5] Miriam Backens, Simon Perdrix & Quanlong Wang (2020): *Towards a Minimal Stabilizer ZX-calculus. Logical Methods in Computer Science* 16(4), doi:10.23638/LMCS-16(4:19)2020.

[6] Niel de Beaudrap, Xiaoning Bian & Quanlong Wang (2020): *Fast and effective techniques for T-count reduction via spider nest identities.* arXiv:2004.05164.

[7] Hector Bombin, Daniel Litinski, Naomi Nickerson, Fernando Pastawski & Sam Roberts (2023): *Unifying Flavors of Fault Tolerance with the ZX Calculus.* arXiv:2303.08829.

[8] Filippo Bonchi, Paweł Sobociński & Fabio Zanasi (2017): *Interacting Hopf Algebras. Journal of Pure and Applied Algebra* 221(1), pp. 144–184, doi:10.1016/j.jpaa.2016.06.002. arXiv:1403.7048.

[9] Robert I. Booth & Titouan Carette (2022): *Complete ZX-calculi for the Stabiliser Fragment in Odd Prime Dimensions.* arXiv:2204.12531.

[10] Robert I. Booth, Titouan Carette & Cole Comfort (2024): *Graphical Symplectic Algebra.* arXiv:2401.07914.

[11] Tristan Cam & Simon Martiel (2023): *Speeding up Quantum Circuits Simulation Using ZX-Calculus.* arXiv:2305.02669.

[12] Titouan Carette (2021): *When Only Topology Matters.* arXiv:2102.03178.

[13] Titouan Carette (2021): *Wielding the ZX-calculus, Flexsymmetry, Mixed States, and Scalable Notations.* Ph.D. thesis, Loria, Université de Lorraine. Available at https://hal.archives-ouvertes.fr/tel-03468027.

[14] Titouan Carette, Dominic Horsman & Simon Perdrix (2019): *SZX-Calculus: Scalable Graphical Quantum Reasoning.* In Peter Rossmanith, Pinar Heggernes & Joost-Pieter Katoen, editors: *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 138, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 55:1–55:15, doi:10.4230/LIPIcs.MFCS.2019.55.

[15] Julien Codsi & John van de Wetering (2023): *Classically Simulating Quantum Supremacy IQP Circuits through a Random Graph Approach.* arXiv:2212.08609.

[16] B. Coecke & R. Duncan (2007): *Interacting Quantum Observables.* Available at www.cs.ox.ac.uk/people/bob.coecke/GreenRed.pdf.

[17] Bob Coecke & Ross Duncan (2008): *Interacting Quantum Observables.* In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir & Igor Walukiewicz, editors: *Automata, Languages and Programming*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 298–310, doi:10.1007/978-3-540-70583-3_25. Available at http://personal.strath.ac.uk/ross.duncan/papers/iqo-icalp.pdf.

[18] Bob Coecke & Bill Edwards (2011): *Three Qubit Entanglement within Graphical Z/X-calculus. Electronic Proceedings in Theoretical Computer Science* 52, pp. 22–33, doi:10.4204/EPTCS.52.3.

[19] Bob Coecke & Stefano Gogioso (2022): *Quantum in Pictures*. Quantinuum.

[20] Bob Coecke & Aleks Kissinger (2017): *Picturing Quantum Processes*. Cambridge University Press, doi:10.1017/9781316219317.

[21] Bob Coecke, Aleks Kissinger, Alex Merry & Shibdas Roy (2011): *The GHZ/W-calculus Contains Rational Arithmetic. Electronic Proceedings in Theoretical Computer Science* 52, pp. 34–48, doi:10.4204/EPTCS.52.4.

[22] Cole Comfort (2023): *The Algebra for Stabilizer Codes*. arXiv:2304.10584.

[23] Alexander Cowtan (2022): *Qudit Lattice Surgery*. arXiv:2204.13228.

[24] Alexander Cowtan & Simon Burton (2023): *CSS Code Surgery as a Universal Construction*. arXiv:2301.13738.

[25] 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.

[26] Niel de Beaudrap, Aleks Kissinger & John van de Wetering (2022): *Circuit Extraction for ZX-Diagrams Can Be #P-Hard*. In Mikołaj Bojańczyk, Emanuela Merelli & David P. Woodruff, editors: *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022), Leibniz International Proceedings in Informatics (LIPIcs)* 229, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 119:1–119:19, doi:10.4230/LIPIcs.ICALP.2022.119.

[27] Giovanni de Felice & Bob Coecke (2022): *Quantum Linear Optics via String Diagrams*. arXiv:2204.12985.

[28] Giovanni de Felice, Boldizsár Poór & Lia Yeh (in prep.): *Fusion and Flow: Formal Protocols to Reliably Build Photonic Graph States*.

[29] Giovanni de Felice, Razin A. Shaikh, Boldizsár Poór, Lia Yeh, Quanlong Wang & Bob Coecke (2023): *Light-Matter Interaction in the ZXW Calculus*. In Shane Mansfield, Benoit Valîron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, Paris, France, 17-21st July 2023, Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, pp. 20–46, doi:10.4204/EPTCS.384.2.

[30] Marc de Visme & Renaud Vilmart (2024): *Minimality in Finite-Dimensional ZW-Calculi*. arXiv:2401.16225.

[31] Ross Duncan, Aleks Kissinger, Simon Perdrix & John van de Wetering (2020): *Graph-Theoretic Simplification of Quantum Circuits with the ZX-calculus. Quantum* 4, p. 279, doi:10.22331/q-2020-06-04-279.

[32] Ross Duncan & Simon Perdrix (2010): *Rewriting Measurement-Based Quantum Computations with Generalised Flow*. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide & Paul G. Spirakis, editors: *Automata, Languages and Programming*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 285–296, doi:10.1007/978-3-642-14162-1_24. Available at http://personal.strath.ac.uk/ross.duncan/papers/gflow.pdf.

[33] Selma Dündar-Coecke, Lia Yeh, Caterina Puca, Sieglinde M.-L. Pfaendler, Muhammad Hamza Waseem, Thomas Cervoni, Aleks Kissinger, Stefano Gogioso & Bob Coecke (2023): *Quantum Picturalism: Learning Quantum Theory in High School*. In: *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 03, pp. 21–32, doi:10.1109/QCE57702.2023.20321. arXiv:2312.03653.

[34] Amar Hadzihasanovic (2015): *A Diagrammatic Axiomatisation for Qubit Entanglement*. In: *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '15, IEEE Computer Society, USA, pp. 573–584, doi:10.1109/LICS.2015.59. arXiv:1501.07082.

[35] Amar Hadzihasanovic (2017): *The Algebra of Entanglement and the Geometry of Composition*. Ph.D. thesis, University of Oxford. arXiv:1709.08086.

[36] Jiaxin Huang, Sarah Meng Li, Lia Yeh, Aleks Kissinger, Michele Mosca & Michael Vasmer (2023): *Graphical CSS Code Transformation Using ZX Calculus*. In Shane Mansfield, Benoit Valîron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic*, *Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, pp. 1–19, doi:10.4204/EPTCS.384.1.

[37] 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, Association for Computing Machinery, New York, NY, USA, pp. 559–568, doi:10.1145/3209108.3209131. arXiv:1705.11151.

[38] Emmanuel Jeandel, Simon Perdrix & Renaud Vilmart (2018): *Diagrammatic Reasoning beyond Clifford+T Quantum Mechanics*. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, Association for Computing Machinery, New York, NY, USA, pp. 569–578, doi:10.1145/3209108.3209139. arXiv:1801.10142.

[39] Aleks Kissinger (2022): *Phase-free ZX diagrams are CSS codes (...or how to graphically grok the surface code)*. arXiv:2204.14038.

[40] Aleks Kissinger & John van de Wetering (2020): *Reducing T-count with the ZX-calculus*. *Physical Review A* 102(2), p. 022406, doi:10.1103/PhysRevA.102.022406. arXiv:1903.10477.

[41] Mark Koch, Richie Yeung & Quanlong Wang (2023): *Speedy Contraction of ZX Diagrams with Triangles via Stabiliser Decompositions*. arXiv:2307.01803.

[42] Tuomas Laakkonen, Konstantinos Meichanetzidis & John van de Wetering (2023): *Picturing Counting Reductions with the ZH-Calculus*. In Shane Mansfield, Benoit Valîron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic*, *Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, Paris, France, pp. 89–113, doi:10.4204/EPTCS.384.6.

[43] Tommy McElvanney & Miriam Backens (2023): *Flow-Preserving ZX-calculus Rewrite Rules for Optimisation and Obfuscation*. In Shane Mansfield, Benoit Valîron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, Paris, France, 17-21st July 2023*, *Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, pp. 203–219, doi:10.4204/EPTCS.384.12.

[44] Kang Feng Ng & Quanlong Wang (2017): *A Universal Completion of the ZX-calculus*. arXiv:1706.09877.

[45] Brendan Pankovich, Alex Neville, Angus Kan, Srikrishna Omkar, Kwok Ho Wan & Kamil Brádler (2023): *Flexible Entangled State Generation in Linear Optics*. arXiv:2310.06832.

[46] Simon Perdrix & Quanlong Wang (2016): *Supplementarity Is Necessary for Quantum Diagram Reasoning*. In: *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 58, Krakow, Poland, pp. 76:1–76:14, doi:10.4230/LIPIcs.MFCS.2016.76.

[47] Boldizsár Poór, Robert I. Booth, Titouan Carette, John van de Wetering & Lia Yeh (2023): *The Qupit Stabiliser ZX-travaganza: Simplified Axioms, Normal Forms and Graph-Theoretic Simplification*. In Shane Mansfield, Benoit Valîron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, Paris, France, 17-21st July 2023*, *Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, pp. 220–264, doi:10.4204/EPTCS.384.13.

[48] 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), Boston, MA, USA, pp. 1–14, doi:10.1109/LICS56636.2023.10175672. arXiv:2302.12135.

[49] Patrick Roy, John van de Wetering & Lia Yeh (2023): *The Qudit ZH-Calculus: Generalised Toffoli+Hadamard and Universality.* In Shane Mansfield, Benoit Valîron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, Paris, France, 17-21st July 2023*, *Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, pp. 142–170, doi:10.4204/EPTCS.384.9.

[50] Christian Schröder de Witt & Vladimir Zamdzhiev (2014): *The ZX-calculus Is Incomplete for Quantum Mechanics.* In Bob Coecke, Ichiro Hasuo & Prakash Panangaden, editors: *Proceedings of the 11th Workshop on Quantum Physics and Logic, Kyoto, Japan*, *Electronic Proceedings in Theoretical Computer Science* 172, Open Publishing Association, pp. 285–292, doi:10.4204/EPTCS.172.20.

[51] Razin A. Shaikh & Stefano Gogioso (2022): *Categorical Semantics for Feynman Diagrams.* arXiv:2205.00466.

[52] Razin A. Shaikh, Quanlong Wang & Richie Yeung (2022): *How to Sum and Exponentiate Hamiltonians in ZXW Calculus.* arXiv:2212.04462.

[53] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington & Ross Duncan (2020): *T$|ket\rangle$: A Retargetable Compiler for NISQ Devices.* *Quantum Science and Technology* 6(1), p. 014003, doi:10.1088/2058-9565/ab8e92. arXiv:2003.10611.

[54] Alex Townsend-Teague, Julio Magdalena de la Fuente & Markus Kesselring (2023): *Floquetifying the Colour Code.* In Shane Mansfield, Benoit Valîron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic*, *Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, pp. 265–303, doi:10.4204/EPTCS.384.14.

[55] Alex Townsend-Teague & Konstantinos Meichanetzidis (2022): *Simplification Strategies for the Qutrit ZX-Calculus.* arXiv:2103.06914.

[56] John van de Wetering (2020): *ZX-calculus for the Working Quantum Computer Scientist.* arXiv:2012.13966.

[57] John van de Wetering & Lia Yeh (2023): *Building Qutrit Diagonal Gates from Phase Gadgets.* In Stefano Gogioso & Matty Hoban, editors: *Proceedings 19th International Conference on Quantum Physics and Logic, Wolfson College, Oxford, UK, 27 June - 1 July 2022*, *Electronic Proceedings in Theoretical Computer Science* 394, Open Publishing Association, pp. 46–65, doi:10.4204/EPTCS.394.4.

[58] John van de Wetering, Richie Yeung, Tuomas Laakkonen & Aleks Kissinger (2024): *Optimal Compilation of Parametrised Quantum Circuits.* arXiv:2401.12877.

[59] 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. arXiv:1812.09114.

[60] 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*, *Electronic Proceedings in Theoretical Computer Science* 266, Open Publishing Association, pp. 58–70, doi:10.4204/EPTCS.266.3.

[61] Quanlong Wang (2022): *Qufinite ZX-calculus: A Unified Framework of Qudit ZX-calculi.* arXiv:2104.06429.

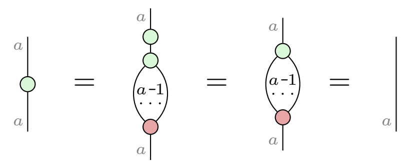
[62] Quanlong Wang, Boldizsár Poór & Razin A. Shaikh (2024): *Completeness of Qufinite ZXW Calculus, a Graphical Language for Finite-Dimensional Quantum Theory.* arXiv:2309.13014.

[63] Quanlong Wang, Richie Yeung & Mark Koch (2022): *Differentiating and Integrating ZX Diagrams with Applications to Quantum Machine Learning.* arXiv:2201.13250.

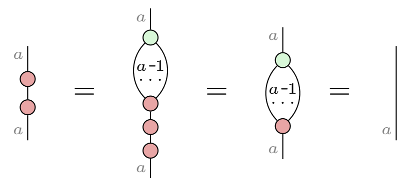
A Lemmas

A.1 General lemmas

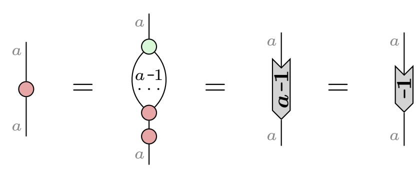
Lemma 4 (Z identity)**.**



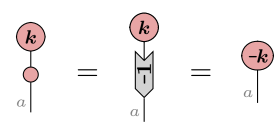
Lemma 5 (X identity)**.**



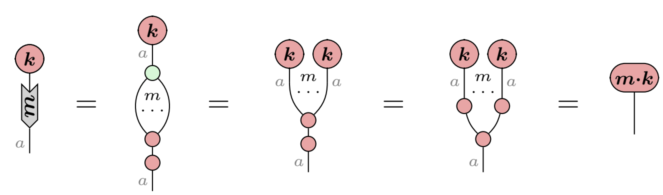
Lemma 6 (Antipode multiplier)**.**



Lemma 7.

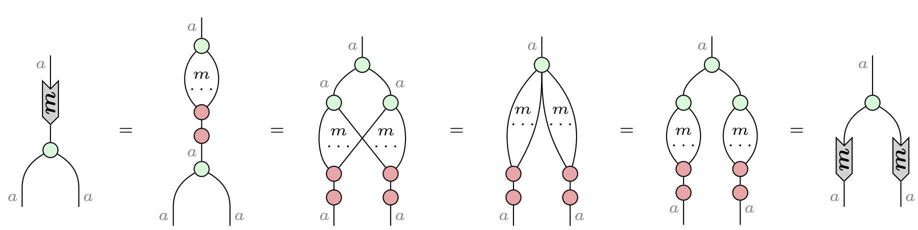


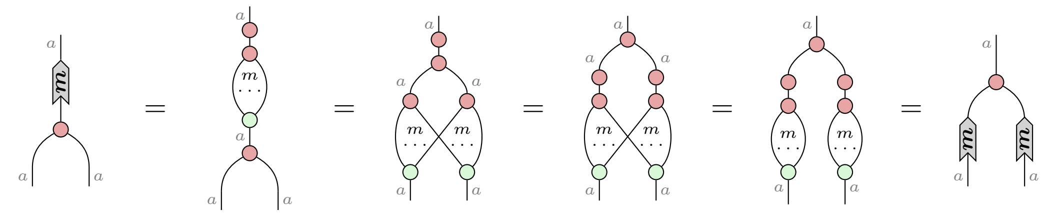
Lemma 8.



Lemma 9.

Lemma 10.

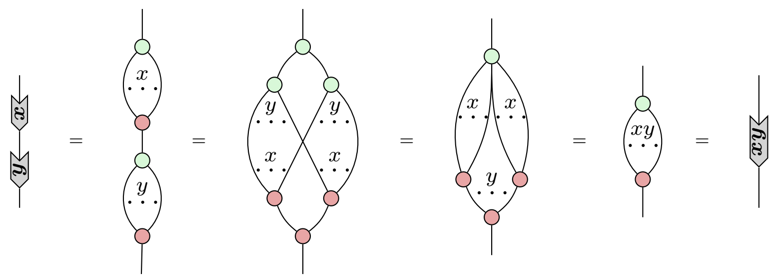




Lemma 11.

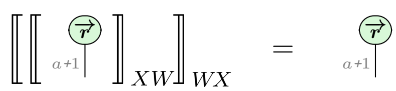
proofs/Lemmas/multiplier/add-

Lemma 12.

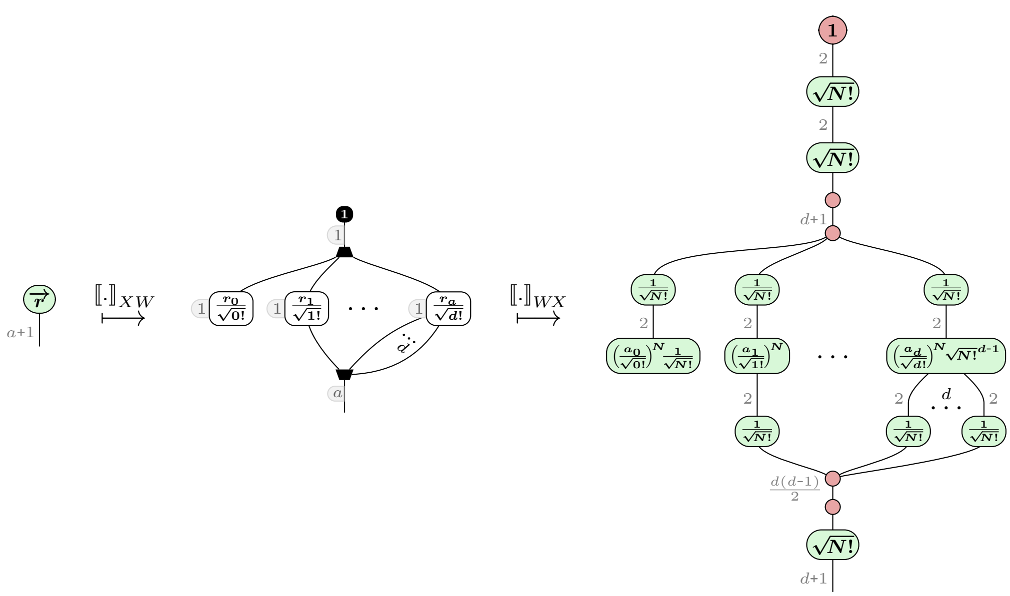


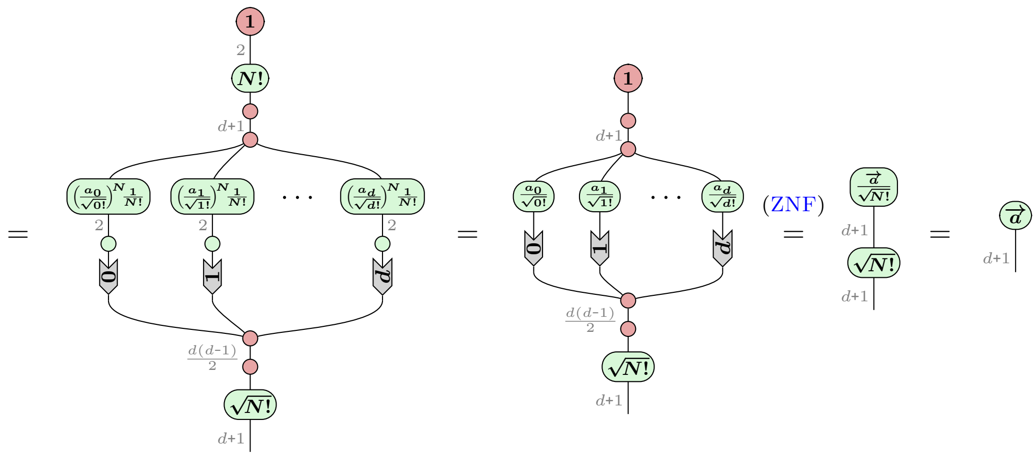
A.2 Recovering generators

Lemma 13.



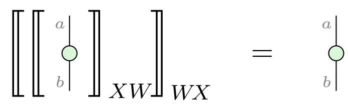
Proof.



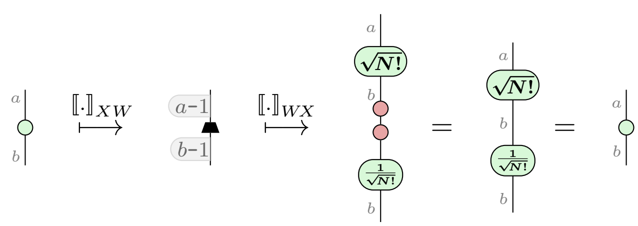


□

Lemma 14.

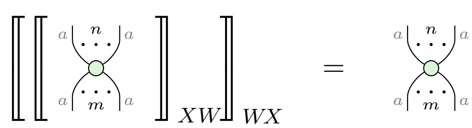


Proof. Suppose $a \geq b$. Then

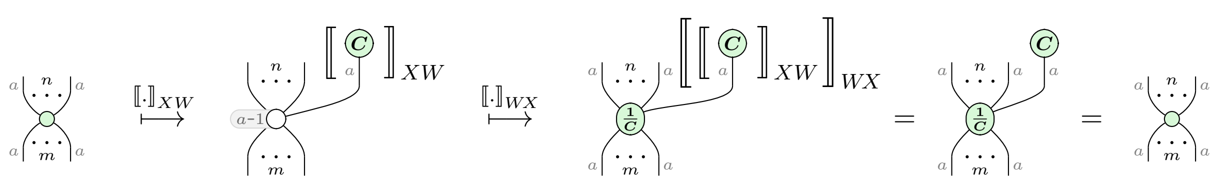


The other case can be proved similarly. □

Lemma 15.

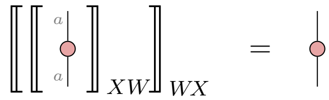


Proof. For $C = \left(\frac{1}{\sqrt{N!}}\right)^{m+n-1}$,

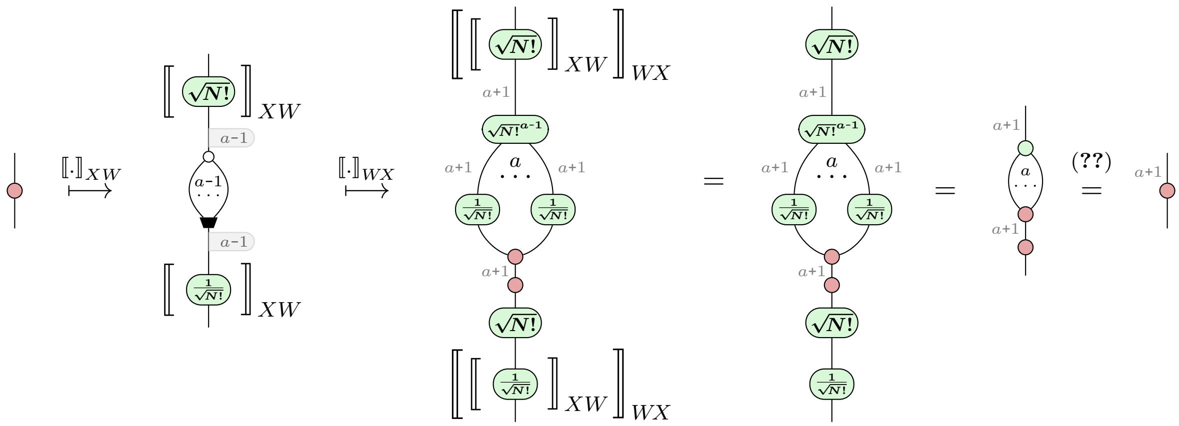


□

Lemma 16.

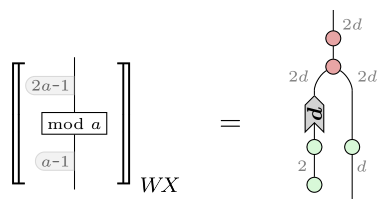


Proof.

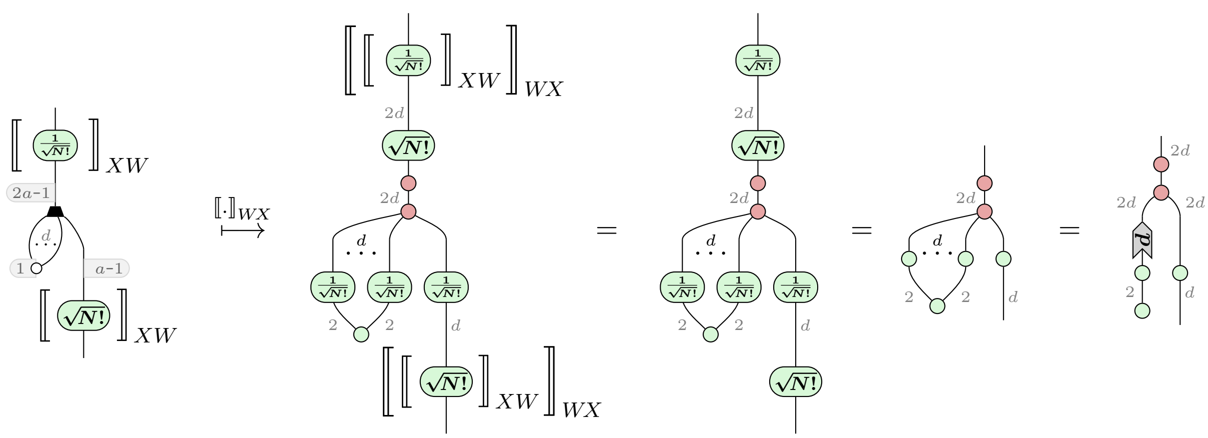


□

Lemma 17.

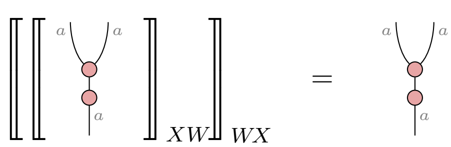


Proof. Expanding the definition of the modulo d box,

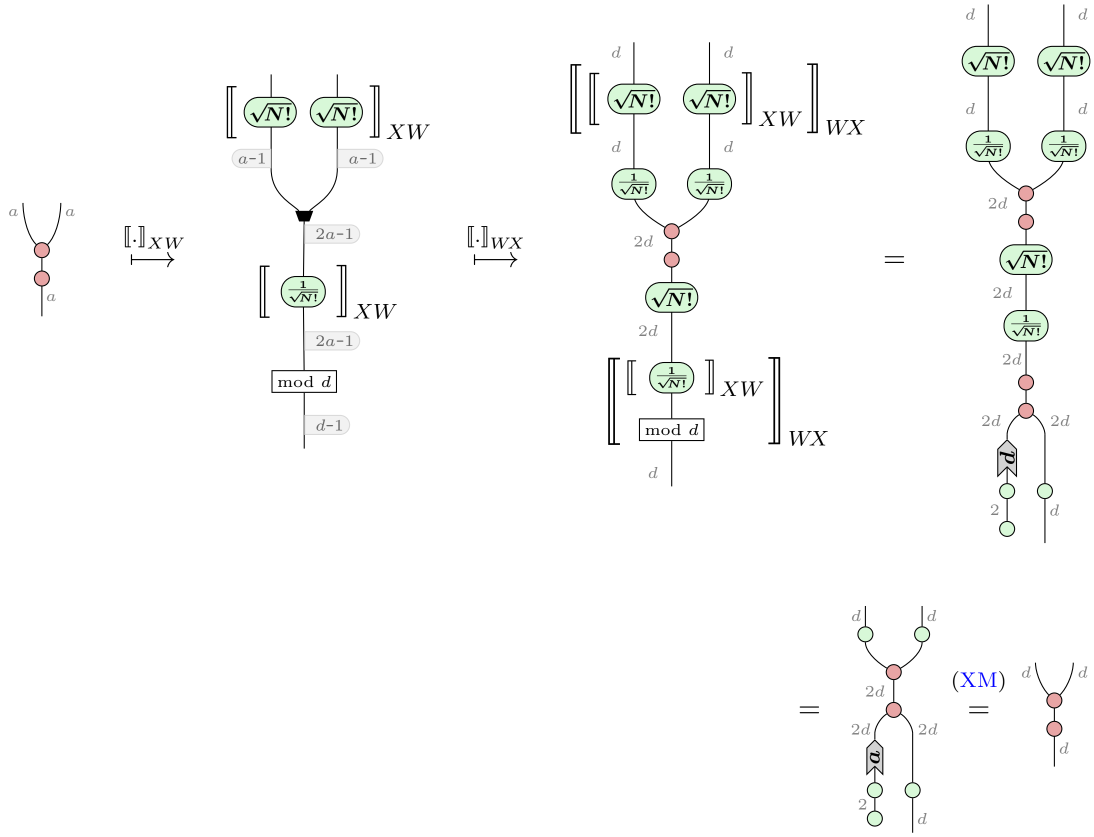


□

Lemma 18.

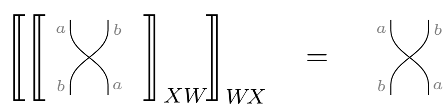


Proof.

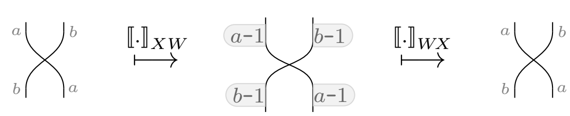


□

Lemma 19.

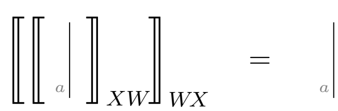


Proof.

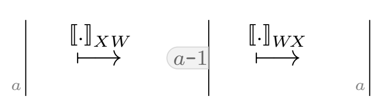


□

Lemma 20.



Proof.

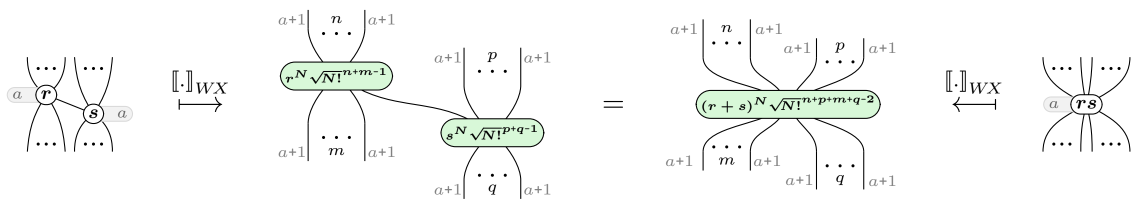


□

A.3 Proving axioms of ZW

Lemma 21. *The translation of the following ZW diagrams under* $[\![\cdot]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

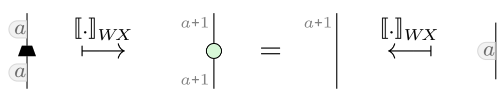
Proof.



□

Lemma 22. *The translation of the following ZW diagrams under* $[\![\cdot]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

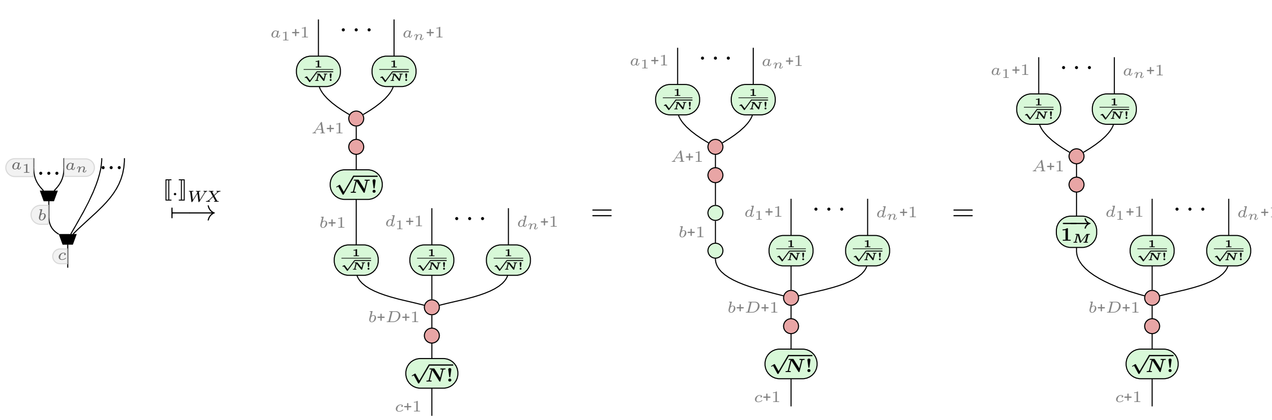
Proof.



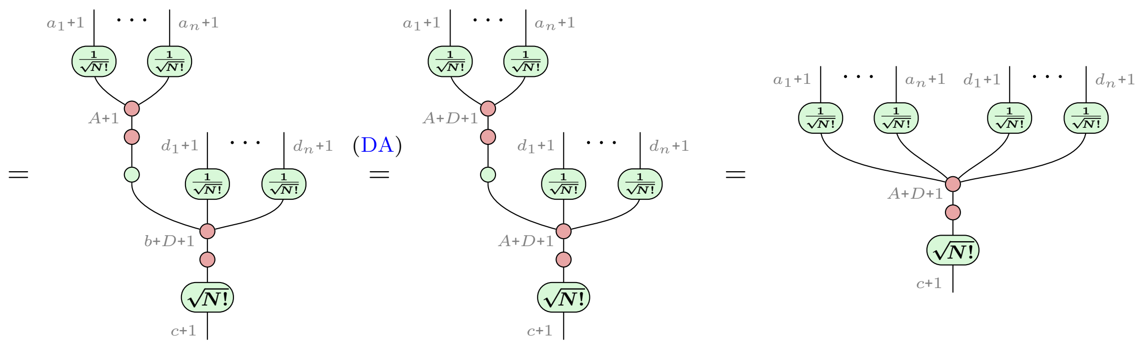
□

Lemma 23. *For* $b \geq \min(c, \sum a_i)$*, the translation of the following ZW diagrams under* $[\![\cdot]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

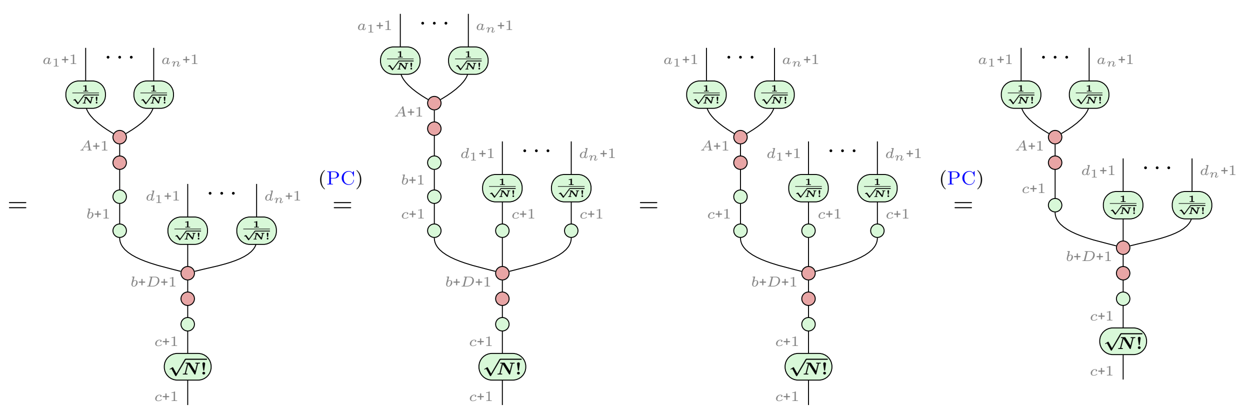
Proof. For $A := \sum a_i$, $D := \sum d_i$, $M = \min\{A, b\}$:



Here, we split the proof based on the value of M. If $A \le b$, then $M = \min\{A, b\} = A$ and we have:



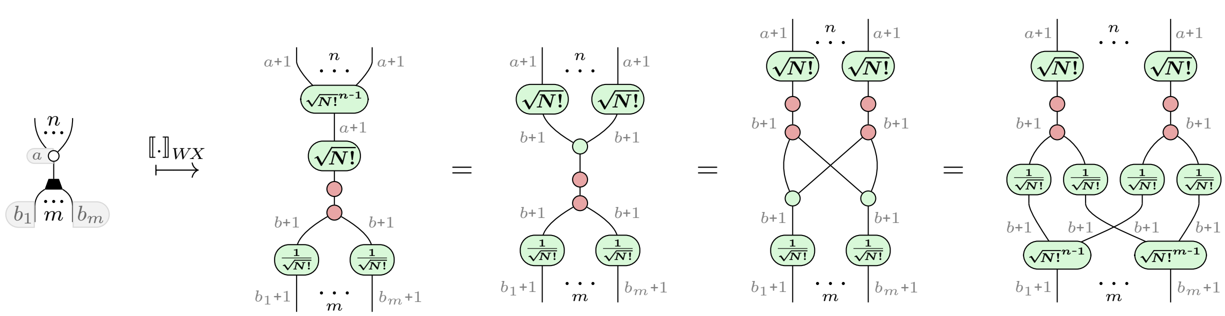
Otherwise, we have $A > b$, and thus $M = \min\{A, b\} = b$. Combining this with the assumption $b \ge \min(c, \sum a_i) = \min(c, A)$, we conclude that $b \ge c$.

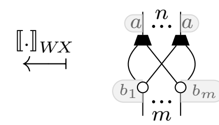


Then the proof follows from the $A \le b$ case. □

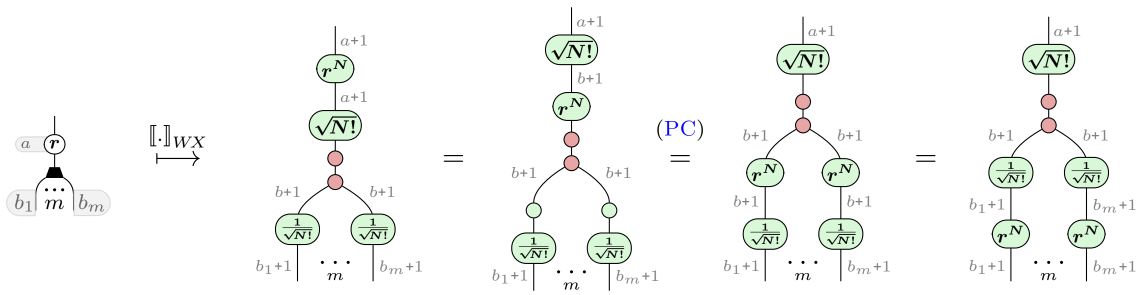
Lemma 24. *For $n \neq 0$, the translation of the following ZW diagrams under $[\![.]\!]_{WX}$ equal in* $\mathbf{ZX_f}$*:*

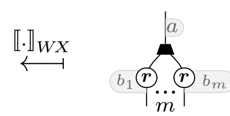
Proof. Let $b \coloneqq \sum b_i$. We first show the following:





Now, we only have to show that the W-spider copies phases:

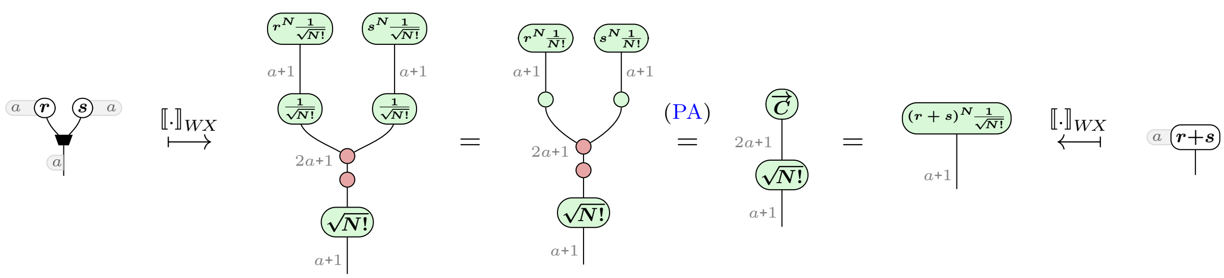




□

Lemma 25. *The translation of the following ZW diagrams under* $[\![.]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

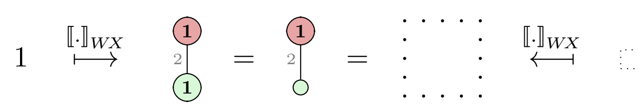
Proof.



where the k-th element of $\overrightarrow{C}$ for $k \leq a$ is given by $c_k = \sum_{i=0}^{a} r^i \frac{1}{i!} s^{k-i} \frac{1}{(k-i)!} = \frac{(r+s)^k}{k!}$ and for $k > a$ it is $c_k = \sum_{i=k-a}^{a} r^i \frac{1}{i!} s^{k-i} \frac{1}{(k-i)!}$; therefore, $\frac{(r+s)^N}{N!}$ defines the first a element of $\overrightarrow{C}$. □

Lemma 26. *The translation of the global scalar* 1 *and the empty diagram* ⬚ *under* $[\![.]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*.*

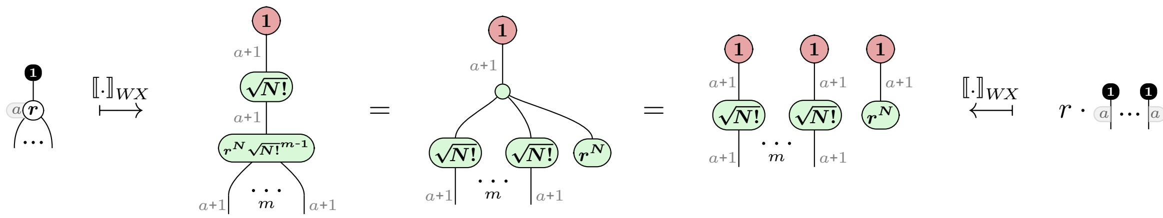
Proof.



□

Lemma 27. *The translation of the following ZW diagrams under* $[\![.]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

Proof.



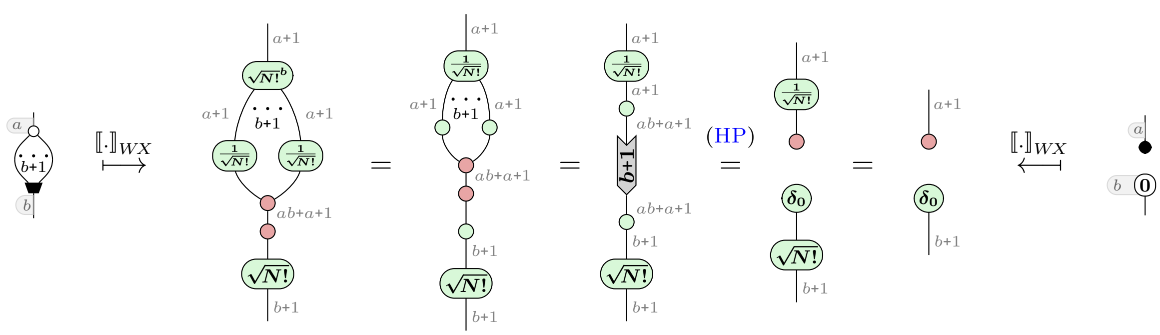
□

Lemma 28. *For* $c \geq \min(\sum a_i, \sum b_i)$*,* $\ell_{ij} = \min(a_i, b_j)$*, the translation of the following ZW diagrams under* $[\![.]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

Proof. This rule follows from Axiom (WW). □

Lemma 29. *The translation of the following ZW diagrams under* $[\![.]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

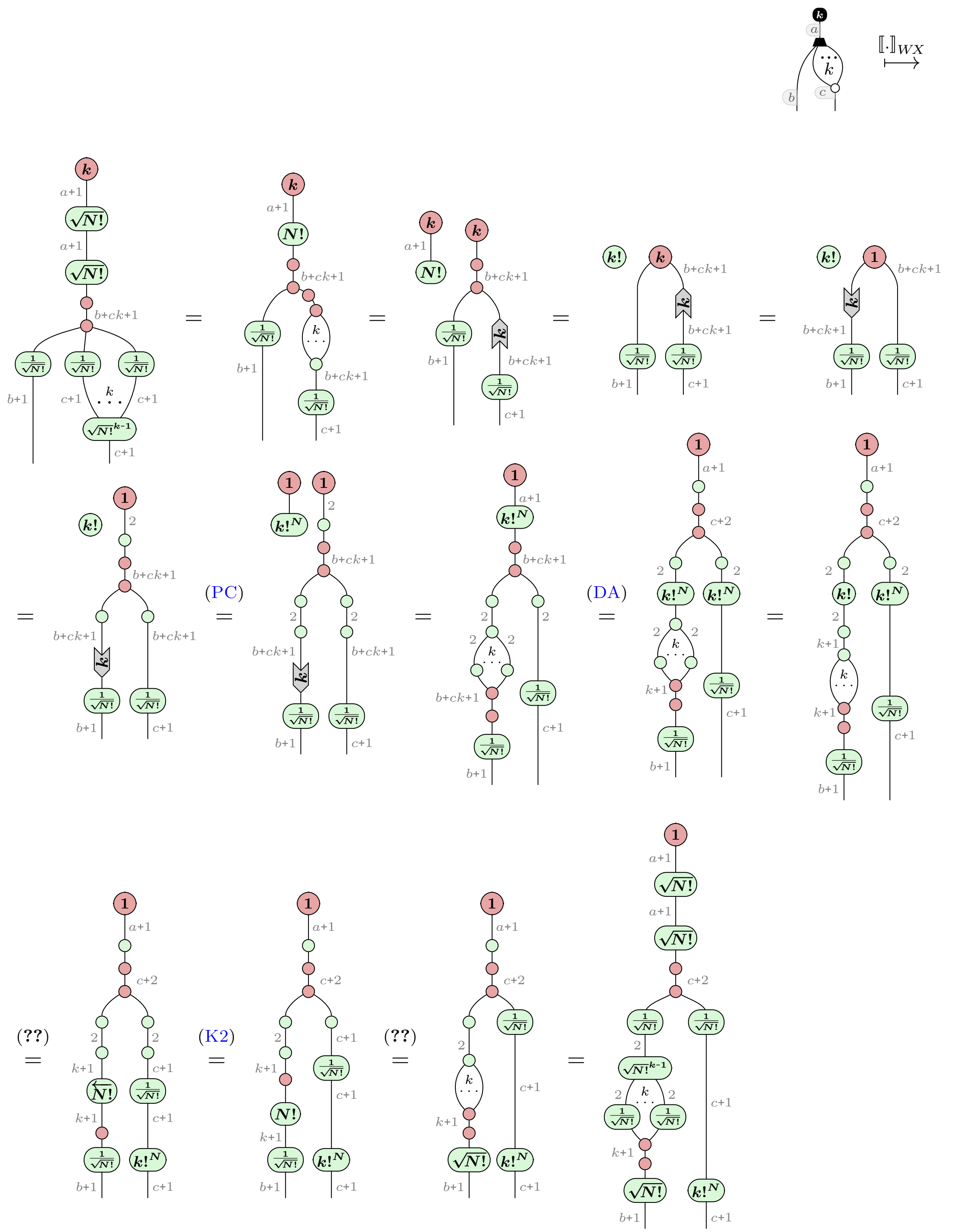
Proof.

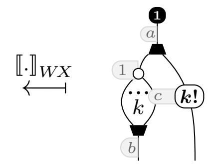


□

Lemma 30. *For* $0 < k \leq a$*, the translation of the following ZW diagrams under* $[\![.]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

Proof.

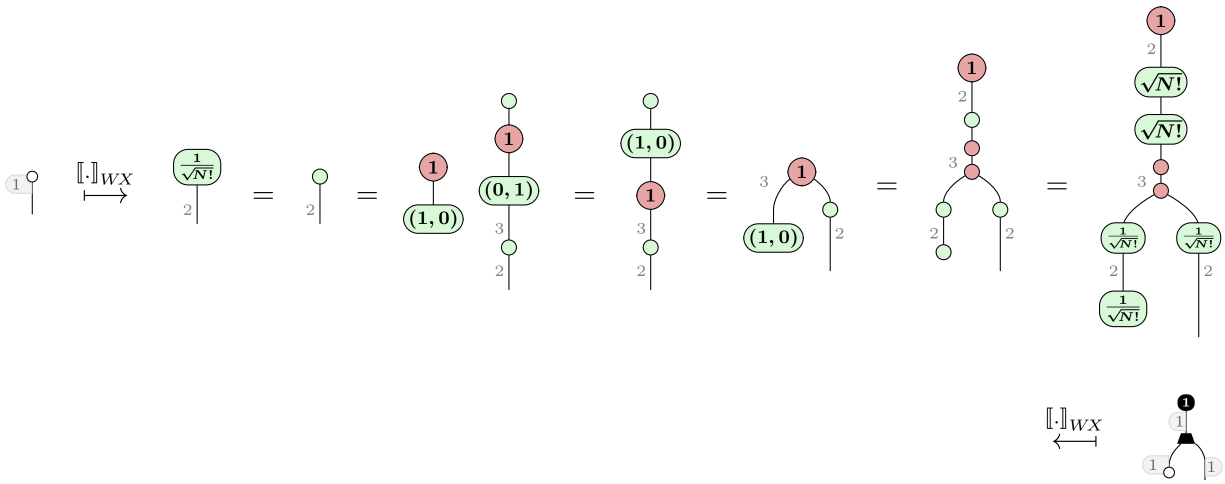




□

Lemma 31. *The translation of the following ZW diagrams under* $[\![.]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

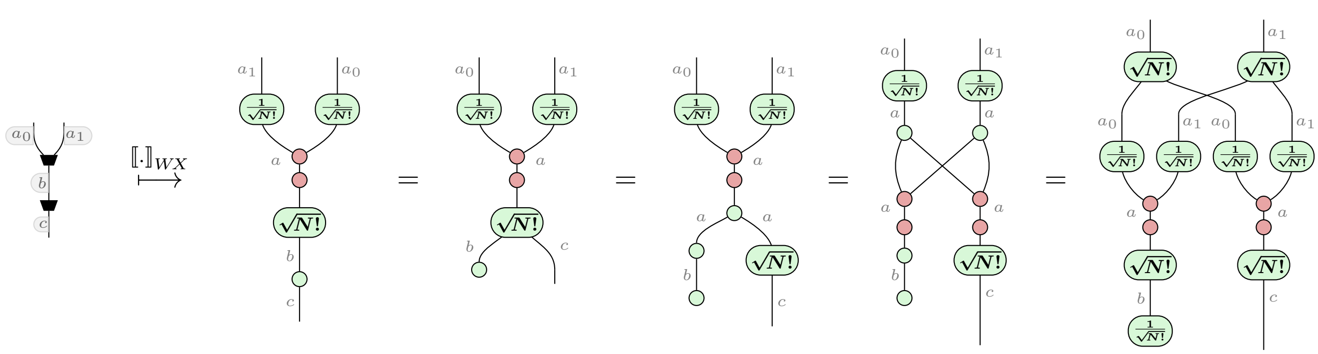
Proof.

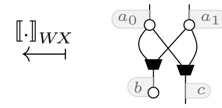


□

Lemma 32. *The translation of the following ZW diagrams under* $[\![.]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

Proof. Let $a = a_0 + a_1$, then

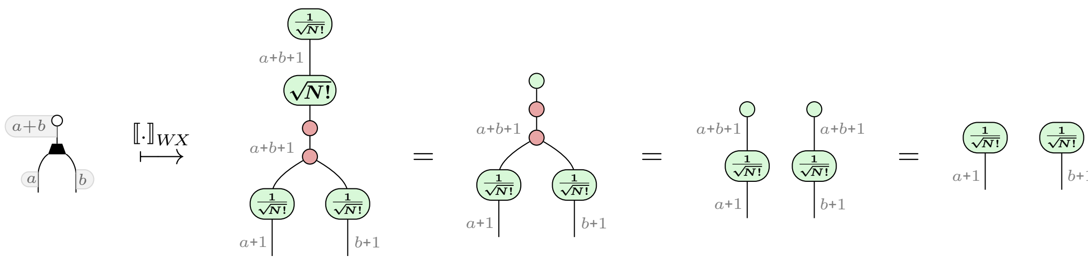




□

Lemma 33. *The translation of the following ZW diagrams under* $[\![.]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

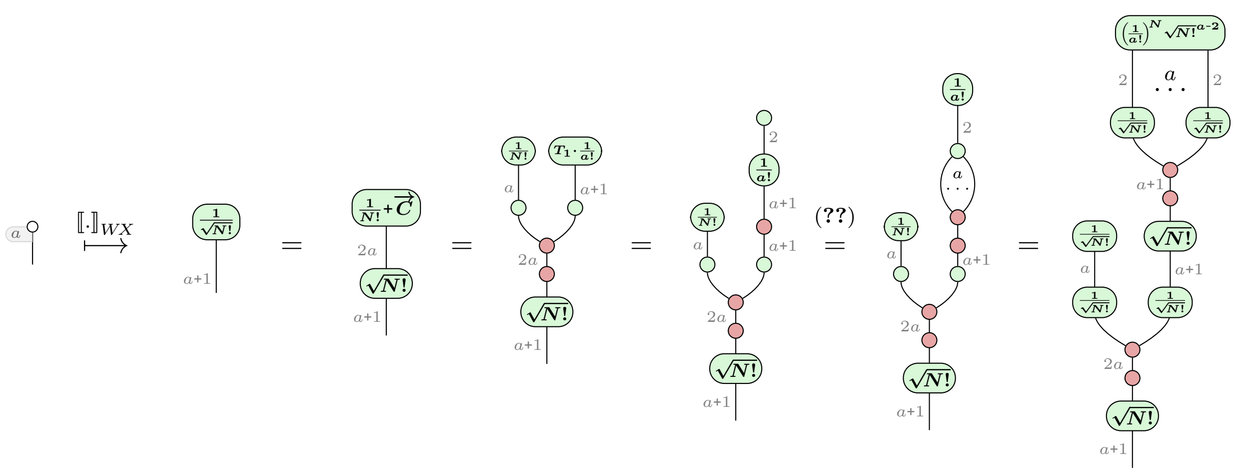
Proof. Let $a = a_0 + a_1$, then

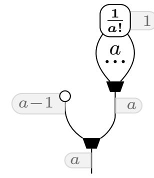


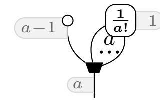
□

Lemma 34. *The translation of the following ZW diagrams under* $[\![.]\!]_{WX}$ *equal in* $\mathbf{ZX_f}$*:*

Proof. For $\overrightarrow{C} = (\underbrace{0, \cdots, 0}_{a+1}, \frac{1}{1!a!}, \frac{1}{2!a!}, \cdots, \frac{1}{(a-1)!a!})$,



$\overset{[\![\cdot]\!]_{WX}}{\longleftarrow\!\!\!\shortmid}$ 

$=$ 

□