Simplification Strategies for the Qutrit ZX-Calculus

Alex Townsend-Teague [1,3] and Konstantinos Meichanetzidis [1,2]

[1] Department of Computer Science, Oxford OX1 3QD, University of Oxford, UK
[2] Quantinuum, 17 Beaumont St. Oxford OX1 2NA, UK
[3] Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany

The ZX-calculus is a graphical language for suitably represented tensor networks, called ZX-diagrams. Calculations are performed by transforming ZX-diagrams with rewrite rules. The ZX-calculus has found applications in reasoning about quantum circuits, condensed matter systems, quantum algorithms, quantum error correcting codes, and counting problems. A key notion is the stabiliser fragment of the ZX-calculus, a subfamily of ZX-diagrams for which rewriting can be done efficiently in terms of derived simplifying rewrites. Recently, higher dimensional qudits - in particular, qutrits - have gained prominence within quantum computing research. The main contribution of this work is the derivation of efficient rewrite strategies for the stabiliser fragment of the qutrit ZX-calculus. Notably, this constitutes a first non-trivial step towards the simplification of qutrit quantum circuits. We then give further unexpected areas in which these rewrite strategies provide complexity-theoretic insight; namely, we reinterpret known results about evaluating the Jones polynomial, an important link invariant in knot theory, and counting graph colourings.

1 Introduction

The ZX-calculus has its origins in quantum foundations [8, 29]. It is a graphical language that allows for reasoning about ZX-diagrams, which are interpreted as tensor networks over a (semi)ring [28]. The calculus is defined by a set of rewrite rules that transform the diagrams. Note that the set of rewrite rules depends on the (semi)ring over which they are interpreted. Importantly, ZX rewrites are sound and complete.

In the context of quantum computing, the Gottesman-Knill theorem states that stabilizer quantum circuits can be simulated efficiently on classical computers [1], where by simulation here we mean the exact computation of quantum probability amplitudes. An alternative proof of the Gottesman-Knill theorem for qubits has been obtained graphically in the qubit ZX-calculus. Stabilizer circuits can be expressed by the *stabilizer fragment*, and stabilizer ZX-diagrams can be rewritten efficiently using derived rewrite strategies [11]. Moreover, these rewrites can be viewed as theorems that are useful for simplifying or simulating universal quantum circuits, not just stabilizer circuits. The key simplifying rewrites for qubit stabilizer ZX-diagrams are called *local complementation* and *pivot*, the latter being a composition of three of the former [11, 12, 22]. In this work, we recall the qutrit version of local complementation [27] and derive the corresponding pivot rule. We then derive simplification strategies which allow for the efficient rewriting of qutrit stabilizer ZX-diagrams, the core result of this work.

Interestingly, though the motivation for studying ZX diagrams stems from quantum computation, they have broader applicability as they can express arbitrary linear maps; every quantum circuit is a ZX-diagram but not vice versa. A plethora of hard problems in physics and computer science regard interacting many-body multi-state systems - both classical and quantum - and reduce to exactly computing a single scalar. These range from quantum amplitudes, partition functions in classical statistical mechanics, counting problems, probabilistic inference, and many more. Problem instances can be encoded

Submitted to:
QPL 2022

as closed tensor networks over an appropriate (semi)ring. The scalar can be evaluated by full tensor contraction, which in general is #P-hard [9].

If a scalar of interest is expressed as a closed ZX-diagram, one can rewrite the diagram by applying rewrite rules; one's goal is then to perform full diagram simplification in order to compute the desired scalar. Again, simplifying arbitrary closed ZX-diagrams is a hard problem. Note that given an arbitrary tensor network, one can attempt to invent rewrite rules by inspecting the contents of the tensors and performing linear-algebra operations [16].

In the spirit of [5], where the ZH-calculus [3], a cousin of ZX, was employed to rederive known complexity results about counting problems, we treat the ZX-calculus as a library of rewrite rules, which we import and use to reason about the complexity of problem families of interest. Here we present two case studies: evaluating the Jones polynomial at lattice roots of unity, and graph colouring. Both of these problem families reduce to evaluating closed tensor networks and show a transition in complexity at a particular dimension, below which the tensor network corresponds to a stabilizer ZX-diagram.

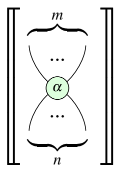
We underline that throughout this work, all rewrites are valid up to a scalar, but keeping track of multiplicative scalar factors that arise under rewriting can be done efficiently.

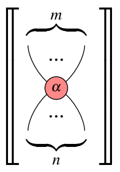
2 Simplifying Qubit ZX-Diagrams

Qubit ZX-diagrams are generated by *spiders*, whose *legs* or *wires* carry vector spaces of dimension $d = 2$. Diagrams are read bottom-to-top; bottom open wires (not connected to anything) are *input wires* and top ones are *outputs*. Diagrams with only outputs are called *states* and those with only inputs are called *effects*. A *closed* diagram is one with no inputs nor outputs and represents a scalar.

Spiders can be *composed*; placing diagrams side by side represents parallel composition ($\otimes$), with concrete interpretation the tensor product. Vertically stacking diagrams corresponds to sequential composition ($\circ$) and concretely means matrix multiplication. Specifically, it means tensor contraction along spider legs; the common tensor indices represented by these wires are summed over. The concrete representation of these operations of course depends on the (semi)ring over which the spiders are interpreted as tensors. For spiders S and S' we write: $[\![S\otimes S']\!] = [\![S]\!]\otimes[\![S']\!]$, $[\![S\circ S']\!] = [\![S]\!]\cdot[\![S']\!]$.

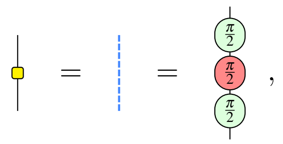
Spiders come in two species: green Z-spiders and red X-spiders, decorated by a *phase* $\alpha \in [0, 2\pi)$. When $\alpha = 0$, we will omit it. The *standard representation* of the spider generators as tensors over $\mathbb{C}$ is:

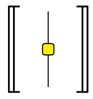


$$= |0\rangle^m\langle 0|^n + e^{i\alpha}|1\rangle^m\langle 1|^n\,,$$


$$= |+\rangle^m\langle +|^n + e^{i\alpha}|-\rangle^m\langle -|^n \qquad (1)$$

where $\{|0\rangle, |1\rangle\}$ is the *Z-basis* and $|\pm\rangle = |0\rangle \pm |1\rangle$ the *X-basis* in $\mathbb{C}^2$, in Dirac notation. The Hadamard gate H, whose function is to switch between the Z and X bases, is denoted as a yellow box. Often we will instead draw a dashed blue line to represent a *Hadamard edge*:




$$\simeq |0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1| \qquad (2)$$

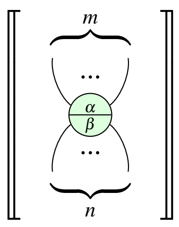
The ZX-calculus is *universal* for multilinear maps; any tensor with entries in $\mathbb{C}$ has a corresponding ZX-diagram. The rewrite rules of the ZX-calculus (see Fig.2 in Appendix A) allow manipulation of the diagrams by *rewrites*. Importantly, up to a scalar, the rewrites are *sound*; that is, they preserve the concrete tensor representation over $\mathbb{C}$. The ZX-calculus is also *complete* in the sense that any true equation between tensors can be proven only in terms of rewrites. An incredibly useful feature of the qubit ZX-diagrams is that *only topology matters*; the concrete tensor semantics of the diagram are invariant under deformations of the network as long as the inter-spider connectivity is respected.

The *stabilizer fragment* of the calculus consists of all diagrams in which all phases are $\alpha = \frac{\pi n}{2}$, $n \in \mathbb{Z}$. In [11, Theorem 5.4] the authors give an efficient algorithm for simplifying any qubit ZX-diagram (see Appendix A). The algorithm consists of consecutive applications of spider-eliminating rewrites. In particular, this algorithm will efficiently simplify any closed stabilizer diagram until it contains at most one spider, at which point the scalar it represents can be easily read off. By 'efficiently' we mean via a sequence of spider elimination rewrites whose cost is polynomial in the initial number of spiders and their legs. Note each such rewrite updates only a polynomial number of edges in the diagram, which prevents an overwhelming memory cost of the simplification procedure.

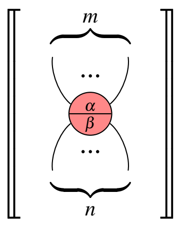
3 Simplifying Qutrit ZX-Diagrams

We now turn to the qutrit ZX-calculus and examine the analogous story to that of the previous section, but for the case where the dimension of the vector space carried by the wires is $d = 3$.

Qutrit spiders again come in two species, Z (green) and X (red), with the three-dimensional *Z-basis* $\{|0\rangle, |1\rangle, |2\rangle\}$. Let $\omega = e^{i\frac{2\pi}{3}}$ denote the third root of unity with $\bar{\omega} = \omega^2$ its complex conjugate. The qutrit *X-basis* is: $\{|+\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), |\omega\rangle = \frac{1}{\sqrt{3}}(|0\rangle + \omega|1\rangle + \bar{\omega}|2\rangle), |\bar{\omega}\rangle = \frac{1}{\sqrt{3}}(|0\rangle + \bar{\omega}|1\rangle + \omega|2\rangle)\}$. Spiders now carry two phases α and β, and have the following *standard representation* as linear maps:



$$= |0\rangle^{\otimes m}\langle 0|^{\otimes n} + e^{i\alpha}|1\rangle^{\otimes m}\langle 1|^{\otimes n} + e^{i\beta}|2\rangle^{\otimes m}\langle 2|^{\otimes n} \tag{3}$$



$$= |+\rangle^{\otimes m}\langle +|^{\otimes n} + e^{i\alpha}|\omega\rangle^{\otimes m}\langle \omega|^{\otimes n} + e^{i\beta}|\bar{\omega}\rangle^{\otimes m}\langle \bar{\omega}|^{\otimes n} \tag{4}$$

When $\alpha = \beta = 0$ we will again omit the angles entirely, and just draw a small green or red dot. Throughout our work, whenever we use an integer n as a spider decoration, this is a shorthand for $\frac{2\pi}{3}n$. Since spider phases hold mod 2π, these integer decorations hold mod 3. Unless otherwise stated, we will use Greek letters to denote general angles, and Roman letters for these integer shorthands.

Hadamard gates are no longer self-adjoint, so we change our notation: we let a yellow box decorated with a 1 (mod 3) denote a Hadamard gate, while decorating with a 2 (mod 3) denotes its adjoint. We will shortly explain this choice. We also use a dashed blue line for the *Hadamard edge* (*H-edge*) and a purple dashed line for its adjoint ($H^\dagger$*-edge*). Those familar with the ZH-calculus should note that this notation

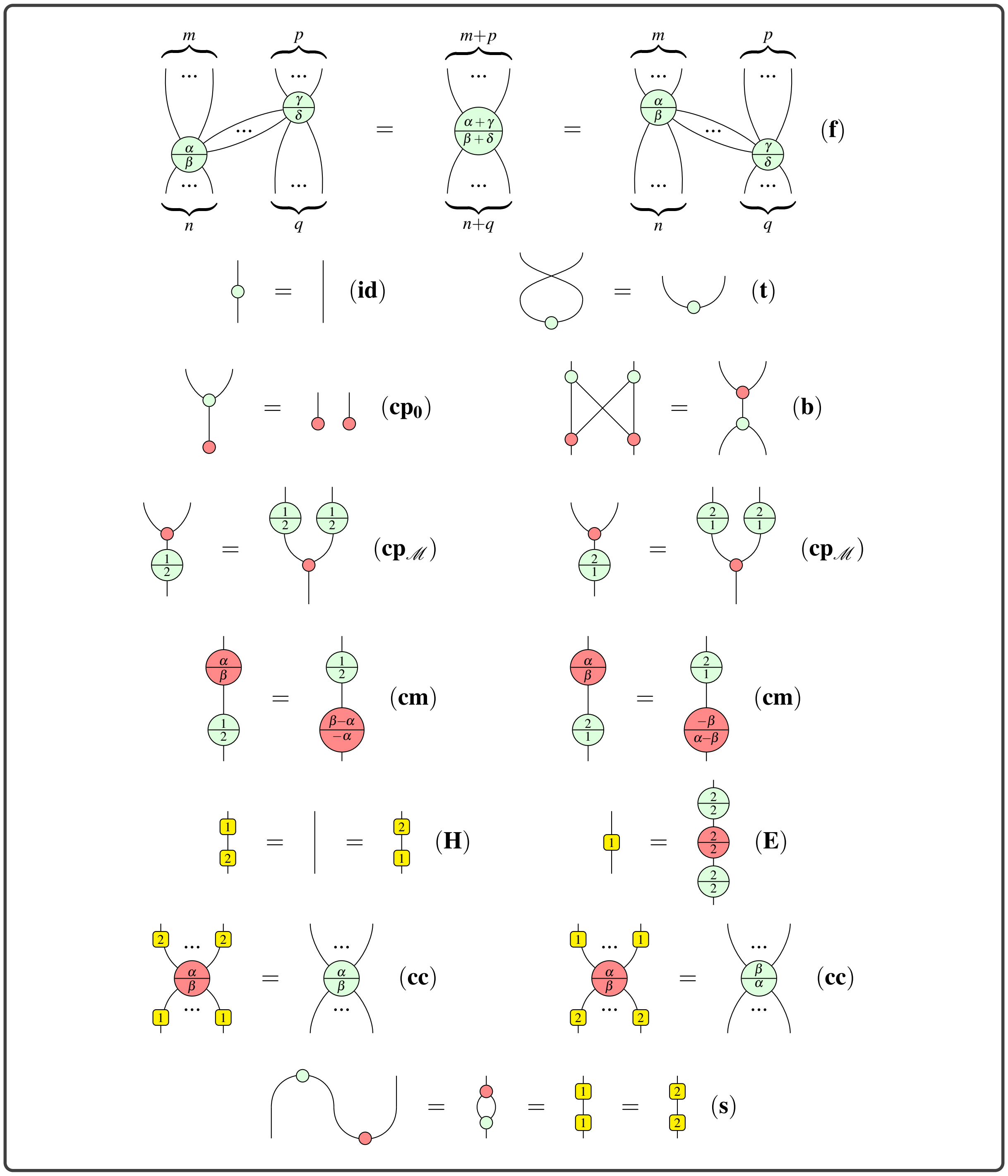
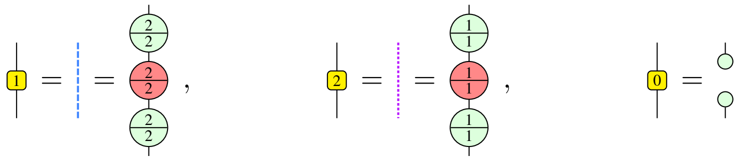


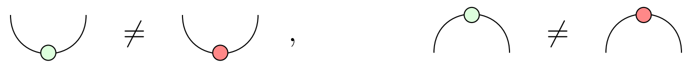
Figure 1: Rewrite rules for the qutrit ZX-calculus where spiders are interpreted as tensors over $\mathbb{C}$.

is *not* analogous to the H-boxes therein.

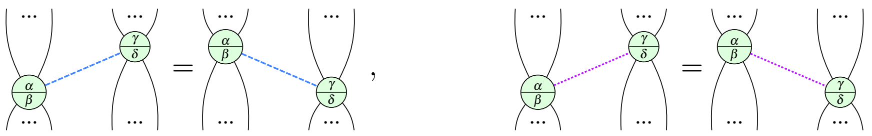


(5)

The last equation above says that in a graph-like diagram (which we will define shortly), a Hadamard edge decorated by a 0 is in fact not an edge at all, thanks to spider fusion. A very important difference from the qubit case is that in qutrit ZX-calculus there is no *plain* cap or cup:


(6)

This has several consequences. Firstly, the maxim that 'only topology matters' no longer applies. That is, it is now important to make clear the distinction between a spider's input and output wires, unlike in the qubit case where we could freely interchange the two. This gives the qutrit calculus a slightly more rigid flavour than its qubit counterpart. That said, this rigidity can be loosened; in particular, this distinction is irrelevant for H- and $H^\dagger$-edges [14]:



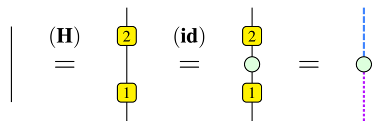
(7)

The full set of rules of the qutrit ZX-calculus is shown in Figure 1, and a more rigorous definition of the calculus as a whole is found in Appendix B.

3.1 Graph-Like Qutrit ZX Diagrams

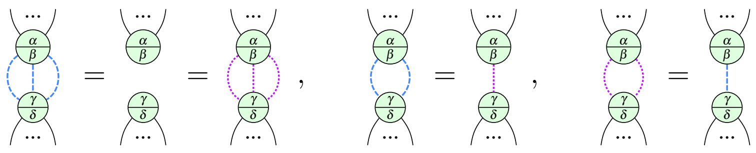
A *graph-like* qutrit ZX-diagram is one where every spider is green, spiders are only connected by blue Hadamard edges (H*-edges*) or their purple adjoints ($H^\dagger$*-edges*), every pair of spiders is connected by at most one H-edge or $H^\dagger$-edge, every input and output is connected to a spider, and every spider is connected to at most one input or output. A graph-like qutrit ZX-diagram is a *graph state* when every spider has zero phases and is connected to an output.

Note the difference compared to the qubit case: we need not worry about self-loops beacuse the qutrit ZX-calculus doesn't define a 'plain' cap or cup. But this comes at a cost: spiders in the qutrit case fuse more fussily. Specifically, when two spiders of the same colour are connected by at least one plain edge and at least one H- or $H^\dagger$-edge, fusion is not possible. Instead, should we want to ensure we have a graph-like diagram, we can replace the plain wire:



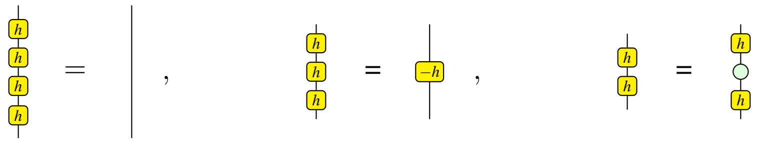
(8)

Indeed, we can show that every qutrit ZX-diagram is equivalent to a graph-like one. The following equations, derived in Appendix B.3, are vital to this:



(9)

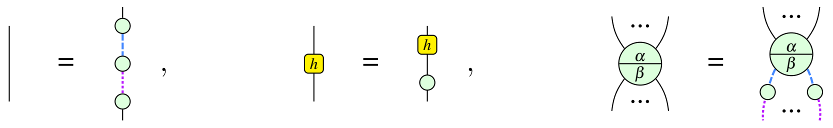
This justifies our notation for Hadamard gates: we can think of Hadamard edges (in blue) as 1-weighted edges and their adjoints (purple) as 2-weighted edges, then work modulo 3, since every triple of parallel edges disappears. Where the previous equations relate single H- and $H^\dagger$-boxes across multiple edges, the next three relate multiple H- and $H^\dagger$- boxes on single edges. They hold for $h \in \{1,2\}$, and are proved via simple applications of rules (**id**), (**H**) and (**s**).



(10)

Proposition 3.1. Every qutrit ZX-diagram is equivalent to one that is graph-like.

Proof. First use the colour change rule to turn all X-spiders into Z-spiders. Then use (10) to remove excess H- and $H^\dagger$-boxes, inserting a spider between any remaining consecutive pair of such boxes, so that all spiders are connected only by plain edges, H-edges or $H^\dagger$-edges. Fuse together as many as possible, and apply (8) where fusion is not possible, so that no plain edge connects two spiders. Apply (9) to all connected pairs of spiders until at most one H- or $H^\dagger$-edge remains between them. Finally, to ensure every input and output is connected to a spider and every spider is connected to at most one input or output, we can use (**H**) and (**id**) to add a few spiders, H- and $H^\dagger$-edges as needed:

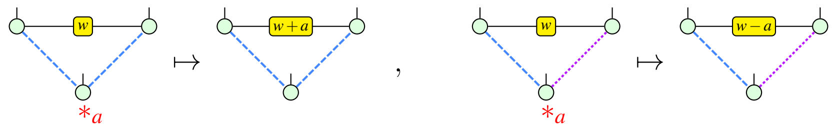


□

A graph state is described fully by its underlying multigraph, or equivalently by an adjacency matrix, where edges take weights in $\mathbb{Z}_3$ [27, Lemma 4.2]. Nodes correspond to phaseless green spiders, edges of weight 1 correspond to Hadamard edges, and edges of weight 2 correspond to $H^\dagger$-edges. As in the qubit case, graph states admit a *local complementation* operation [27, Definition 2.6], though the effect is now slightly more complicated. We'll give the intuition after the formal definition:

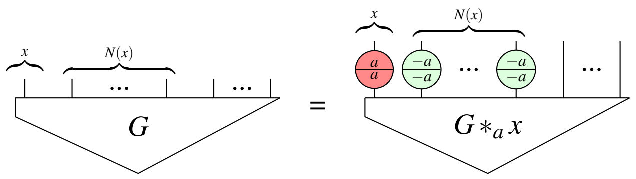
Definition 3.2. Given $a \in \mathbb{Z}_3$ and a graph state G with adjacency matrix $W = (w_{i,j})$, the *a-local complentation* at node x is the new graph state $G *_a x$, whose adjacency matrix $W' = (w'_{i,j})$ is given by $w'_{i,j} = w_{i,j} + a w_{i,x} w_{j,x}$.

So only those edges between neighbours of node x are affected. Specifically, for two nodes i and j both connected to x by the same colour edge, a-local complementation at x increases weight $w_{i,j}$ by a. If instead i and j are connected to x by edges of different colours, a-local complementation at x decreases $w_{i,j}$ by a. This is shown graphically below, and holds with the roles of blue and purple interchanged:



(11)

Theorem 3.3. [27, Theorem 4.4, Corollary 4.5] Given $a \in \mathbb{Z}_3$ and a graph state (G,W) containing a node x, let $N(x)$ denote the neighbours of x; that is, nodes i with weight $w_{i,x} \in \{1,2\}$. Then the following equality holds:

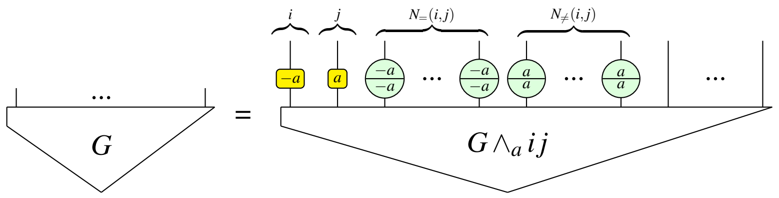


(12)

Definition 3.4. Given $a,b,c \in \mathbb{Z}_3$ and a graph state G containing nodes i and j, the (a,b,c)*-pivot* along ij is the new graph state $G \wedge_{(a,b,c)} ij := ((G \ast_a i) \ast_b j) \ast_c i$.

This pivot operation again leads to an equality, up to introducing some extra gates on outputs, whose proof is found in Appendix C.1. Here we shall only consider an $(a,-a,a)$-pivot along an edge ij of non-zero weight, for $a \in \{1,2\}$. We will call this a *proper a-pivot* along ij, and denote it $G \wedge_a ij$.

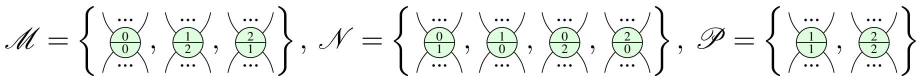
Theorem 3.5. Given $a \in \mathbb{Z}_3$ and a graph state (G,W) containing connected nodes i and j, define $N_{=}(i,j) := \{x \in N(i) \cap N(j) \mid w_{x,i} = w_{x,j}\}$ and $N_{\neq}(i,j) := \{x \in N(i) \cap N(j) \mid w_{x,i} \neq w_{x,j}\}$. Then the following equation relates G and its proper a-pivot along ij:



(13)

3.2 Qutrit Elimination Theorems

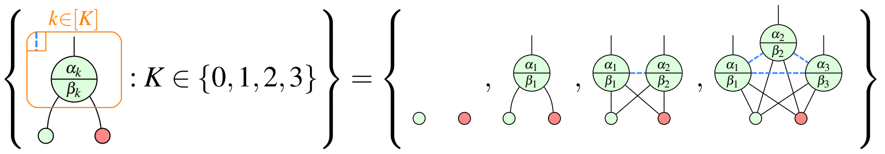
We classify spiders into three families exactly as in [27, Theorem 3.1]:



(14)

We call a spider in a graph-like ZX-diagram *interior* if it isn't connected to an input or output. Given any graph-like ZX-diagram, we will show that we can eliminate standalone interior $\mathscr{P}$- and $\mathscr{N}$-spiders by local complementation, and pairs of connected interior $\mathscr{M}$-spiders by pivoting.

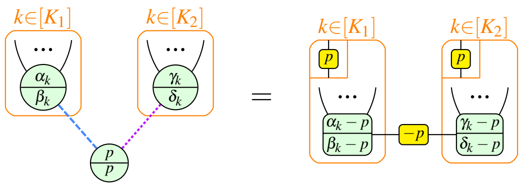
First, we define a modification of the *!-box* notation (pronounced *'bang-box'*), as introduced in [10] for general string diagrams. A !-box is a compressed notation for a family of diagrams; the contents of the !-box are 'repeated' or 'unfolded' $n \geq 0$ times. Following the style of [3], we allow a parameter denoting the maximum number of copies. We also extend this notation as follows: we decorate the !-box with an edge-type, which denotes that the spiders unfolded are all-to-all connected by an edge of this type. In qutrit ZX-diagrams this notation is well-defined in certain scenarios: in particular, when the denoted edge-type is a H- or $H^\dagger$-edge (since then the distinction between a spider's input and output wires disappears) and the main contents of the !-box is equivalent to a single spider (since then there is no ambiguity about which spiders are connected). For example:



(15)

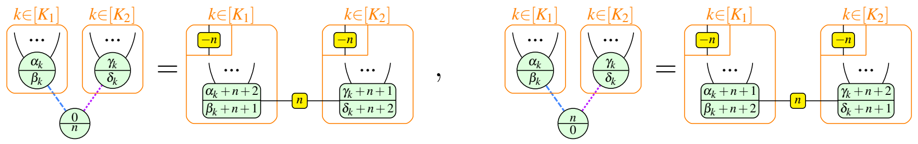
If a !-box has no corner decoration, the unfolded spiders within are not connected by any edge, just as in the usual !-box notation.

Theorem 3.6. Given any graph-like ZX-diagram containing an interior $\mathscr{P}$-spider x with phase $\frac{p}{p}$ for $p \in \{1,2\}$, suppose we perform a p-local complementation at x. Then the new ZX-diagram is related to the old one by the following equality:



(16)

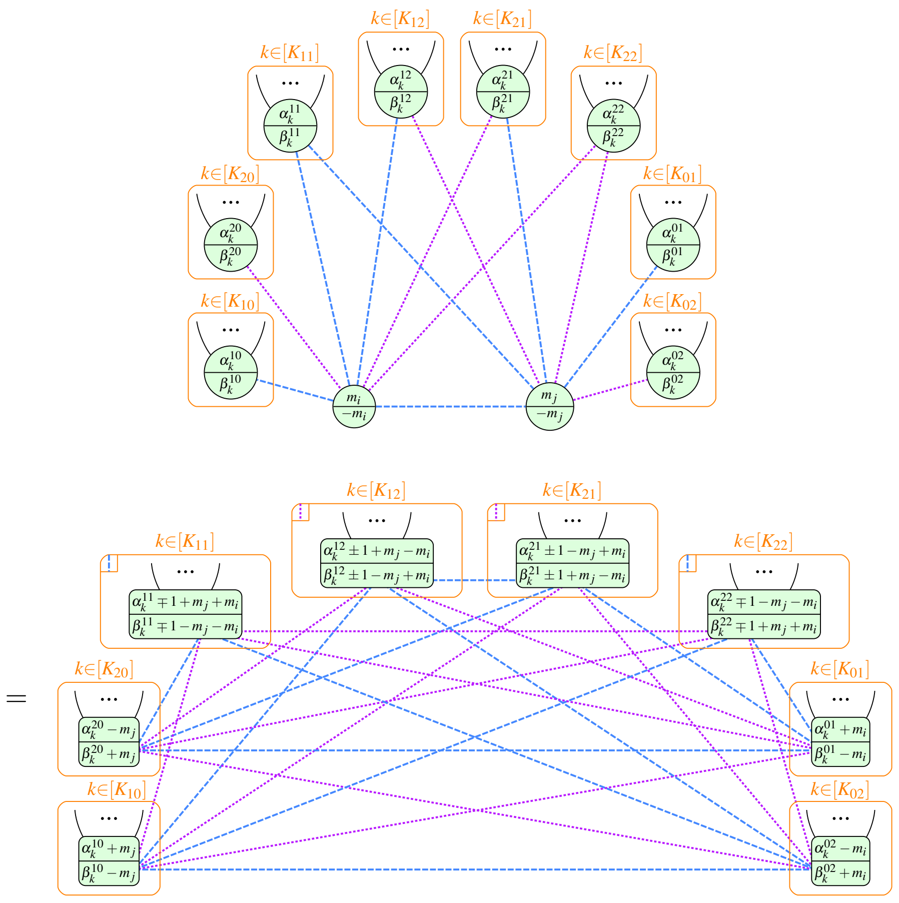
Theorem 3.7. Given any graph-like ZX-diagram containing an interior $\mathscr{N}$-spider x with phase $\frac{0}{n}$ or $\frac{n}{0}$ for $n \in \{1,2\}$, suppose we perform a $(-n)$-local complementation at x. Then, treating the two cases separately, the new ZX-diagrams are related to the old ones by the equalities:



$k\in[K_1]$
$k\in[K_2]$
α_k β_k
γ_k δ_k
0 n
=
$-n$
α_k+n+2 β_k+n+1
n
γ_k+n+1 δ_k+n+2
,
n 0
α_k+n+1 β_k+n+2
γ_k+n+2 δ_k+n+1

(17)

Theorem 3.8. Given any graph-like ZX-diagram containing two interior $\mathscr{M}$-spiders i and j connected by edge ij of weight $w_{i,j} =: w \in \{1,2\}$, suppose we perform a proper $\pm w$-pivot along ij (both choices give the same result). For $w = 1$, the new ZX-diagram is related to the old one by the following equality:



$k\in[K_{11}]$
$k\in[K_{12}]$
$k\in[K_{21}]$
$k\in[K_{22}]$
$k\in[K_{20}]$
$k\in[K_{01}]$
$k\in[K_{10}]$
$k\in[K_{02}]$
α_k^{11} β_k^{11}
α_k^{12} β_k^{12}
α_k^{21} β_k^{21}
α_k^{22} β_k^{22}
α_k^{20} β_k^{20}
α_k^{01} β_k^{01}
α_k^{10} β_k^{10}
α_k^{02} β_k^{02}
m_i $-m_i$
m_j $-m_j$
=
$\alpha_k^{11} \mp 1 + m_j + m_i$ $\beta_k^{11} \mp 1 - m_j - m_i$
$\alpha_k^{12} \pm 1 + m_j - m_i$ $\beta_k^{12} \pm 1 - m_j + m_i$
$\alpha_k^{21} \pm 1 - m_j + m_i$ $\beta_k^{21} \pm 1 + m_j - m_i$
$\alpha_k^{22} \mp 1 - m_j - m_i$ $\beta_k^{22} \mp 1 + m_j + m_i$
$\alpha_k^{20} - m_j$ $\beta_k^{20} + m_j$
$\alpha_k^{01} + m_i$ $\beta_k^{01} - m_i$
$\alpha_k^{10} + m_j$ $\beta_k^{10} - m_j$
$\alpha_k^{02} - m_i$ $\beta_k^{02} + m_i$

(18)

The case $w = 2$ differs only as follows: in the first diagram (the left hand side of the equation), the edge ij is purple (by definition), while in the lower diagram (the right hand side of the equation), a $\pm 2 = \mp 1$ will replace all occurences of ± 1, and the roles of purple and blue will be swapped throughout.

Proofs of these theorems are found in Appendix D.4, D.7 and D.8 respectively. We can now combine them into an algorithm for efficiently simplifying a *closed* graph-like ZX-diagram. First note that after applying any one of the three elimination theorems to such a diagram, and perhaps removing parallel H- or $H^\dagger$-edges via (9), we again have a graph-like diagram.

Theorem 3.9. Given any closed graph-like ZX-diagram, the following algorithm will always terminate after a finite number of steps, returning an equivalent graph-like ZX-diagram with no $\mathscr{N}$-spiders, $\mathscr{P}$-spiders, or adjacent pairs of $\mathscr{M}$-spiders. Repeat the steps below until no rule matches. After each step, apply (9) as needed until the resulting diagram is graph-like:

1. Eliminate a $\mathscr{P}$-spider via Theorem 3.6.
2. Eliminate an $\mathscr{N}$-spider via Theorem 3.7.
3. Eliminate two adjacent $\mathscr{M}$-spiders via Theorem 3.8.

Proof. At every step the total number of spiders decreases by at least one, so since we start with a finite diagram the algorithm terminates after a finite number of steps. By construction, when it does so we are left with an equivalent graph-like ZX-diagram with no $\mathscr{N}$-spiders, $\mathscr{P}$-spiders, or adjacent pairs of $\mathscr{M}$-spiders. □

In particular, if we start with a stabilizer diagram, we can eliminate all but perhaps one spider, depending on whether the initial number of $\mathscr{M}$-spiders was odd or even. This is because no step introduces any non-stabilizer phases. The algorithm above could be extended to a *non-closed* graph-like diagrams as in [11, Theorem 5.4] as part of a qutrit circuit optimisation algorithm, by supplementing with a method for circuit extraction method. We leave this for future work.

4 Case Studies

In this section we present two problems which can naturally be cast in tensor network form. These problem families are interesting in that they show a transition in complexity from easy to hard when the dimension d carried by the wires is greater than a specific value. We recast known results about evaluating the *Jones polynomial*, and finally we briefly look at graph colouring.

4.1 Jones Polynomial at Lattice Roots of Unity

A *knot* K is a circle embedded in $\mathbb{R}^3$. A set of knots tangled together make a *link* L. A link L can be represented by a *link diagram* by projecting it on to the plane but retaining the information of over- or under-crossings. We say that $L \simeq L'$ iff the diagram of link L can be deformed to that of link L' without cutting or gluing strands, or passing strands through each other. The Jones polynomial $V_L(t)$ is a Laurent polynomial in a variable $t \in \mathbb{C}$ and is a *link invariant*. This means that $V_L(t) \neq V_{L'}(t) \Rightarrow L \not\simeq L'$.

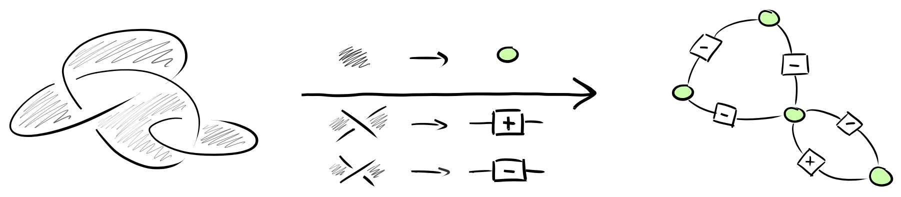
In general, computing $V_L(t)$ is exponentially costly in the number of crossings c, something made explicit when one uses the Kauffman bracket method [18]. Exactly evaluating the Jones polynomial at points $t \in \mathbb{C}$ is #P-hard, except at the *lattice roots of unity* $\Lambda = \{\pm 1, \pm i, \pm e^{i2\pi/3}, \pm e^{i4\pi/3}\}$, where it can be evaluated at cost $O(poly(c))$ [17].

Additively approximating the Jones polynomial at non-lattice roots of unity is a paradigmatic BQP-complete complete problem [2, 19]. Topological quantum computation [13, 23] is the most natural model for such knot theoretic questions. In this model, quantum states are defined in the fusion space of *anyons*, emergent quasiparticles with nontrivial exchange statistics arising in two-dimensional exotic phases of matter. Quantum computation is performed by creating anyons from the vacuum, then braiding them, and finally fusing them back to the vacuum, where braids play the role of unitary gates. The world-lines of the anyons define a closed braid, i.e. a link. Such a link encodes a quantum amplitude corresponding to its Jones polynomial evaluated at a root of unity depending on the anyon theory at hand [31]. Specifically, in the case of $\mathrm{SU}(2)_k$ anyons, the Jones polynomial is evaluated at $t(k) = e^{i2\pi/(2+k)}$ [24].

Also, the evaluation of the Jones polynomial at certain points can be expressed, up to an efficiently computable scalar that depends on the link diagram, as the partition function $Z_{G_L}(d)$ of a d-state Potts model with suitable spin-spin interactions [32]. This Potts model is defined on a signed graph G_L, obtained as follows. The link diagram is bicoloured checkerboard-style, then every coloured area is mapped to a vertex and every crossing is mapped to a signed edge according to its orientation relative to the surrounding colours, as in (19) below. The relation between the point $t(d)$ at which the Jones polynomial is evaluated and the dimension d of the spins is $d = t + t^{-1} + 2$, which can be solved for $t(d)$.

Note the correspondence between the dimension d in the Potts approach and the level k in the anyon-braiding approach to the Jones polynomial: $\{t(d) |\, d \in \{1,2,3,4\}\} = \{t(k) |\, k \in \{1,2,4,\infty\}\} \subseteq \Lambda$. This is consistent with the fact that braiding $\mathrm{SU}(2)_2$ anyons (Ising) or $\mathrm{SU}(2)_4$ anyons is not universal (unless the $\mathrm{SU}(2)_4$ anyons are augmented by fusion and measurements [20]).

The partition function $Z_{G_L}(d \in \mathbb{N})$ can be expressed as a closed tensor network in terms of phaseless (green) d-dimensional Z-spiders connected via wires that go through $\pm$-*boxes* [21]:

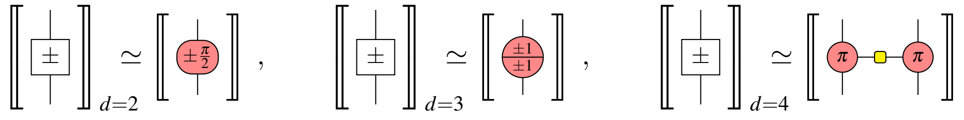


(19)

The $\pm$-boxes have the following concrete interpretation as $d \times d$ matrices:

$$\left[\!\left[\ \boxed{\pm}\ \right]\!\right] = \begin{pmatrix} -t(d)^{\mp 1} & 1 & \dots & 1 \\ 1 & -t(d)^{\mp 1} & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & -t(d)^{\mp 1} \end{pmatrix} \tag{20}$$

The $\pm$-matrices above for $d \in \{2,4\}$ are equal up to a scalar to concrete interpretations of qubit stabilizer ZX-diagrams, and the $\pm$-matrices for $d = 3$ of qutrit stabilizer ZX-diagrams (see Appendix E.1):



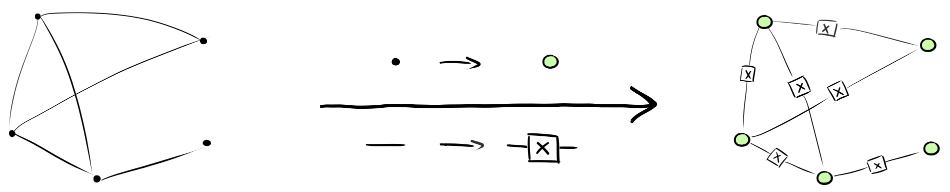
(21)

Since these generators decompose as stabilizer diagrams, $Z_{G_L}(d \in \{2,3,4\})$ can be evaluated efficiently via stabilizer ZX-diagram simplification. Thus, we recover the known result that evaluating the Jones polynomial at $t \in \Lambda$ is in P. On the other hand, computing $Z_{G_L}(d \geq 5)$ is #P-hard.

4.2 Graph Colouring

Finally, let us briefly look at the *graph colouring problem*. A *d-colouring* of a graph G is an assignment of colours $\{1,...,d\}$ to the vertices of G so that no neighbouring vertices have the same colour. Given a graph G and an integer d, we wish to count the number of such d-colourings. Again, this problem can be interpreted as the zero-temperature partition function of an antiferromagnetic d-state Potts model [25]. Through the lens of our graphical exposition we see that counting problems and computing partition functions are essentially the same problem, since both can be straightforwardly encoded as closed tensor networks.

Given a graph G, the graph colouring problem can be encoded as a ZX-diagram as follows. Every vertex of the graph is mapped to a d-dimensional phaseless (green) Z-spider which copies spin states so that they can enter into the X-boxes, each of which in turn enforces that spin states entering it are not the same.


(22)

The X-boxes have the following concrete interpretation as $d \times d$ matrices, and for $d = 2$ they are just the Pauli X:

$$\left[\!\left[\boxed{X} \right]\!\right] = \begin{pmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 0 \end{pmatrix} , \qquad \left[\!\left[\boxed{X} \right]\!\right]_{d=2} \simeq \left[\!\left[\pm\pi \right]\!\right] \tag{23}$$

Counting k-colourings for $k \geq 3$ is a canonical #P-complete problem. For $k = 0, 1, 2$ the problem is in P [17]. For $k = 2$ this is witnessed by the fact that the problem reduces to simplifying a qubit stabilizer diagram, which can be done efficiently. However, for $d = 3$ the X-matrix cannot be expressed as a stabilizer qutrit diagram; we prove this using our qutrit elimination theorems in Appendix E.3. This is consistent with the fact that counting 3-colourings is #P-complete.

5 Discussion and Outlook

In this work, we leveraged local complementation and pivot operations to flesh out non-trivial simplifications strategies for qutrit ZX diagrams. We expect these results to have immediate uses in qutrit circuit optimisation.

Somewhat tangentially, we have also used our new simplification strategies to provide complexity-theoretic insight into certain tensor network problems that demonstrate a 'step-change' in complexity when the dimension d carried by the wires is greater than a particular value. When such problems are translated into the ZX-calculus, this step-change corresponds to whether a diagram is inside or outside of the efficiently reducible stabilizer fragment of the calculus.

A main path for future work entails the generalisation of our stabilizer simplification rules to qudits of higher dimensions. Recent work on the completeness of the stabilizer fragment of the qudit ZX-calculus for odd prime d should provide insight and motivation, as well as some very elegant simplifications to the calculus [6]. Among these simplifications are a restoration of the maxim 'only topology matters' [7], and a representation of any stabilizer phase as a tuple $(x,y) \in \mathbb{Z}_d \times \mathbb{Z}_d$. It would be useful to rederive the qutrit results given here in the language of Ref. [7]. The relevant notion therein is that of 'flexsymmetry', which has also been employed in recent relevant work on qutrit ZX-calculus [30].

On a related note, Ref. [6] concludes by asking about extending their stabilizer fragment completeness results to non-prime dimensions d. In the first author's MSc theis, we defined a parametrisation of stabilizer phases as tuples (x,y) from some group isomorphic to $\mathbb{Z}_d \times \mathbb{Z}_d$ valid for *all* dimensions d, not just odd prime ones [26, Theorem 5.2]. Although finding such a parametrisation is not the main obstacle to extending these completeness results to non-prime dimensions, we feel obliged to refer to this result in case it can contribute towards this endeavour, as to the extent of our knowledge it has not appeared yet in the literature.

The other primary direction of further applied research is in circuit extraction [4] for qudit circuits, where one could hope to find graph-theoretic simplification [11] strategies analogous to those for the case of qubits.

6 Acknowledgements

We wish to thank Niel de Beaudrap, Aleks Kissinger, Stefanos Kourtis and Quanlong Wang for inspiring and helpful discussions, as well as the QPL 2022 and QPL 2022 reviewers for valuable feedback. KM acknowledges financial support from the Royal Commission for the Exhibition of 1851 through a postdoctoral research fellowship, while ATT thanks the Einstein Foundation (Einstein Research Unit on Quantum Devices) for funding.

References

[1] Scott Aaronson & Daniel Gottesman (2004): *Improved simulation of stabilizer circuits*. Physical Review A 70(5), doi:10.1103/physreva.70.052328. Available at `http://dx.doi.org/10.1103/PhysRevA.70.052328`.

[2] Dorit Aharonov, Vaughan Jones & Zeph Landau (2008): *A Polynomial Quantum Algorithm for Approximating the Jones Polynomial*. Algorithmica 55(3), p. 395â€“421, doi:10.1007/s00453-008-9168-0. Available at `http://dx.doi.org/10.1007/s00453-008-9168-0`.

[3] Miriam Backens & Aleks Kissinger (2018): *ZH: A Complete Graphical Calculus for Quantum Computations Involving Classical Non-linearity*. arXiv:1805.02175.

[4] Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski & John van de Wetering (2020): *There and back again: A circuit extraction tale*. arXiv:2003.01664.

[5] Niel de Beaudrap, Aleks Kissinger & Konstantinos Meichanetzidis (2020): *Tensor Network Rewriting Strategies for Satisfiability and Counting*. arXiv:2004.06455.

[6] Robert I. Booth & Titouan Carette (2022): *Complete ZX-calculi for the stabiliser fragment in odd prime dimensions*, doi:10.48550/ARXIV.2204.12531. Available at `https://arxiv.org/abs/2204.12531`.

[7] Titouan Carette (2021): *When Only Topology Matters*, doi:10.48550/ARXIV.2102.03178. Available at `https://arxiv.org/abs/2102.03178`.

[8] 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 `http://dx.doi.org/10.1088/1367-2630/13/4/043016`.

[9] Carsten Damm, Markus Holzer & Pierre McKenzie (2002): *The complexity of tensor calculus*. computational complexity 11(1), pp. 54–89, doi:10.1007/s00037-000-0170-4. Available at `https://doi.org/10.1007/s00037-000-0170-4`.

[10] Lucas Dixon & Ross Duncan (2009): *Graphical reasoning in compact closed categories for quantum computation*. Annals of Mathematics and Artificial Intelligence 56(1), pp. 23–42.

[11] Ross Duncan, Aleks Kissinger, Simon Perdrix & John van de Wetering (2020): *Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus*.

[12] Ross Duncan & Simon Perdrix (2014): *Pivoting makes the ZX-calculus complete for real stabilizers*. Electronic Proceedings in Theoretical Computer Science 171, pp. 50–62, doi:10.4204/eptcs.171.5. Available at `https://doi.org/10.4204%2Feptcs.171.5`.

[13] Michael H. Freedman, Alexei Kitaev, Michael J. Larsen & Zhenghan Wang (2002): *Topological quantum computation*. Bulletin of the American Mathematical Society 40(01), pp. 31–39, doi:10.1090/s0273-0979-02-00964-3. Available at `https://doi.org/10.1090/s0273-0979-02-00964-3`.

[14] Xiaoyan Gong & Quanlong Wang (2017): *Equivalence of Local Complementation and Euler Decomposition in the Qutrit ZX-calculus*.

[15] Xiaoyan Gong & Quanlong Wang (2017): *Equivalence of local complementation and Euler decomposition in the qutrit ZX-calculus*. arXiv preprint arXiv:1704.05955.

[16] Johnnie Gray & Stefanos Kourtis (2020): *Hyper-optimized tensor network contraction*. arXiv:2002.01935.

[17] F. Jaeger, D. L. Vertigan & D. J. A. Welsh (1990): *On the computational complexity of the Jones and Tutte polynomials*. Mathematical Proceedings of the Cambridge Philosophical Society 108(1), p. 35â€“53, doi:10.1017/S0305004100068936.

[18] Louis H Kauffman (2001): *Knots and Physics*. WORLD SCIENTIFIC, doi:10.1142/4256. Available at `https://doi.org/10.1142/4256`.

[19] Greg Kuperberg (2014): *How hard is it to approximate the Jones polynomial?* arXiv:0908.0512.

[20] Claire Levaillant, Bela Bauer, Michael Freedman, Zhenghan Wang & Parsa Bonderson (2015): *Universal gates via fusion and measurement operations onSU(2)4anyons*. Physical Review A 92(1), doi:10.1103/physreva.92.012301. Available at `http://dx.doi.org/10.1103/PhysRevA.92.012301`.

[21] Konstantinos Meichanetzidis & Stefanos Kourtis (2019): *Evaluating the Jones polynomial with tensor networks*. Phys. Rev. E 100, p. 033303, doi:10.1103/PhysRevE.100.033303. Available at `https://link.aps.org/doi/10.1103/PhysRevE.100.033303`.

[22] Maarten Van den Nest & Bart De Moor (2005): *Edge-local equivalence of graphs*, doi:10.48550/ARXIV.MATH/0510246. Available at `https://arxiv.org/abs/math/0510246`.

[23] Jiannis K. Pachos (2012): *Introduction to Topological Quantum Computation*. Cambridge University Press, doi:10.1017/CBO9780511792908.

[24] Eric C. Rowell & Zhenghan Wang (2018): *Mathematics of topological quantum computing*. Bulletin of the American Mathematical Society 55(2), p. 183â€“238, doi:10.1090/bull/1605. Available at `http://dx.doi.org/10.1090/BULL/1605`.

[25] Alan D. Sokal (1999): *Chromatic Polynomials, Potts Models and All That*. doi:10.1016/S0378-4371(99)00519-1. arXiv:arXiv:cond-mat/9910503.

[26] Alex Townsend-Teague (2021): *Qudit ZX-Diagram Simplification*. Master's thesis. Available at `https://github.com/OzYossarian/Writing/blob/main/ZX%20Qutrits%20and%20Qudits.pdf`.

[27] Quanlong Wang (2018): *Qutrit ZX-calculus is Complete for Stabilizer Quantum Mechanics*.

[28] Quanlong Wang (2020): *Completeness of algebraic ZX-calculus over arbitrary commutative rings and semirings*. arXiv:1912.01003.

[29] John van de Wetering (2020): *ZX-calculus for the working quantum computer scientist*. arXiv:2012.13966.

[30] John van de Wetering & Lia Yeh (2022): *Phase gadget compilation for diagonal qutrit gates*, doi:10.48550/ARXIV.2204.13681. Available at `https://arxiv.org/abs/2204.13681`.

[31] Edward Witten (1989): *Quantum field theory and the Jones polynomial*. *Communications in Mathematical Physics* 121(3), pp. 351–399, doi:10.1007/bf01217730. Available at `https://doi.org/10.1007/bf01217730`.

[32] F. Y. Wu (1992): *Knot theory and statistical mechanics*. *Rev. Mod. Phys.* 64, pp. 1099–1131, doi:10.1103/RevModPhys.64.1099. Available at `https://link.aps.org/doi/10.1103/RevModPhys.64.1099`.

A Qubit ZX-Calculus

The well-known rewrite rules of the qubit ZX-calculus are shown in Fig.2. The *stabilizer fragment* of the calculus consists of all diagrams in which all phases are $\alpha = \frac{\pi n}{2}$, $n \in \mathbb{Z}$. In [11, Theorem 5.4] the authors give an efficient algorithm for simplifying any qubit ZX-diagram to an equivalent diagram with fewer spiders. The algorithm consists of consecutive applications of spider-eliminating rewrites.

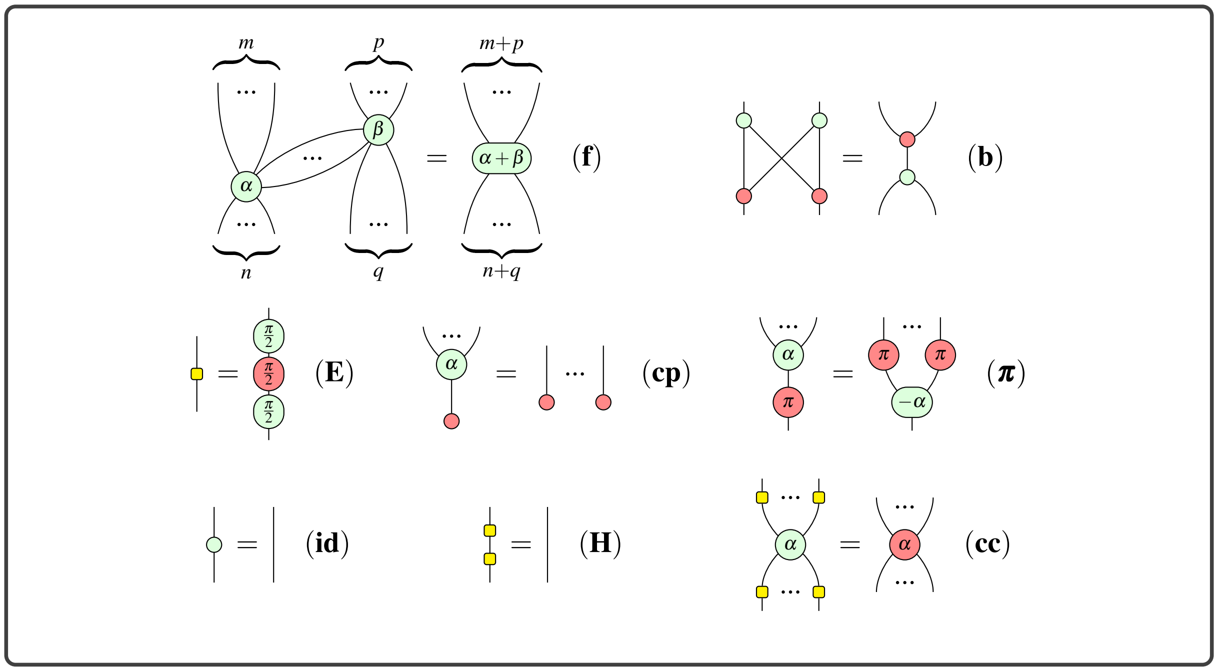
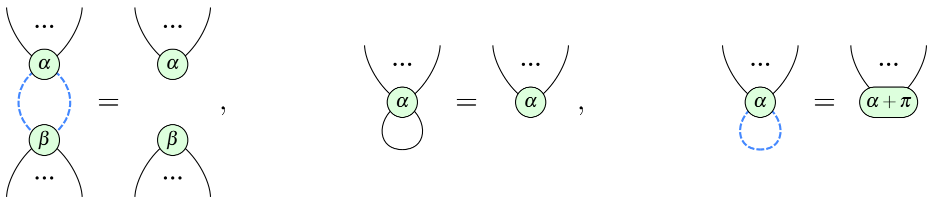
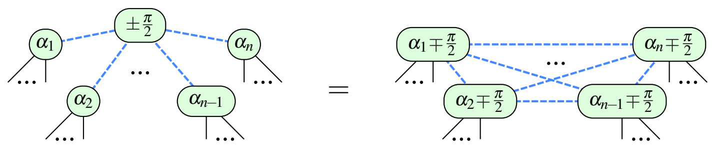


Figure 2: Rewrite rules for the qubit ZX-calculus where spiders are interpreted as tensors over $\mathbb{C}$.

Every diagram is equivalent to a *graph-like* diagram: every spider is green, every edge is a Hadamard edge, there are no parallel edges or self-loops, every input and output wire is connected to a spider and every spider has at most one input or output wire. The following derivable rules are key to this:

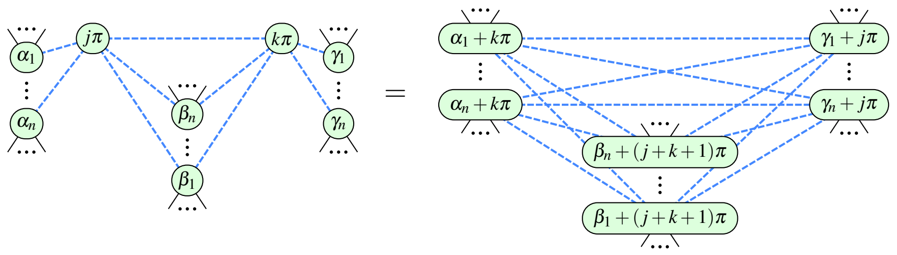
 (24)

Then the following two rewrite rules, derived via *local complementation* and *pivoting*, can be used to eliminate spiders:



$\pm\frac{\pi}{2}$
α_1
α_n
α_2
α_{n-1}
$=$
$\alpha_1\mp\frac{\pi}{2}$
$\alpha_n\mp\frac{\pi}{2}$
$\alpha_2\mp\frac{\pi}{2}$
$\alpha_{n-1}\mp\frac{\pi}{2}$

(25)



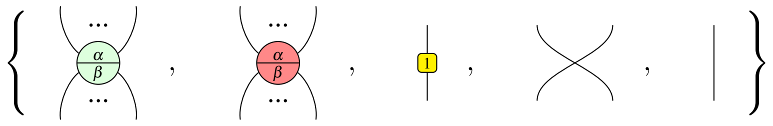
(26)

For more details, see [11, Section 4]. Eq. (25) says that we can remove any spider with phase $\pm\frac{\pi}{2}$ at the cost of performing a local complementation at said spider. Furthermore, Eq. (26) says we can remove any pair of spiders with phases in $\{0,\pi\}$ connected by a Hadamard edge at the cost of performing a pivot along said edge. After each application of (25) or (26), we can use (24) to remove any parallel Hadamard edges and ensure the diagram remains graph-like.

B Qutrit ZX-Calculus

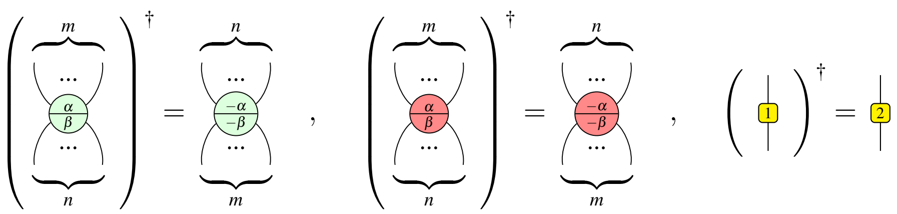
We now give a fuller definition of the qutrit ZX-calculus. Our presentation aims for clarity and accessibility; for a more rigourous description, see [27].

Definition B.1. The *qutrit ZX-calculus* is a graphical calculus generated by the following diagrams, where $\alpha,\beta\in[0,2\pi)$:



(27)

and their adjoints $(-)^\dagger$. Adjoints are found by swapping inputs and outputs and negating any decorations - recall negation is mod 2π for general spider phases, and mod 3 for integer spider phases and Hadamard boxes. Thus the two rightmost generators are self-adjoint, whereas the first three satisfy:



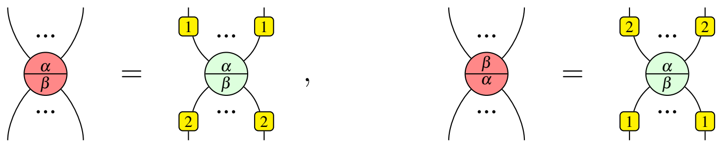
(28)

These generators can be composed in parallel ($\otimes$) and sequentially ($\circ$), and the resulting diagrams are governed by the rewrite rules in Figure 1, wherein addition is modulo 2π. The fusion rule (**f**) applies to spiders of the same colour connected by at least one wire. Importantly, all the rules hold under taking adjoints, where for diagrams D and E we have:

$$(D \otimes E)^\dagger = D^\dagger \otimes E^\dagger\ , \qquad (D \circ E)^\dagger = D^\dagger \circ E^\dagger \tag{29}$$

Furthermore, it can be derived that all but the commutation equations (**cm**) and the colour change equations (**cc**) continue to hold when the roles of green and red (i.e. Z and X) are interchanged. For these four exceptions, however, analogous equations can be derived from the existing ones; for example, the corresponding colour change equations will be relevant for us later.

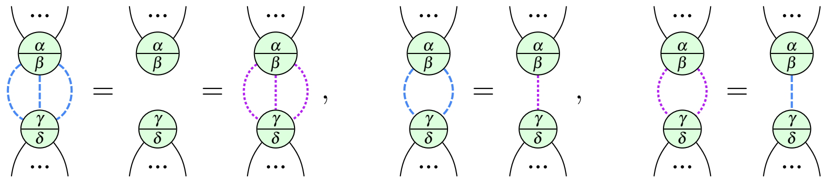
Proposition B.2. The following equations are derivable in the qutrit ZX-calculus:



(30)

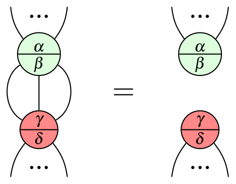
Proof. Add H- and $H^\dagger$-boxes to both sides of the original colour change equations in such a way that we can then cancel Hadamards on the legs of the red spiders via (**H**). □

Lemma B.3. The following equations hold in the qutrit ZX-calculus:



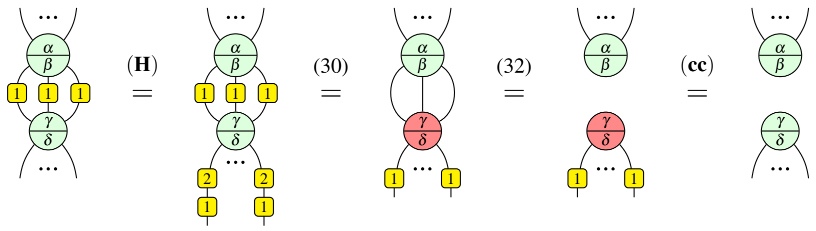
(31)

Proof. It is shown in [14, Lemma 2.8] that the qutrit ZX-calculus satisfies the following ‘Hopf law’:



(32)

Therefore, to prove the first equality of the first equation, we can argue as follows:

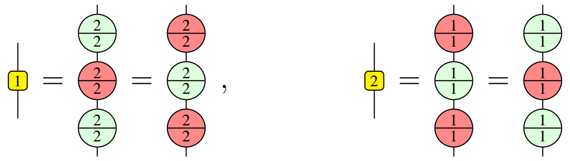


(33)

The second equality of the first equation is proved analogously, and the other two equations are proved in [15, Lemma 3.4]. □

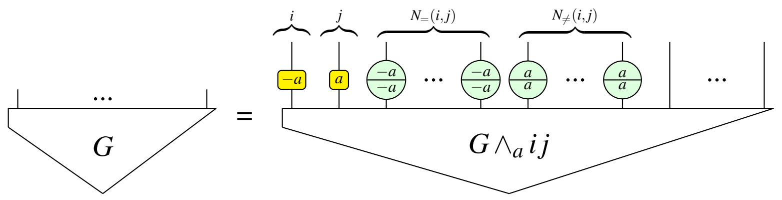
C Pivoting in the Qutrit ZX-Calculus

Next we prove the pivot equality from Theorem 3.5. We require the following: since (**E**) holds under taking adjoints, and swapping the roles of red and green, we have:



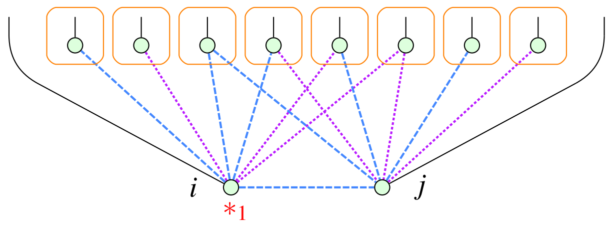
(34)

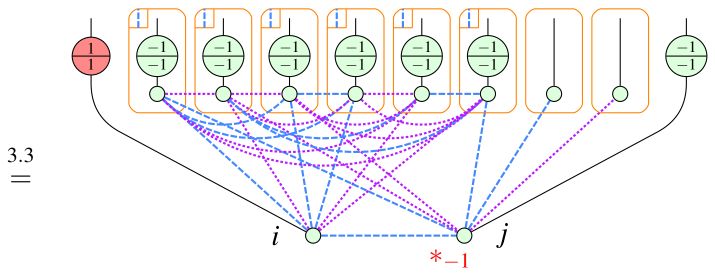
Theorem C.1. / Theorem 3.5. Given $a \in \mathbb{Z}_3$ and a graph state (G,W) containing connected nodes i and j, define $N_=(i,j) := \{x \in N(i) \cap N(j) \mid w_{x,i} = w_{x,j}\}$ and $N_{\neq}(i,j) := \{x \in N(i) \cap N(j) \mid w_{x,i} \neq w_{x,j}\}$. Then the following equation relates G and its proper a-pivot along ij:

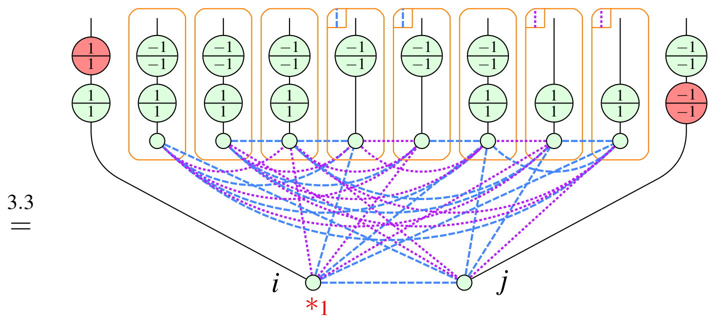


(35)

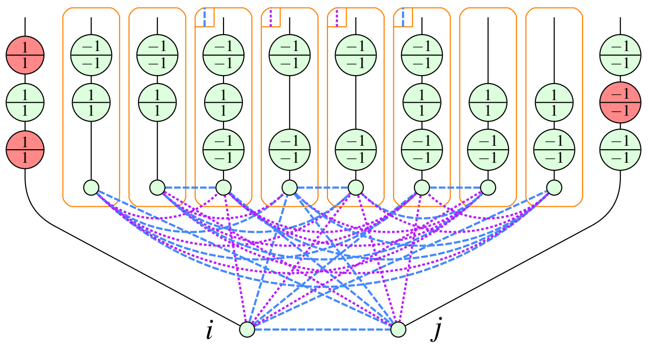
Proof. There are four cases ($a, w_{i,j} \in \{1,2\}$), which split into two pairs of symmetric cases: $a = w_{i,j}$ and $a \neq w_{i,j}$. We show just one case - the 1-pivot along ij of weight 1 - the remaining cases being analogous. We again employ the !-notation (15). An asterisk $*_a$ next to a spider means that at the next step an a-local complementation will be performed at this spider.



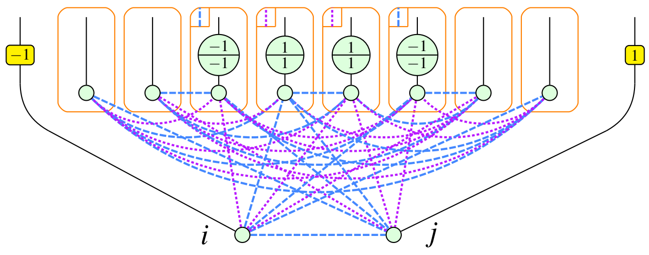




$\overset{3.3}{=}$



$\overset{(\mathbf{f})}{\underset{(34)}{=}}$

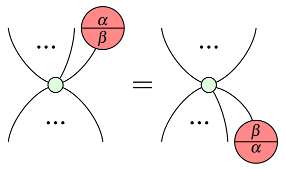


□

D Spider Elimination Rules for Qutrits

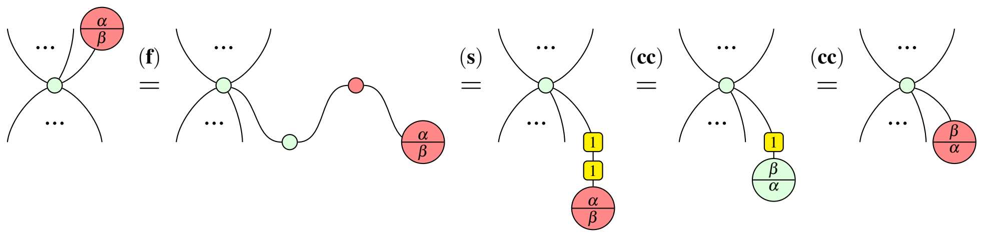
Now we prove the three elimination theorems for $\mathscr{P}$-, $\mathscr{N}$- and $\mathscr{M}$-spiders. All three require the following two lemmas.

Lemma D.1. The following 'leg flip' equation holds in the qutrit ZX-calculus:



(36)

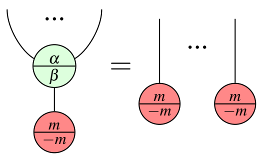
Proof.



(37)

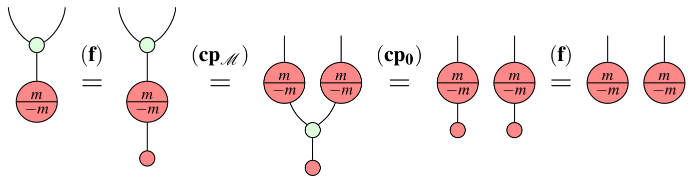
□

Lemma D.2. The following more substantial '$\mathscr{M}$-copy' rule holds in the qutrit ZX-calculus, for any $\mathscr{M}$-spider state (i.e. $m \in \{0,1,2\}$ below):



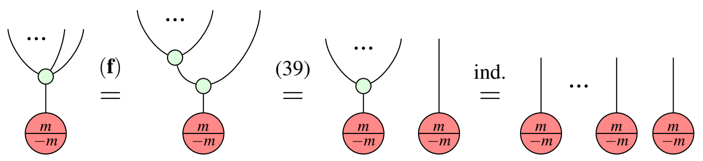
(38)

Proof. First we prove that an $\mathscr{M}$-spider with non-trivial phase (i.e. $m \in \{1,2\}$) satisfies a copy rule exactly like the rule $(\mathbf{cp_0})$:



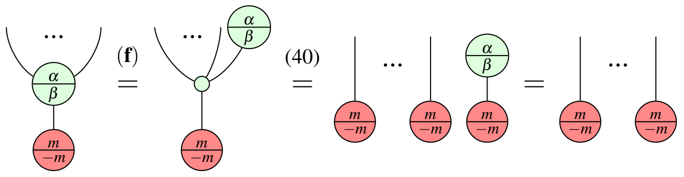
(39)

Then we prove the case $\alpha = \beta = 0$ by induction:



(40)

Which finally allows us to prove the full statement, where the last equality is just dropping the scalar term:



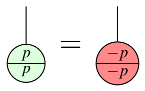
(41)

□

D.1 $\mathscr{P}$-spider Elimination

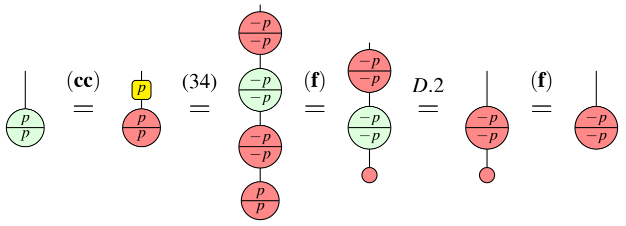
The $\mathscr{P}$-spider elimination theorem additionally requires a lemma allowing us to turn a $\mathscr{P}$-spider state of one colour into a $\mathscr{P}$-spider state of the other. We will use this lemma its adjoint form in the proof of the main theorem.

Lemma D.3. The following rule holds in the qutrit ZX-calculus, for $p \in \{1,2\}$:



(42)

Proof.

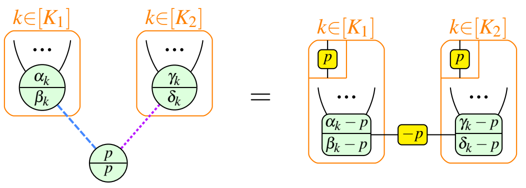


(43)

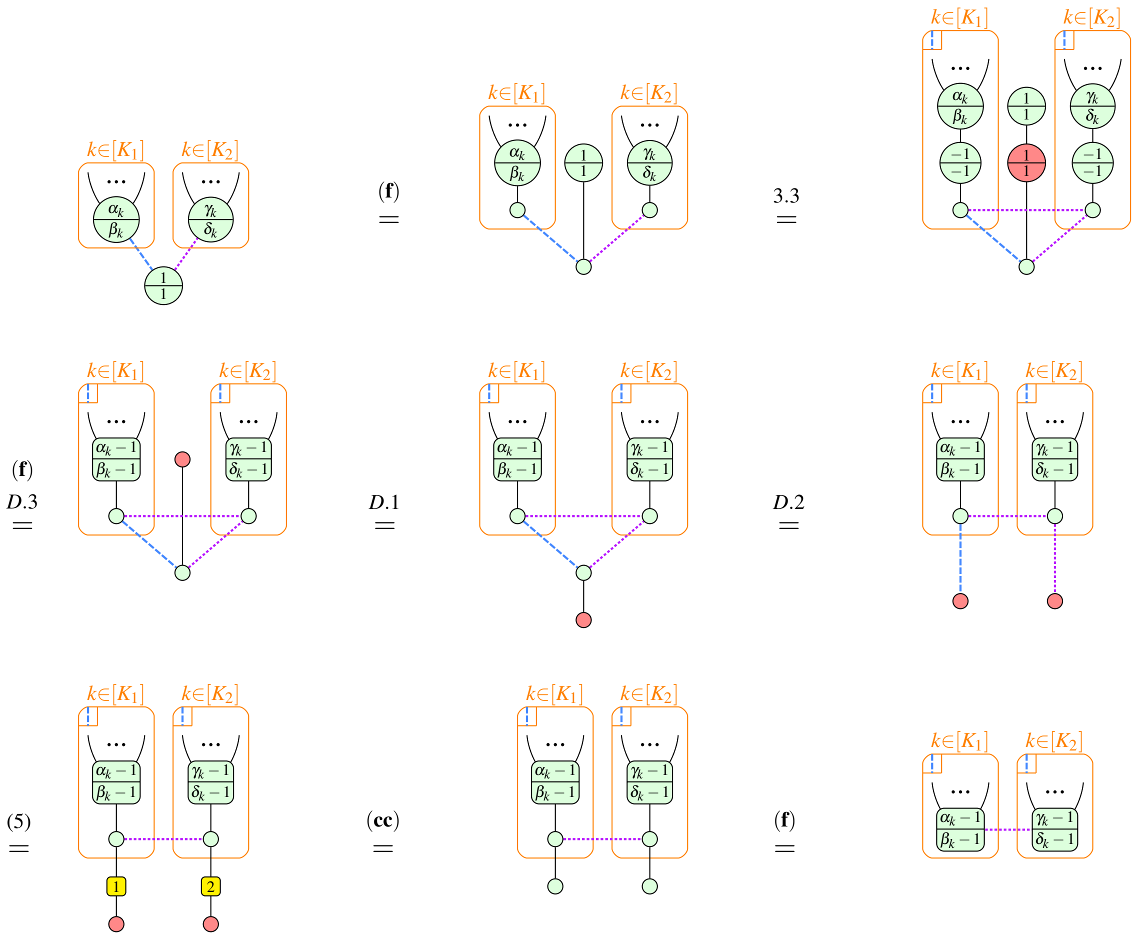
□

Theorem D.4. / Theorem 3.6. Given any graph-like ZX-diagram containing an interior $\mathscr{P}$-spider x with phase $\left(\frac{p}{p}\right)$ for $p \in \{1,2\}$, suppose we perform a p-local complementation at x. Then the new

ZX-diagram is related to the old one by the following equality:


(44)

Proof. For clarity of presentation we only show the case $\frac{p}{p} = \frac{1}{1}$, the other case being near-identical.

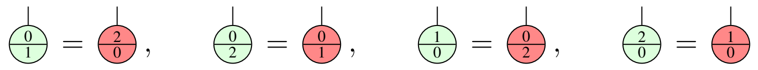


□

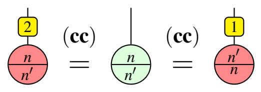
D.2 $\mathcal{N}$-spider Elimination

Proving the corresponding $\mathcal{N}$-spider elimination theorem again requires a lemma allowing us to turn an $\mathcal{N}$-spider state of one colour into an $\mathcal{N}$-spider state of the other.

Lemma D.5. The following rules hold in the qutrit ZX-calculus:

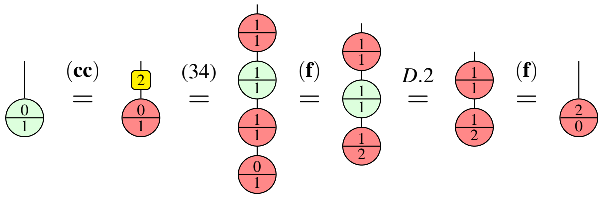

(45)

Proof. For any green $\mathcal{N}$-spider state with phase $\frac{n}{n'}$, we have a choice of two colour change rules which we could use to turn it into a red $\mathcal{N}$-spider state with a H- or $H^{\dagger}$-box on top:



(46)

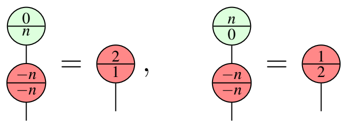
Of these two choices, exactly one has a decomposition of of the H-/$H^{\dagger}$-box as in (34) that allows the bottom two red spiders to fuse into an $\mathcal{M}$-spider, which we can then move past the green spider above it via D.2. For brevity we only show the case $\frac{n}{n'} = \frac{0}{1}$:



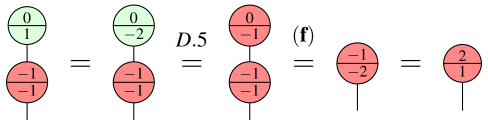
(47)

□

Corollary D.6. The following equations hold in the qutrit ZX-calculus, for $n \in \{1,2\}$:


(48)

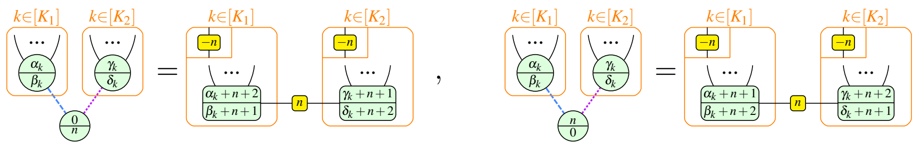
Proof. Again we only prove one case, the other three being analogous. Each case uses D.5 in its adjoint form - recall that the adjoint of a spider is found by swapping inputs and outputs and negating angles.



(49)

□

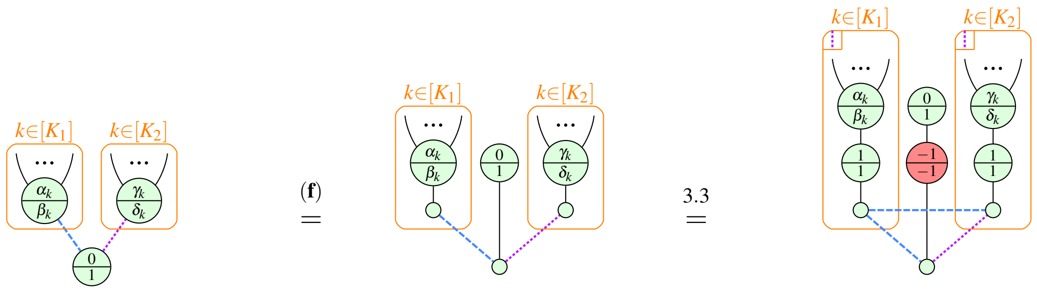
Theorem D.7. / Theorem 3.7. Given any graph-like ZX-diagram containing an interior $\mathcal{N}$-spider x with phase $\frac{0}{n}$ or $\frac{n}{0}$ for $n \in \{1,2\}$, suppose we perform a $(-n)$-local complementation at x. Then, treating the two cases separately, the new ZX-diagrams are related to the old ones by the equalities:

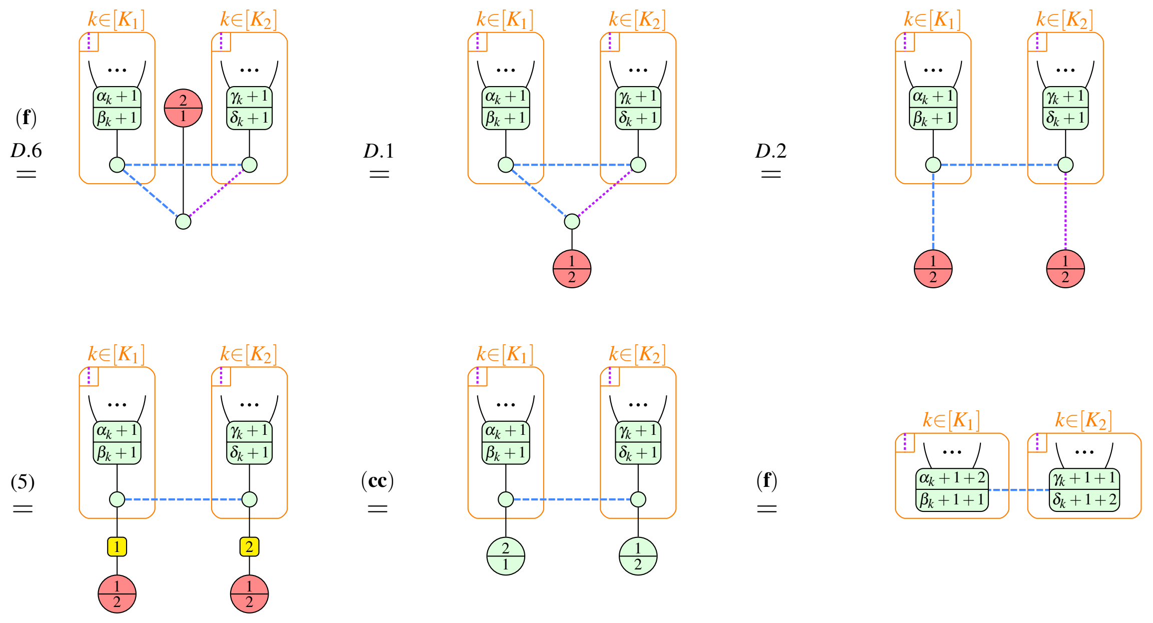


$k \in [K_1]$
$k \in [K_2]$
$-n$
α_k+n+2, β_k+n+1
γ_k+n+1, δ_k+n+2
α_k+n+1, β_k+n+2
γ_k+n+2, δ_k+n+1

(50)

Proof. We prove the case where x has phase $\frac{0}{1}$, the other cases being near-identical.

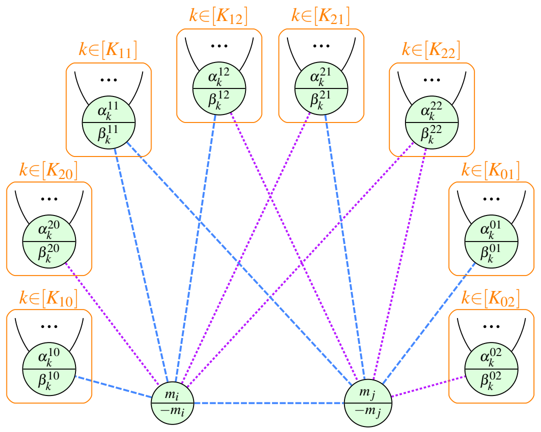




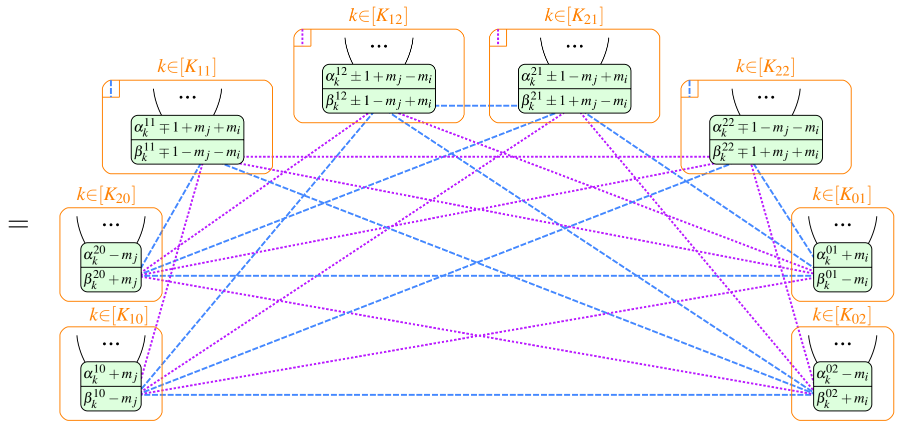
□

D.3 $\mathscr{M}$-spider Elimination

Theorem D.8. / Theorem 3.8. Given any graph-like ZX-diagram containing two interior $\mathscr{M}$-spiders i and j connected by edge ij of weight $w_{i,j} =: w \in \{1,2\}$, suppose we perform a proper $\pm w$-pivot along ij (both choices give the same result). For $w = 1$, the new ZX-diagram is related to the old one by the following equality:

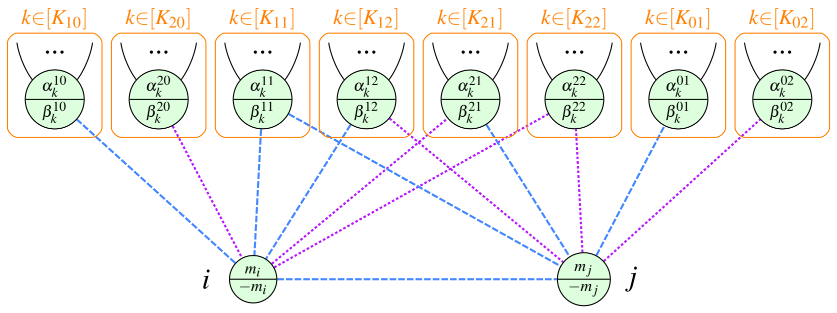


(51)

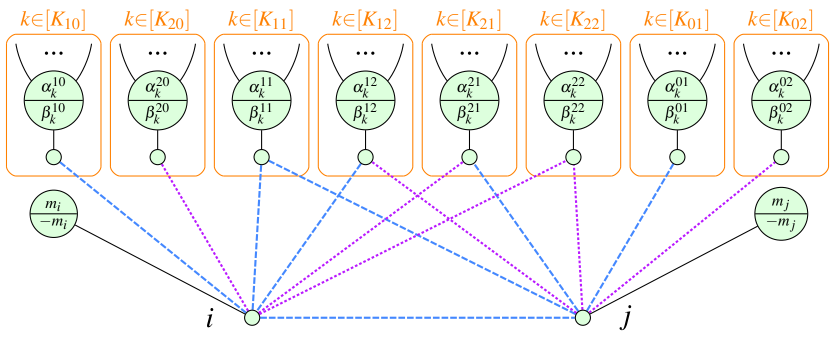


The case $w = 2$ differs only as follows: in the first diagram (the left hand side of the equation), the edge ij is purple (by definition), while in the lower diagram (the right hand side of the equation), a $\pm 2 = \mp 1$ will replace all occurences of ± 1, and the roles of purple and blue will be swapped throughout.

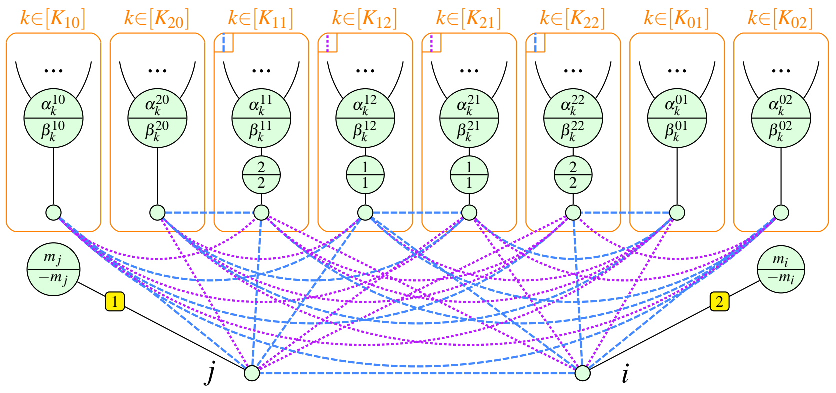
Proof. We show the case where $w_{ij} =: w = 1$, with the case $w = 2$ being completely analogous. We can choose either a proper 1-pivot or a proper 2-pivot; both give the same result. Here we only show the former:



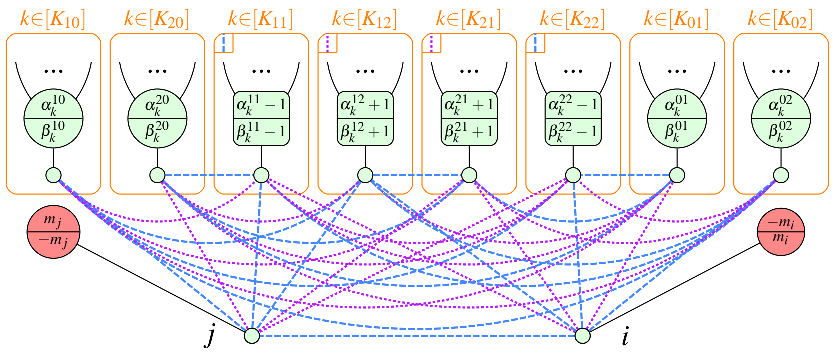
$\overset{(\mathbf{f})}{=}$



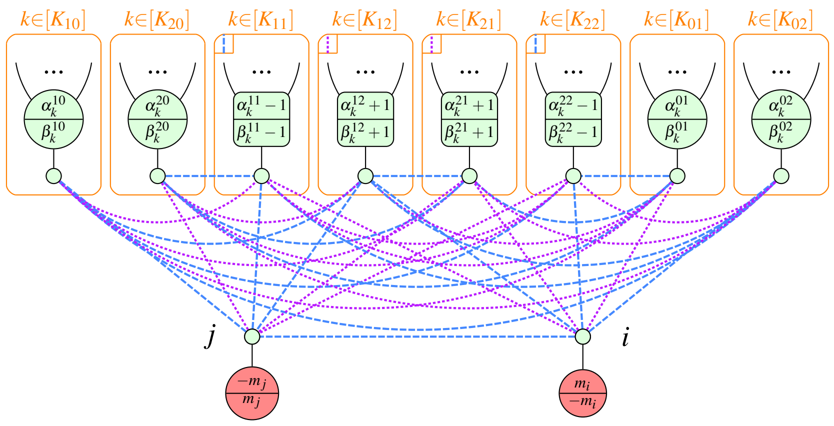
$\overset{C.1}{=}$



$\overset{(\mathbf{cc})}{=}$

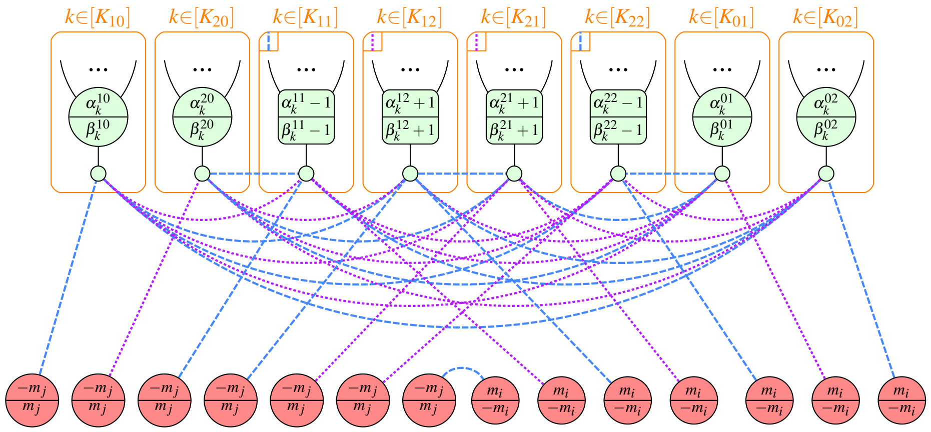


$\overset{D.1}{=}$



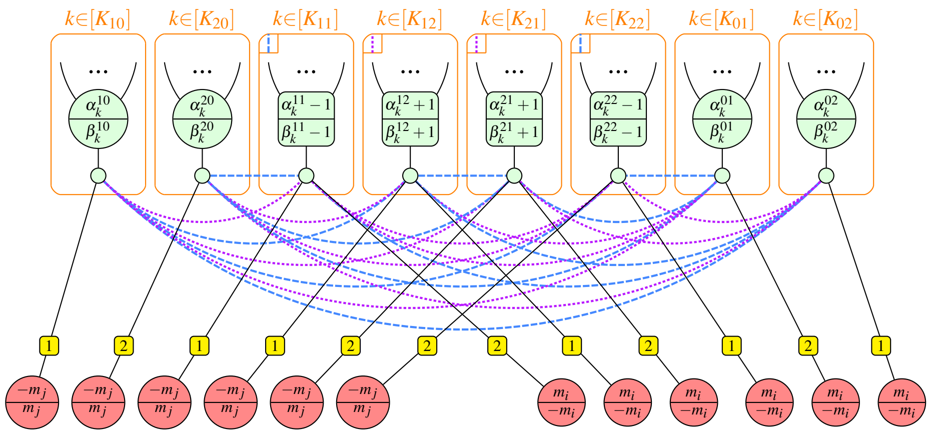
$k\in[K_{10}]$
$k\in[K_{20}]$
$k\in[K_{11}]$
$k\in[K_{12}]$
$k\in[K_{21}]$
$k\in[K_{22}]$
$k\in[K_{01}]$
$k\in[K_{02}]$
α_k^{10} / β_k^{10}
α_k^{20} / β_k^{20}
$\alpha_k^{11}-1$ / $\beta_k^{11}-1$
$\alpha_k^{12}+1$ / $\beta_k^{12}+1$
$\alpha_k^{21}+1$ / $\beta_k^{21}+1$
$\alpha_k^{22}-1$ / $\beta_k^{22}-1$
α_k^{01} / β_k^{01}
α_k^{02} / β_k^{02}
j
i
$-m_j$ / m_j
m_i / $-m_i$

$\overset{D.2}{=}$



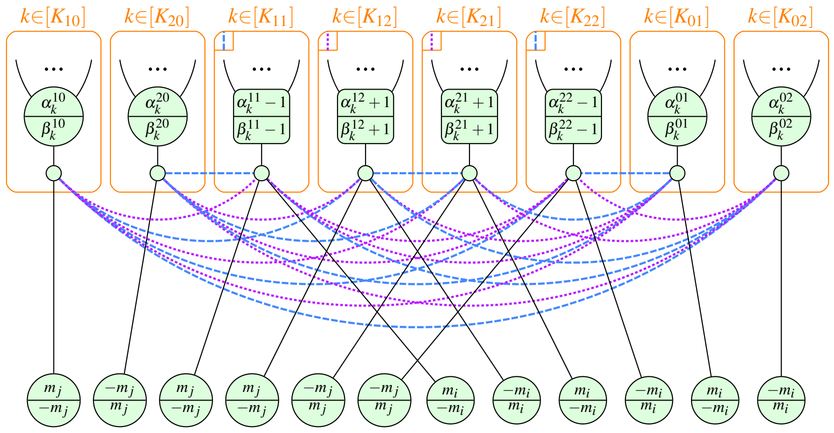
$k\in[K_{10}]$
$k\in[K_{20}]$
$k\in[K_{11}]$
$k\in[K_{12}]$
$k\in[K_{21}]$
$k\in[K_{22}]$
$k\in[K_{01}]$
$k\in[K_{02}]$
α_k^{10} / β_k^{10}
α_k^{20} / β_k^{20}
$\alpha_k^{11}-1$ / $\beta_k^{11}-1$
$\alpha_k^{12}+1$ / $\beta_k^{12}+1$
$\alpha_k^{21}+1$ / $\beta_k^{21}+1$
$\alpha_k^{22}-1$ / $\beta_k^{22}-1$
α_k^{01} / β_k^{01}
α_k^{02} / β_k^{02}
$-m_j$ / m_j
m_i / $-m_i$

$=$

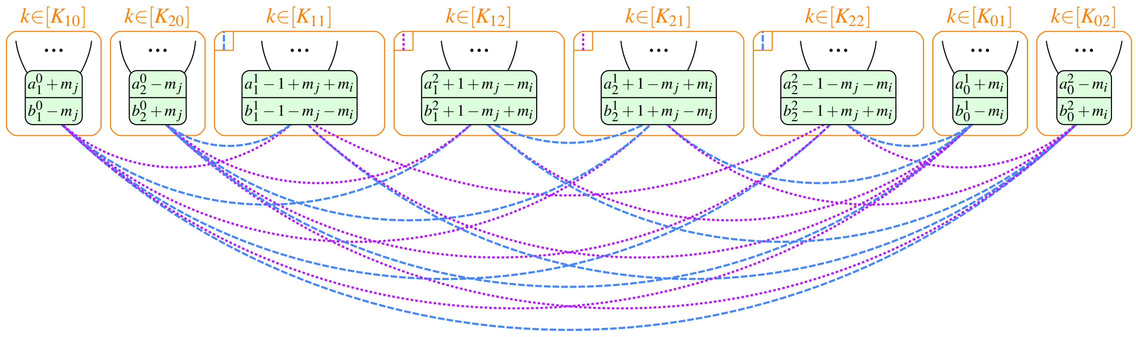


$k\in[K_{10}]$
$k\in[K_{20}]$
$k\in[K_{11}]$
$k\in[K_{12}]$
$k\in[K_{21}]$
$k\in[K_{22}]$
$k\in[K_{01}]$
$k\in[K_{02}]$
α_k^{10} / β_k^{10}
α_k^{20} / β_k^{20}
$\alpha_k^{11}-1$ / $\beta_k^{11}-1$
$\alpha_k^{12}+1$ / $\beta_k^{12}+1$
$\alpha_k^{21}+1$ / $\beta_k^{21}+1$
$\alpha_k^{22}-1$ / $\beta_k^{22}-1$
α_k^{01} / β_k^{01}
α_k^{02} / β_k^{02}
1 2 1 1 2 2 2 1 2 1 2 1
$-m_j$ / m_j
m_i / $-m_i$

$\overset{(\mathbf{cc})}{=}$



$\overset{(\mathbf{f})}{=}$

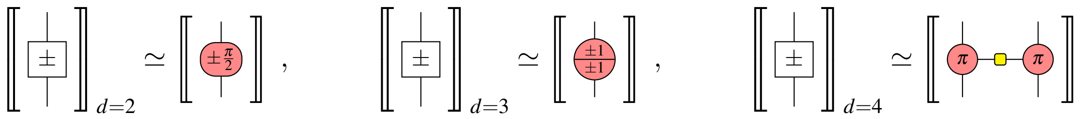

$k\in[K_{10}]$
$k\in[K_{20}]$
$k\in[K_{11}]$
$k\in[K_{12}]$
$k\in[K_{21}]$
$k\in[K_{22}]$
$k\in[K_{01}]$
$k\in[K_{02}]$
$a_1^0+m_j$ / $b_1^0-m_j$
$a_2^0-m_j$ / $b_2^0+m_j$
$a_1^1-1+m_j+m_i$ / $b_1^1-1-m_j-m_i$
$a_1^2+1+m_j-m_i$ / $b_1^2+1-m_j+m_i$
$a_2^1+1-m_j+m_i$ / $b_2^1+1+m_j-m_i$
$a_2^2-1-m_j-m_i$ / $b_2^2-1+m_j+m_i$
$a_0^1+m_i$ / $b_0^1-m_i$
$a_0^2-m_i$ / $b_0^2+m_i$

□

E ±-Boxes in the ZX-Calculus

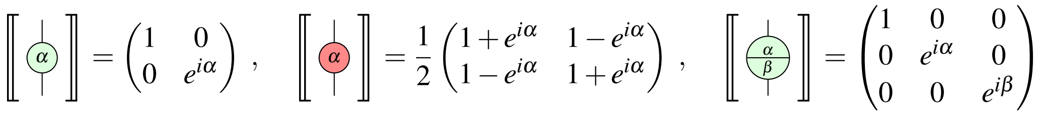
We close by proving that the ±-boxes in (20) correspond to stabilizer ZX-diagrams, while the qutrit X-box in (23) does not.

Proposition E.1. The following equalities hold up to a scalar under the standard interpretation:

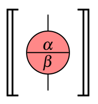


(52)

Proof. Recalling $\omega = e^{i\frac{2\pi}{3}}$, the standard interpretations of phase gates in matrix form are:

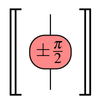


$$= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}, \quad = \frac{1}{2}\begin{pmatrix} 1+e^{i\alpha} & 1-e^{i\alpha} \\ 1-e^{i\alpha} & 1+e^{i\alpha} \end{pmatrix}, \quad = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{i\alpha} & 0 \\ 0 & 0 & e^{i\beta} \end{pmatrix}$$

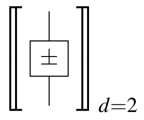


$$= \frac{1}{3}\begin{pmatrix} 1+e^{i\alpha}+e^{i\beta} & 1+\bar{\omega}e^{i\alpha}+\omega e^{i\beta} & 1+\omega e^{i\alpha}+\bar{\omega}e^{i\beta} \\ 1+\omega e^{i\alpha}+\bar{\omega}e^{i\beta} & 1+e^{i\alpha}+e^{i\beta} & 1+\bar{\omega}e^{i\alpha}+\omega e^{i\beta} \\ 1+\bar{\omega}e^{i\alpha}+\omega e^{i\beta} & 1+\omega e^{i\alpha}+\bar{\omega}e^{i\beta} & 1+e^{i\alpha}+e^{i\beta} \end{pmatrix}$$

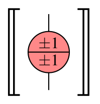
So in the simplest case $d = 2$ it is fairly straightforward to see that:



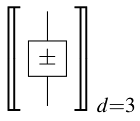
$$= \frac{1}{2}\begin{pmatrix} 1\pm i & 1\mp i \\ 1\mp i & 1\pm i \end{pmatrix} = \frac{\sqrt{2}}{2}e^{\mp i\frac{\pi}{4}}\begin{pmatrix} \pm i & 1 \\ 1 & \pm i \end{pmatrix} = \frac{\sqrt{2}}{2}e^{\mp i\frac{\pi}{4}} \tag{53}$$



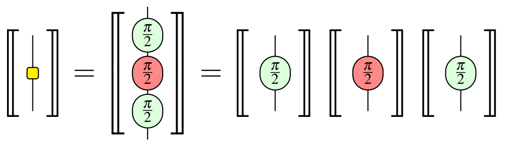
The next case $d = 3$ is proved similarly:



$$\begin{aligned} &= \frac{1}{3}\begin{pmatrix} 1+e^{\pm i\frac{2\pi}{3}}+e^{\pm i\frac{2\pi}{3}} & 1+\bar{\omega}e^{\pm i\frac{2\pi}{3}}+\omega e^{\pm i\frac{2\pi}{3}} & 1+\omega e^{\pm i\frac{2\pi}{3}}+\bar{\omega}e^{\pm i\frac{2\pi}{3}} \\ 1+\omega e^{\pm i\frac{2\pi}{3}}+\bar{\omega}e^{\pm i\frac{2\pi}{3}} & 1+e^{\pm i\frac{2\pi}{3}}+e^{\pm i\frac{2\pi}{3}} & 1+\bar{\omega}e^{\pm i\frac{2\pi}{3}}+\omega e^{\pm i\frac{2\pi}{3}} \\ 1+\bar{\omega}e^{\pm i\frac{2\pi}{3}}+\omega e^{\pm i\frac{2\pi}{3}} & 1+\omega e^{\pm i\frac{2\pi}{3}}+\bar{\omega}e^{\pm i\frac{2\pi}{3}} & 1+e^{\pm i\frac{2\pi}{3}}+e^{\pm i\frac{2\pi}{3}} \end{pmatrix} \\ &= \frac{1}{3}\begin{pmatrix} \sqrt{3}e^{\pm i\frac{\pi}{2}} & \sqrt{3}e^{\mp i\frac{\pi}{6}} & \sqrt{3}e^{\mp i\frac{\pi}{6}} \\ \sqrt{3}e^{\mp i\frac{\pi}{6}} & \sqrt{3}e^{\pm i\frac{\pi}{2}} & \sqrt{3}e^{\mp i\frac{\pi}{6}} \\ \sqrt{3}e^{\mp i\frac{\pi}{6}} & \sqrt{3}e^{\mp i\frac{\pi}{6}} & \sqrt{3}e^{\pm i\frac{\pi}{2}} \end{pmatrix} \\ &= \frac{\sqrt{3}}{3}e^{\mp i\frac{\pi}{6}}\begin{pmatrix} e^{\pm i\frac{2\pi}{3}} & 1 & 1 \\ 1 & e^{\pm i\frac{2\pi}{3}} & 1 \\ 1 & 1 & e^{\pm i\frac{2\pi}{3}} \end{pmatrix} \\ &= \frac{\sqrt{3}}{3}e^{\mp i\frac{\pi}{6}} \end{aligned} \tag{54}$$

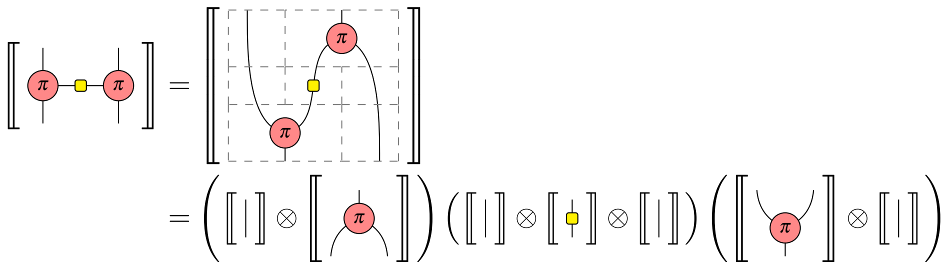


For the other qubit case $d = 4$ we first note:



$$= \frac{1}{2\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{55}$$

Then using the standard interpretation for spiders as in (1), we decompose the diagram in such a way that we can apply the standard interpretation:



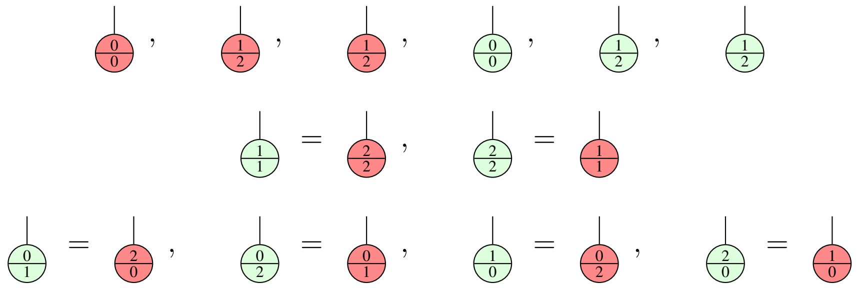
$$= \frac{\sqrt{2}}{8}\begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}$$

$$= \frac{\sqrt{2}}{8}\left[\!\left[\ \boxed{\pm}\ \right]\!\right]_{d=4}$$

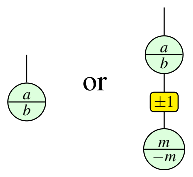
□

Now we prove the qutrit X-box is not a stabilizer diagram. The following lemma is required:

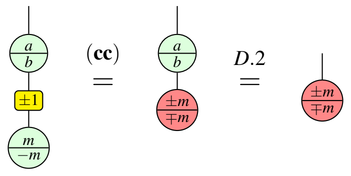
Lemma E.2. Every stabilizer state is equivalent to one of the following 12 diagrams:



Proof. We can use our elimination rules to prove this. A stabilizer state is a diagram with no inputs and one output, in which every phase component is a multiple of $\frac{2\pi}{3}$. After putting it in graph-like form (via Proposition 3.1), all but one spider will be interior. We'll call the non-interior one the *boundary* spider. Our elimination rules say we can remove all interior $\mathscr{P}$- and $\mathscr{N}$-spiders, plus all pairs of connected interior $\mathscr{M}$-spiders. So applying these rules until we can do so no more, we have two cases. The easiest case is where end up with just the single boundary spider, which must have no inputs and one output. In the other case we get a single $\mathscr{M}$-spider connected to a boundary spider. The $\mathscr{M}$-spider can have no other legs, and the boundary spider must have exactly one other leg, which is the output of the overall diagram:


(56)

In the first case, we're done. In the second case, we need only note:

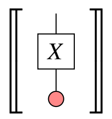


□

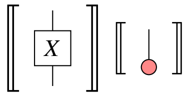
Proposition E.3. The qutrit X-box defined below is not a stabilizer diagram:

$$\left[\!\left[\ \boxed{X}\ \right]\!\right] = \sum_{i,j=0}^{d-1} (1-\delta_{ij})\,|i\rangle\langle j| = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Proof. Stabilizer diagrams are closed under composition (they form a group). Hence if the X-box is stabilizer, it must send the phaseless red spider state to a stabilizer state. This has matrix:

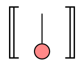


$$=$$

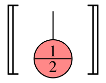


$$= \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

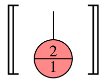
But, up to a scalar, this is not equal to any of the 12 stabilizer states above:



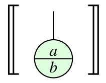
$$= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$



$$= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$



$$= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$



$$= \begin{pmatrix} 1 \\ \omega^a \\ \omega^b \end{pmatrix}$$

□