Classifying Complexity with the ZX Calculus

Alex Townsend-Teague [0,1] and Konstantinos Meichanetzidis [1,2]
[0] Mathematical Institute, University of Oxford
[1] Department of Computer Science, University of Oxford
[2] Cambridge Quantum Computing Ltd.

The ZX calculus is a graphical language which allows for reasoning about suitably represented tensor networks, aka ZX diagrams, in terms of rewrite rules. Here, we focus on problems which amount to exactly computing a scalar encoded as a closed tensor network. In general, such problems are #P-hard. However, there are families of such problems which are known to be in P when the dimension is bellow a certain value. By expressing problem-instances from these families as ZX diagrams, we see that the easy instances belong to the stabiliser fragment of the ZX calculus. Building on previous work on efficient simplification of qubit stabiliser diagrams, we present simplifying rewrites for the case of qutrits. Finally, we look at the specific examples of evaluating the Jones polynomial and of counting graph-colourings. Our exposition further constitutes the ZX calculus as a suitable and unifying language for studying the complexity of a broad range of classical or quantum problems.

1 Introduction

A plethora of hard problems of interest to physics and computer science regard interacting many-body multi-state systems, classical or quantum, and reduce to *exactly* computing a single scalar. These range from probability amplitudes in quantum mechanics, partition functions in classical statistical mechanics, counting problems, probabilistic inference, and many more. Problem instances can be encoded as a *closed* tensor-networks over an appropriate (semi)ring. The scalar can be evaluated by *full tensor contraction*, which in general is #P-hard [7]. In fact, finding the optimal contraction path for a general tensor network is NP-hard.

The ZX calculus is a graphical language whose origin lies in the field of quantum foundations [6]. It allows for reasoning about ZX diagrams, i.e. tensor networks which are expressed in terms of primitive generators [11]. The generators have a concrete representation as tensors over a (semi)ring and they obey a set of rewrite rules which respect semantics [10]. Importantly, the rewrite rules *depend* on the (semi)ring over which the diagram is interpreted as a tensor network. If a scalar of interest is expressed as a ZX diagram, one can rewrite the diagram by applying rewrite rules. One's goal is then to apply a sequence of rewrites and perform *full diagram simplification* in order to compute the desired scalar. 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 [9].

In the context of quantum computing, the Gottesman-Knill theorem states that stabiliser (or Clifford) circuits can be simulated *efficiently* on classical computers [1]. By simulation here we mean the *exact* computation of *amplitudes*. These circuits are can be expressed by the *stabiliser* fragment of the ZX calculus. An alternative, and in fact more general, proof of the Gottesman-Knill theorem for qubits has been obtained graphically in terms of the qubit ZX calculus. Interestingly, even if the motivation for studying stabiliser diagrams stems from quantum computation, the result that stabiliser ZX diagrams can be simplified efficiently has broader applicability. This is because ZX diagrams can express arbitrary

Submitted to:
QPL 2021

© Alex Townsend-Teague, Konstantinos Meichanetzidis
This work is licensed under the
Creative Commons Attribution License.

linear maps; not every ZX diagram is a quantum circuit, but every quantum circuit can be cast as a ZX diagram.

Beyond quantum computing, ZX diagrams have been leveraged to study the complexity of boolean satisfiability (SAT) and model counting (#SAT) [5]. Model counting entails computing the number of satisfying assignments to a boolean formula (which is given in conjunctive normal form) and this number can be expressed a closed diagram. Specifically, one finds that in the case of XORSAT and #XORSAT, which are known to be tractable, the corresponding ZX diagrams representing the problems exactly correspond to qubit stabilizer diagrams, which are efficient to simplify. Going further, the ZH calculus [3], an extension of the ZX calculus, can be imported to make graphically obvious why 2SAT is in P but #2SAT is #P-complete while 3SAT is NP-complete and #3SAT is #P-hard. That is, the counting version of 2SAT is hard while deciding is easy, but this is not the case for 3SAT where both deciding and counting are hard. The *key realisation* is in that the decision problem is a counting problem but where the diagram is interpreted over the Boolean semiring. Then the *rewrite rules change accordingly* and obviously imply the above result by allowing efficient simplification strategies.

Such observations make apparent the *unifying power of diagrammatic reasoning for studying the complexity* of problem families of universal interest, where the degrees of freedom are two-dimensional. Building on the spirit of [5], we treat ZX as a library comprising out-of-the-box and readily-available diagrammatic rewrite-rules from which we import the appropriate tools for the job depending on the occasion. The key rewrite rules that enable the efficient diagram simplification of qubit stabiliser ZX diagrams are called *local complementation* and *pivot*, the latter being composition of three of the former [8]. In this work, we recall the qutrit version of local complementation and derive the corresponding pivot rule which implies that qutrit stabiliser diagrams can be simplified efficiently. We then 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, bellow which the tensor network corresponds to a stabiliser ZX diagram.

We underline that throughout this work, all rewrites are valid *up to scalar* since we are only concerned about making statements about complexity.

2 Simplifying Qubit ZX-Diagrams

In this section we briefly review the ZX calculus and recalling the definitions of a graph-like diagrams and stabiliser diagrams. We also recall the simplifying rewrites that enable the efficient simplification of stabiliser diagrams.

2.1 The Qubit ZX-Calculus

Here we give a quick introduction to the qubit ZX-calculus. The qubit ZX-calculus is a diagrammatic language for quantum mechanics. The generators are tensors called *spiders*. The spider legs, or *wires*, representing vector spaces of dimension $d = 2$, can be connected to form networks which are again valid ZX diagrams. ZX diagrams are part of a formal language. Diagrams are read bottom-to-top; bottom open wires (not connected to anything) are *input wires* and top ones are *output*. Diagrams with only outputs are called *states* and with only inputs are called *effects*. A *closed* diagram is one with no input nor output wires and it represents a scalar.

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 concrete representation of the spider generators as tensors over $\mathbb{C}$ is:

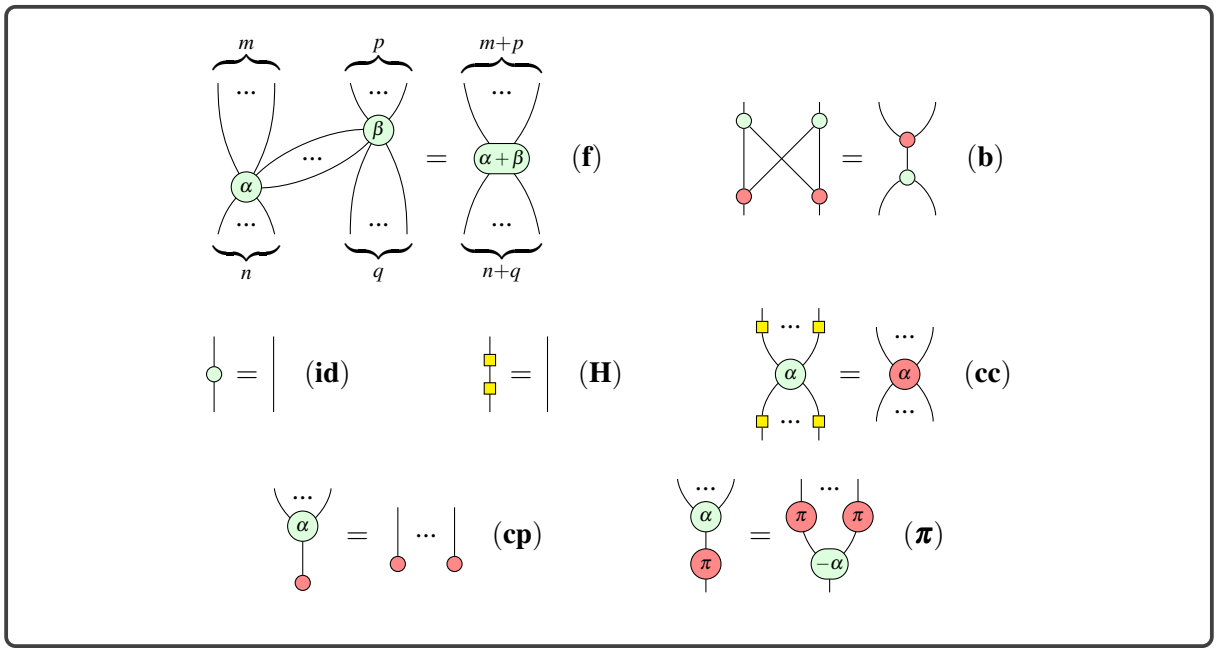


Figure 1: Rewrite rules for the qubit ZX-calculus where spiders are interpreted as tensors over $\mathbb{C}$.

$$\left[\!\!\left[\; \right]\!\!\right] = |0\rangle^m \langle 0|^n + e^{i\alpha} |1\rangle^m \langle 1|^n , \qquad \left[\!\!\left[\; \right]\!\!\right] = |+\rangle^m \langle +|^n + e^{i\alpha} |-\rangle^m \langle -|^n$$

where $\{|0\rangle, |1\rangle\}$ is the *Z-basis* and $\{|+\rangle, |-\rangle\}$ the *X-basis* in $\mathbb{C}^2$.

The Hadamard gate H is denoted as a yellow box. Often we will instead draw a dashed blue line to represent a *Hadamard edge*:

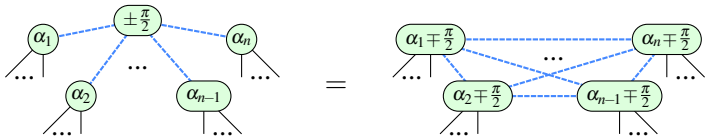
$$= \quad = \quad , \qquad \left[\!\!\left[\; \right]\!\!\right] \simeq |0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1| \tag{1}$$

The rewrite rules of the ZX calculus, shown in Fig.1, allow manipulation of the diagrams by local tensor-rewrites. Importantly, the rewrites preserve the semantics, i.e. the concrete tensor representation over $\mathbb{C}$, up to a scalar.

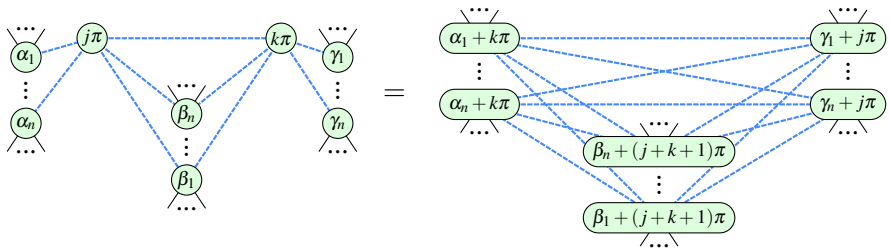
The *stabiliser fragment* of the calculus consists of all diagrams in which all phases are $a = \kappa\frac{\pi}{2}$, $\kappa \in \mathbb{Z}$. In (Theorem 5.4 [8]) the authors give an efficient algorithm for simplifying any *qubit* ZX-diagram to an equivalent diagram with fewer spiders. The algorithm consists of consecutive applications of spider-eliminating rewrites.

First it is shown that 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. Then the following two

rewrite rules, derived via *local complementation* and *pivoting*, can be used to eliminate spiders. For more details, see Section 4 [8].

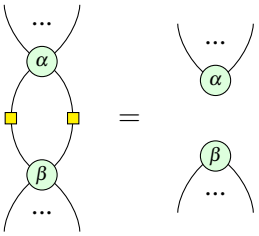


(2)



(3)

Eq.2 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.3 says we can remove any pair of spiders with phases in $\{0,\pi\}$ connected by a Hadamard edge at the cost of performing a local pivot along said edge. After each application of (2) or (3), the following rewrite rule, aka the Hopf rule which can be derived from the qubit ZX rules, can be used to remove any parallel Hadamard edges and ensure the diagram remains graph-like: **what about self-loops? also, make the edges blue dashed.**



(4)

In particular, and importantly to this work, given a closed stabilizer diagram the algorithm *efficiently simplifies* it until it contains at most one spider. If it exists, this spider is green, legless and has phase in $\{0,\pi\}$. The point relevant to this work is that spiders can be eliminated efficiently, i.e. with a polynomial number of rewrites in the initial number of spiders. Note that every spider elimination introduces a polynomial number of edges in the diagram, which prevents an overwhelming memory cost of the simplification procedure.

3 Simplifying Qutrit ZX-Diagrams

For the remaining case $q = 3$, we will show that we can generalise all the ideas of the previous section to the qutrit ZX-calculus, which we must first define.

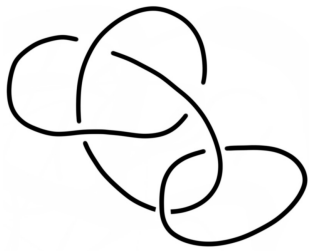
4 Case studies

4.1 Jones Polynomial at Lattice Roots of Unity

For $d = 2,3,4$ it's easy. For $d \geq 5$ it's hard.

Mention Potts and also anyon stuff but the important thing is the tensornet.

A knot K is a circle embedded in $\mathbb{R}^3$. A set of knots tangled together is a *link* L. A link L is represented as the projection of the link on $\mathbb{R}^2$ but retaining the information of over or under crossings. For example, the trefoil knot linked with an unknot can be drawn as:



(5)

We say that $L \simeq L'$ if there is a sequence of Reidemeister moves from L to L', i.e. reversible local strand deformations that do not change the topology, for example by cutting or gluing strands (see Appendix).

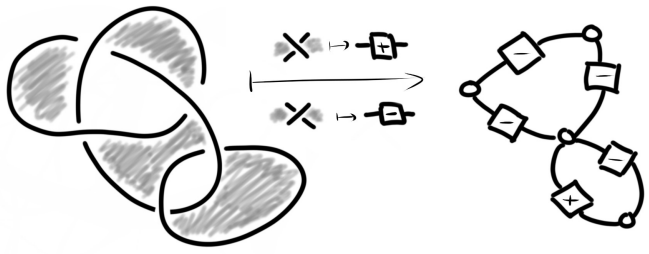
The Jones polynomial $V_L(t)$ is a Laurent polynomial in $t \in \mathbb{C}$ and is a link *invariant*. This means that $V_L(t) \neq V_{L'}(t) \Rightarrow L \not\simeq L'$, where $L \simeq L'$.

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 where a skein relation is used iteratively on every crossing [KauffmanBook]. Exactly evaluating the Jones polynomial at specific points $t \in \mathbb{C}$ is #P-hard. *except* at the *lattice roots of unity* $\pm 1, \pm i, \pm e^{i2\pi/3}, \pm e^{i4\pi/3}$ where it can be evaluated efficiently (in time $O(poly(|C|))$) [Welsh].

In the context of quantum computation, additively approximating the Jones polynomial at non-lattice roots of unity is the paradigmatic BQP-complete complete problem [aharonov]. For the lattice roots of unity, the quantum amplitude to which the problem reduces can be computed efficiently. Specifically, the quantum circuits that need to be simulated are stabiliser circuits which are known to be classically tractable [ref? Kuperberg? Aharonov?]. From the more natural point of view of topological quantum computation, the jones polynomial at roots of unity corresponds to a quantum amplitude in the fusion space of non-abelian anyons. Computation is performed by initialising, braiding, and the anyons. Specifically, in the case of $\mathrm{SU}(2)_k$ anyons, the Jones polynomial is evaluated at $e^{i\frac{2\pi}{2+k}}$ [Rowel-Wang]. Note the correspondence $q \in \{1,2,3,4\} \Leftrightarrow k \in 1,2,4$. This is consistent with the fact that topological quantum computation with $\mathrm{SU}(2)_2$ anyons (Ising) or $\mathrm{SU}(2)_4$ anyons is not universal (unless the $\mathrm{SU}(2)_4$ anyons are augmented by fusion and measurements in which case they are capable of universal quantum computation [1504.02098]).

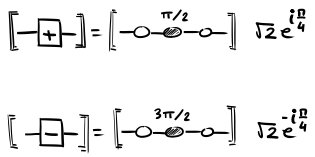
Remarkably, the Jones polynomial can be expressed in terms of the partition function of a q-state Potts model [Wu]. The Potts model is defined on a signed graph, called the Tait graph, which is obtained as follows. The link diagram is bicoloured (checkerboard style) and every coloured area is mapped to a vertex and every crossing is mapped to a signed edge according its orientation relative to the surrounding colours. Every vertex is assigned a q-state classical spin and every signed edge indicates a spin-spin Tait-sign-dependent interaction $J_\pm \in \mathbb{C}$ which relate to the Jones polynomial's variable as $e^{J_\pm} = -t^{\mp}$. The role of the link invariant is played by the partition function $Z(q)$; in fact, the interactions $J_\pm$ are such so that the graph operations corresponding to Reidemeister moves leave $Z(q)$ invariant. Multiplying with an efficiently computatble prefactor $\mathscr{A}(t) = (-t^{\frac{1}{2}} - t^{-\frac{1}{2}})^{(-|V|-1)}(-t^{\frac{3}{4}})^{w} t^{\frac{1}{4}\tau}$ for bookkeeping of twist factors, the relation to the Jones polynomial is $V_L(t) = \mathscr{A}(t)Z(q)$. The link shown in Eq.5 returns the following

tensor network.



(6)

For $q = 2$ in ZX notation:



(7)

We provide a graphical proof that evaluating the Jones polynomial of arbitrary links at the lattice roots of unity is in P. For these cases, the problem reduces to the evaluation of the partition function of a planar q-state Potts model. The partition is represented as a closed tensor network. We employ the ZX-calculus, which is a sound and complete graphical language for tensor networks and allows reasoning via graphical rewrites. We show that there exist polynomially long rewrite sequences that fully simplify the tensor network and return $Z(q)$ for $q \in \{1,2,3,4\}$, which correspond to evaluating the Jones polynomial at the lattice roots of unity.

Remark 4.1. A small technical remark is in order: the rule set given here is complete for qubit stabilizer quantum mechanics when equality is taken only up to a scalar factor. To achieve completeness under exact equality, a slighly modified rule set would be required, as in (for example) [2]. But for our purposes this would be overkill; in order to show that the Jones polynomial of knots at lattice roots of unity is efficiently computable, it suffices to consider the 'non-exact' rules - i.e. it suffices to show such a Jones polynomial is efficiently computable up to a scalar factor. But of course to actually compute such a Jones polynomial, we will need to keep track of scalars.

We can now give ZX-diagrams whose standard interpretations are equal (up to a scalar) to the matrices $T_{\pm}^{(q)}$ in (**??**) for $q \in \{2,4\}$.

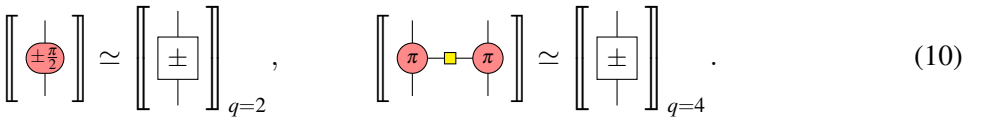
For spiders S and T, we then have:

$$[\![S \otimes T]\!] = [\![S]\!] \otimes [\![T]\!] \tag{8}$$

$$[\![S \circ T]\!] = [\![S]\!]\,[\![T]\!] \tag{9}$$

where on the right hand side of the first equation the $\otimes$ symbol means the Kronecker product of two matrices. Thus a diagram with standard interpretation $T_{2\pm}$ will have one input and one output, while a diagram for $T_{4\pm}$ will have two inputs and two outputs.

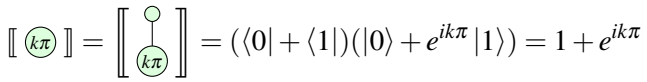
Proposition 4.2. Under the standard interpretation as linear maps, the following diagrams give (up to a scalar) the required matrices:



(10)

Proof. See Appendix A.1. □

Since any ZX-diagram derived from a knot in the manner described in Section **??** will be closed, this algorithm suffices to prove that the calculation of the Jones polynomial of any knot at the lattice roots of unity ± 1 and $\pm i$ is efficient. This is because reading off the scalar at the end is trivial; either we have the empty diagram, which has standard interpretation 1, or a single legless Z-spider with phase $k\pi$:



$$= (\langle 0| + \langle 1|)(|0\rangle + e^{ik\pi}|1\rangle) = 1 + e^{ik\pi} \tag{11}$$

Furthermore, keeping track of the scalar factors introduced with each application of Theorem **??** or Lemma **??** can be done efficiently. [ToDo: proof, if not explicitly doing all this in a scalar-exact fashion. Reference Backens]

4.2 Graph Colouring

5 Outlook

This also motivates further work in circuit extraction for qutrit circuits so that one can attempt graph-theoretic simplification with the qutrit ZX calculus, analogous to the case of qubits [8, 4].

Future work, scalar exact version of the qutrit rewrite rules, for concrete applications to problems where 3-state systems are involved. Further, generalisation of local complementation and pivot rewrites to arbitrary prime dimension. Software implementations a la pyzx could be extended accordingly.

6 Acknowledgments

Teague?l KM wishes to thank Niel de Beaudrap, Aleks Kissinger, Stefanos Kourtis, and Quanlong Wang for inspiring and helpful discussions. KM acknowledges financial support from the Royal Commission for the Exhibition of 1851 through a research fellowship.

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] Miriam Backens (2015): *Making the stabilizer ZX-calculus complete for scalars*.

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

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

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

[6] 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`.

[7] 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`.

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

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

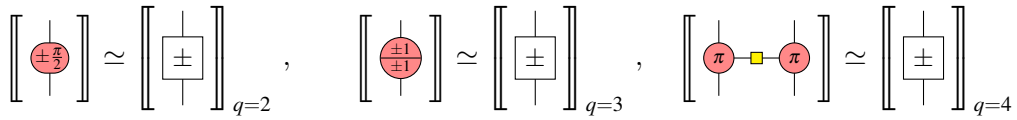
[10] Quanlong Wang (2020): *Completeness of algebraic ZX-calculus over arbitrary commutative rings and semirings*.

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

A Appendix

We begin with proofs of the translations of the matrix $T_{\pm}^{(q)}$ into the ZX-calculus.

Proposition A.1. / Propositions 4.2, ??. The following equalities hold up to a scalar under the standard interpretation:

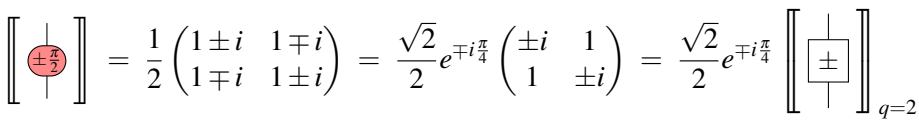


Proof. Recalling $\omega = e^{i\frac{2\pi}{3}}$, the standard interpretations of phase gates in matrix form are:

$$\left[\!\left[\alpha \right]\!\right] = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}, \quad \left[\!\left[\alpha \right]\!\right] = \frac{1}{2}\begin{pmatrix} 1+e^{i\alpha} & 1-e^{i\alpha} \\ 1-e^{i\alpha} & 1+e^{i\alpha} \end{pmatrix}, \quad \left[\!\left[\begin{matrix} \alpha \\ \beta \end{matrix} \right]\!\right] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{i\alpha} & 0 \\ 0 & 0 & e^{i\beta} \end{pmatrix}$$

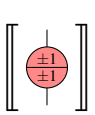
$$\left[\!\left[\begin{matrix} \alpha \\ \beta \end{matrix} \right]\!\right] = \frac{1}{3}\begin{pmatrix} 1+e^{i\alpha}+e^{i\beta} & 1+\bar{\omega}e^{i\alpha}+\omega e^{i\beta} & 1+\omega e^{i\alpha}+\bar{\omega}e^{i\beta} \\ 1+\omega e^{i\alpha}+\bar{\omega}e^{i\beta} & 1+e^{i\alpha}+e^{i\beta} & 1+\bar{\omega}e^{i\alpha}+\omega e^{i\beta} \\ 1+\bar{\omega}e^{i\alpha}+\omega e^{i\beta} & 1+\omega e^{i\alpha}+\bar{\omega}e^{i\beta} & 1+e^{i\alpha}+e^{i\beta} \end{pmatrix}$$

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

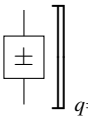


(12)

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

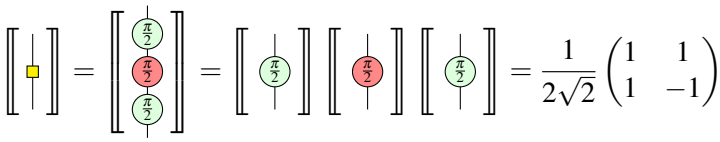


$$
\begin{aligned}
&= \frac{1}{3}\begin{pmatrix} 1+e^{\pm i\frac{2\pi}{3}}+e^{\pm i\frac{2\pi}{3}} & 1+\bar{\omega}e^{\pm i\frac{2\pi}{3}}+\omega e^{\pm i\frac{2\pi}{3}} & 1+\omega e^{\pm i\frac{2\pi}{3}}+\bar{\omega}e^{\pm i\frac{2\pi}{3}} \\ 1+\omega e^{\pm i\frac{2\pi}{3}}+\bar{\omega}e^{\pm i\frac{2\pi}{3}} & 1+e^{\pm i\frac{2\pi}{3}}+e^{\pm i\frac{2\pi}{3}} & 1+\bar{\omega}e^{\pm i\frac{2\pi}{3}}+\omega e^{\pm i\frac{2\pi}{3}} \\ 1+\bar{\omega}e^{\pm i\frac{2\pi}{3}}+\omega e^{\pm i\frac{2\pi}{3}} & 1+\omega e^{\pm i\frac{2\pi}{3}}+\bar{\omega}e^{\pm i\frac{2\pi}{3}} & 1+e^{\pm i\frac{2\pi}{3}}+e^{\pm i\frac{2\pi}{3}} \end{pmatrix} \\
&= \frac{1}{3}\begin{pmatrix} \sqrt{3}e^{\pm i\frac{\pi}{2}} & \sqrt{3}e^{\mp i\frac{\pi}{6}} & \sqrt{3}e^{\mp i\frac{\pi}{6}} \\ \sqrt{3}e^{\mp i\frac{\pi}{6}} & \sqrt{3}e^{\pm i\frac{\pi}{2}} & \sqrt{3}e^{\mp i\frac{\pi}{6}} \\ \sqrt{3}e^{\mp i\frac{\pi}{6}} & \sqrt{3}e^{\mp i\frac{\pi}{6}} & \sqrt{3}e^{\pm i\frac{\pi}{2}} \end{pmatrix} \\
&= \frac{\sqrt{3}}{3}e^{\mp i\frac{\pi}{6}}\begin{pmatrix} e^{\pm i\frac{2\pi}{3}} & 1 & 1 \\ 1 & e^{\pm i\frac{2\pi}{3}} & 1 \\ 1 & 1 & e^{\pm i\frac{2\pi}{3}} \end{pmatrix} \\
&= \frac{\sqrt{3}}{3}e^{\mp i\frac{\pi}{6}}
\end{aligned}
\qquad (13)
$$



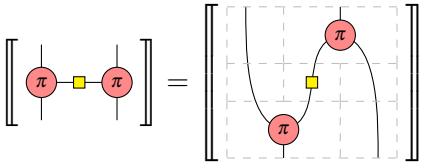
$q=3$

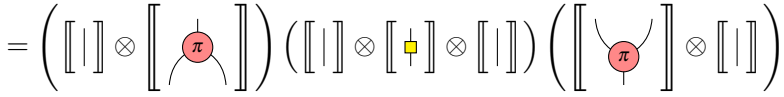
For the other qubit case $q = 4$ we first note:



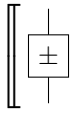
$$= \frac{1}{2\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad (14)$$

Then using the standard interpretation for spiders (Definition **??**) we decompose the diagram in such a way that we can apply the standard interpretation:





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

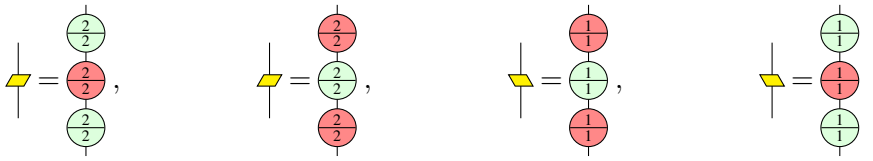


$q=4$

□

Next we prove the local pivot equality from Theorem **??**. For this, recall that all but the (**cm**) and (**cc**) qutrit rewrite rules hold under taking adjoints, and with the roles of green and red swapped, so in

particular rule (**E**) also gives:

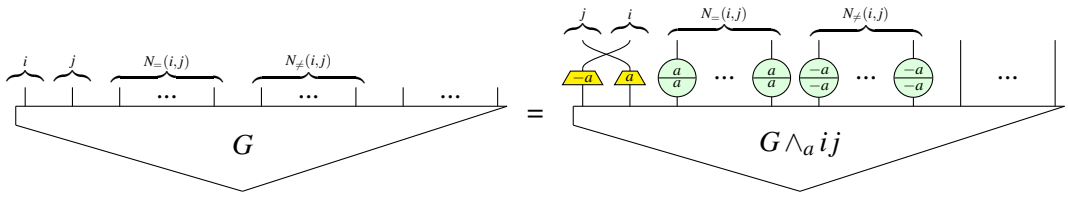


(15)

Theorem A.2. / Theorem ??. Given $a \in \mathbb{Z}_3$ and a graph state (G,W) containing connected nodes i and j, define the following:

- $N_{=}(i,j) := \{k \in N(i) \cap N(j) \mid w_{k,i} = w_{k,j}\}$
- $N_{\neq}(i,j) := \{k \in N(i) \cap N(j) \mid w_{k,i} \neq w_{k,j}\}$

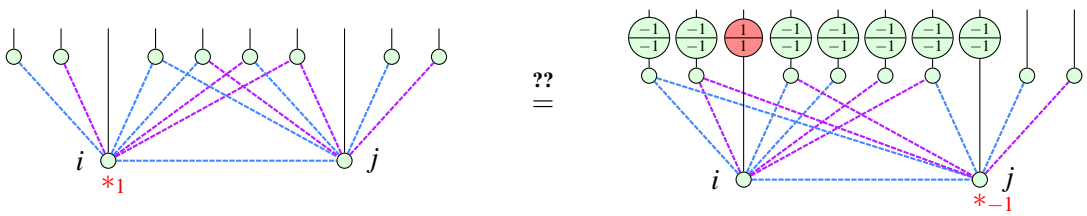
Then the following equation relates G and its proper a-local pivot along ij:

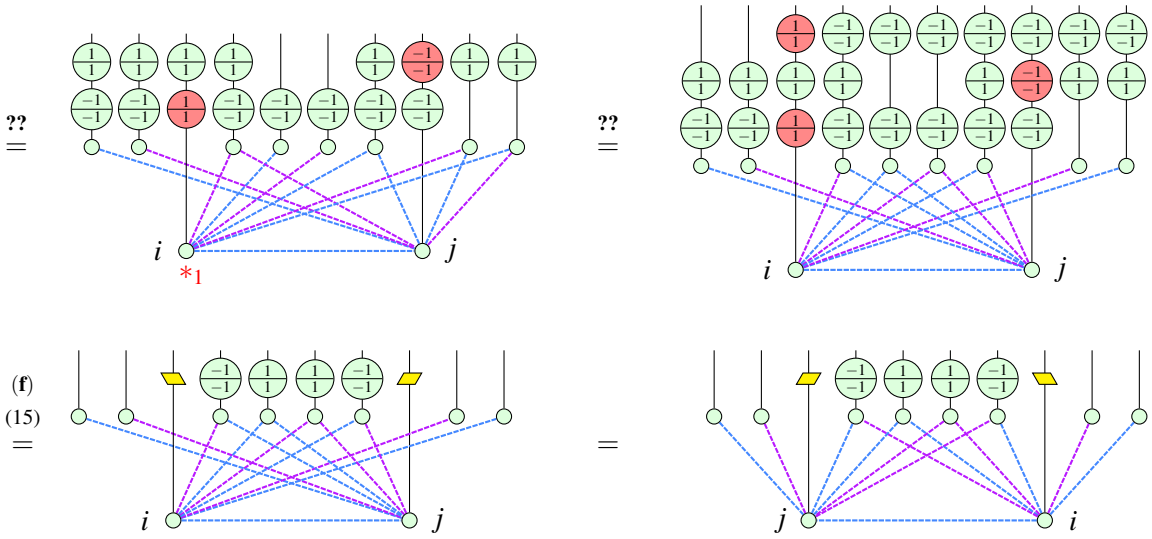


Proof. Annoyingly, proving this in full generality in one go - i.e. for a proper a-local pivot along an edge ij of weight b - becomes a bit tricky diagramatically, because it becomes hard to keep track of all the variable edge weights. Fortunately the four cases ($a,b \in \{1,2\}$) split into two pairs of symmetric cases: $a = b$ and $a \neq b$.

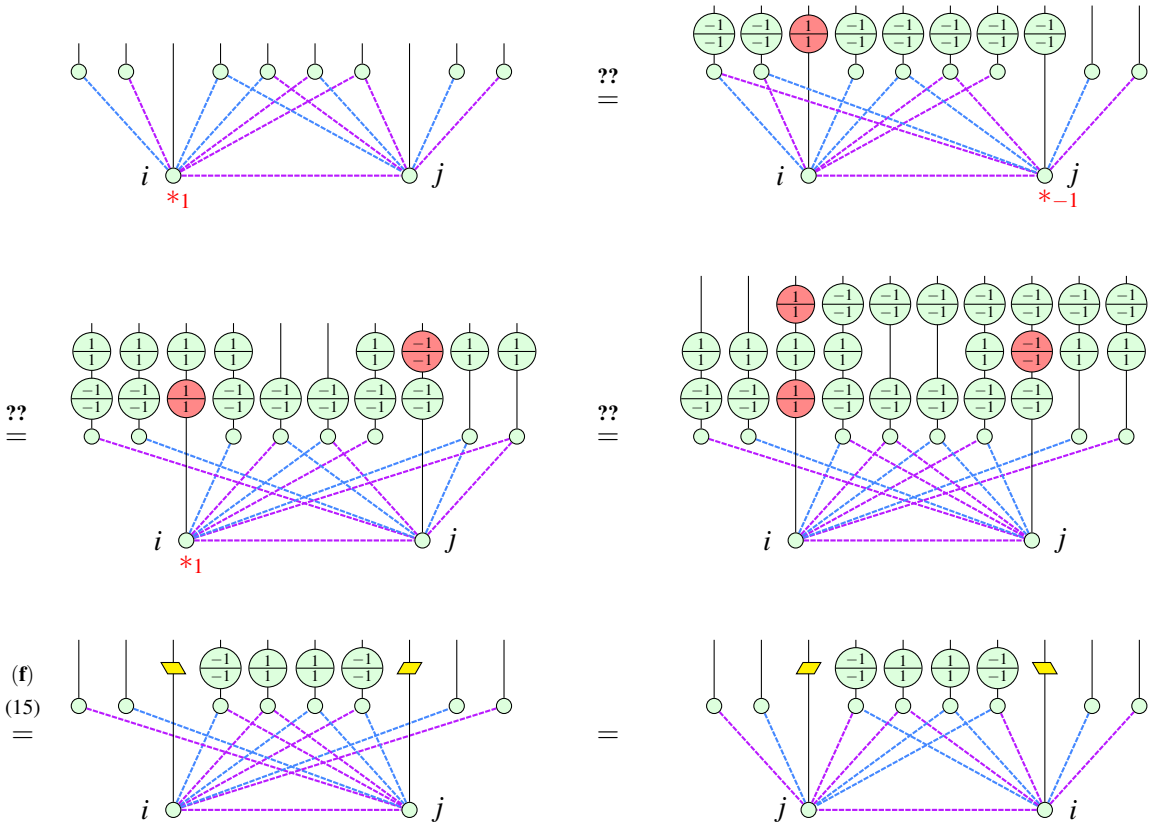
Now, it suffices to only draw a fragment of a graph state. Certainly we consider nodes i and j and the edge ij between them. Then define N_x^y to be the set $\{k \mid w_{k,i} = x, w_{k,j} = y\}$, for $x,y, \in \mathbb{Z}_3$. We will consider a representative node k_x^y from each $N_x^y \neq N_0^0$, as well as its edges ik_x^y and jk_x^y. All other nodes and edges are irrelevant; this is because we are only interested in nodes and edges that *affect* the three local complementation operations on i and j - we aren't concerned with those that are only *affected by* the operations.

So for the case $a = b$, we show the proper 1-local pivot along ij of weight 1:





The story is similar for the case $a \neq b$; here we show the proper 1-local pivot along ij of weight 2:



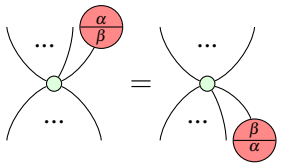
The case $a = b = 2$ is the same as $a = b = 1$, except with the roles of purple and blue edges inter-

changed, so by symmetry of the diagram the only difference is in the phase gates added to the outputs. Namely, on each phase gate we replace all instances of 1 with a 2. Thus the roles of H and $H^\dagger$ are swapped too. Likewise for the case $(a,b) = (2,1)$ with respect to the case $(a,b) = (1,2)$.

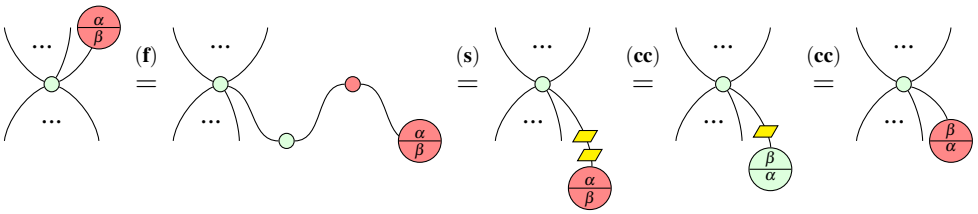
□

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

Lemma A.3. The following 'leg flip' equation holds in the qutrit ZX-calculus. Moreover, it holds with the roles of green and red swapped.

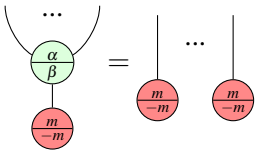


Proof.

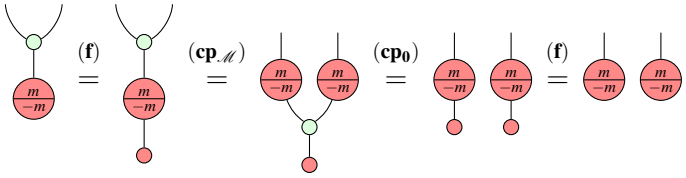


□

Lemma A.4. 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). Moreover, it holds with the roles of green and red swapped.

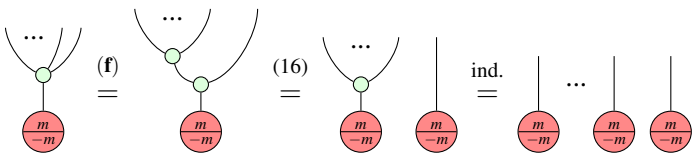


Proof. First we prove that an $\mathscr{M}$-spider with non-trivial phase (i.e. $m \in \{1,2\}$) satisfies a copy rule exactly like the rule $(\mathbf{cp_0})$:



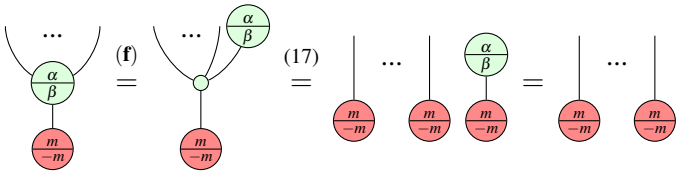
(16)

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



(17)

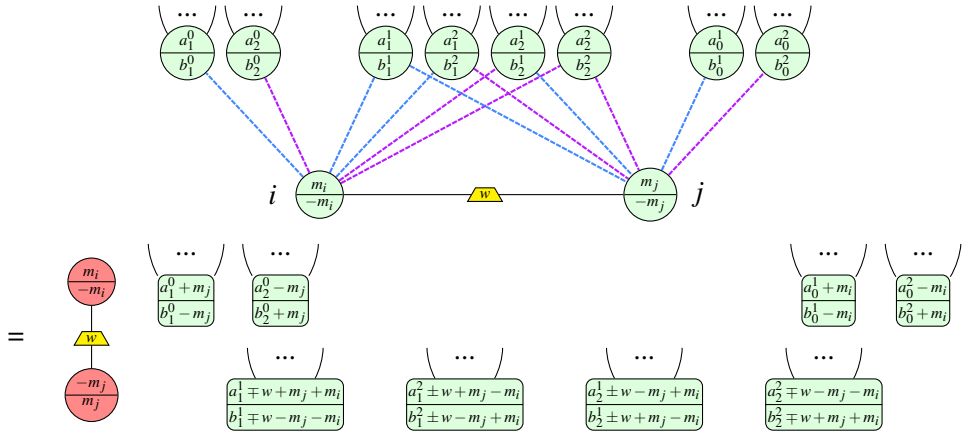
Which finally allows us to prove the full statement, where the last equality is just dropping the scalar term:



□

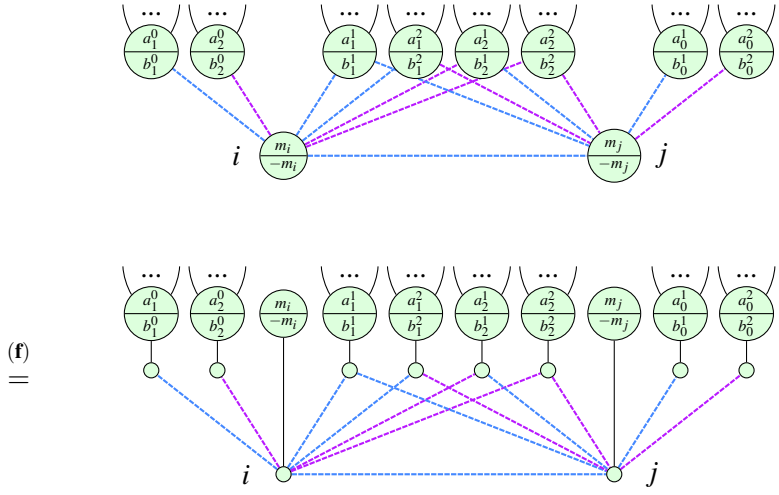
We are now in a position to prove the $\mathscr{M}$-spider elimination theorem.

Theorem A.5. / Theorem ??. 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 $\pm w$-local pivot along ij. Then the new ZX-diagram is related to the old one by the equality:

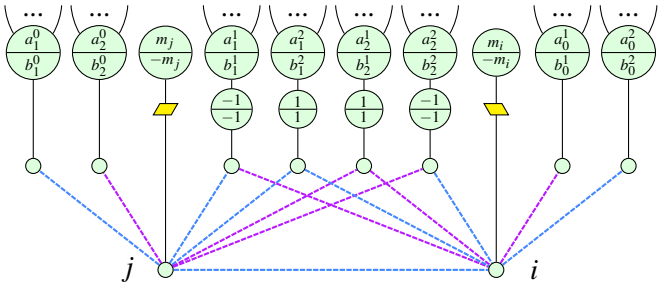


where all changes to weights of edges where neither endpoint is i or j are omitted. Furthermore, in order to save space, each node with phase $\frac{a_x^y}{b_x^y}$ is representative of *all* nodes connected to i by an x-weighted edge and to j by a y-weighted edge.

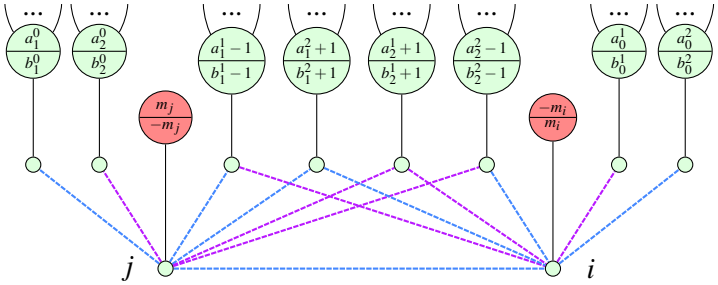
Proof. We show the case where $w_{ij} =: w = 1$, with the case $w = 2$ being completely analogous. We can choose either a proper 1-local pivot or a proper 2-local pivot; both give the same result. Here we only show the former:



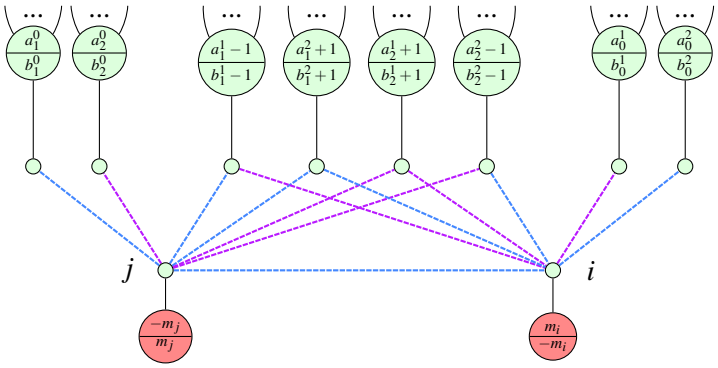
A.2
=



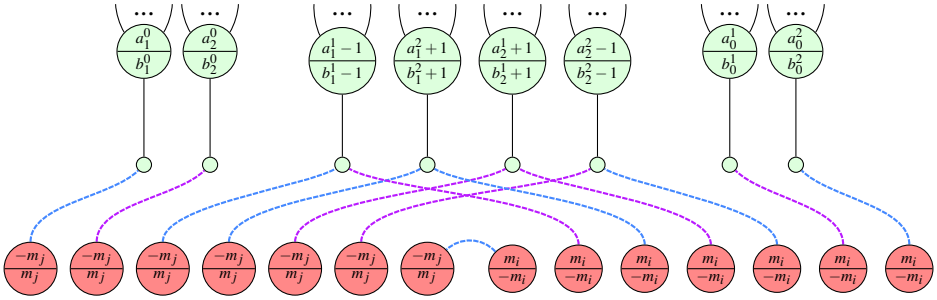
(**cc**)
=

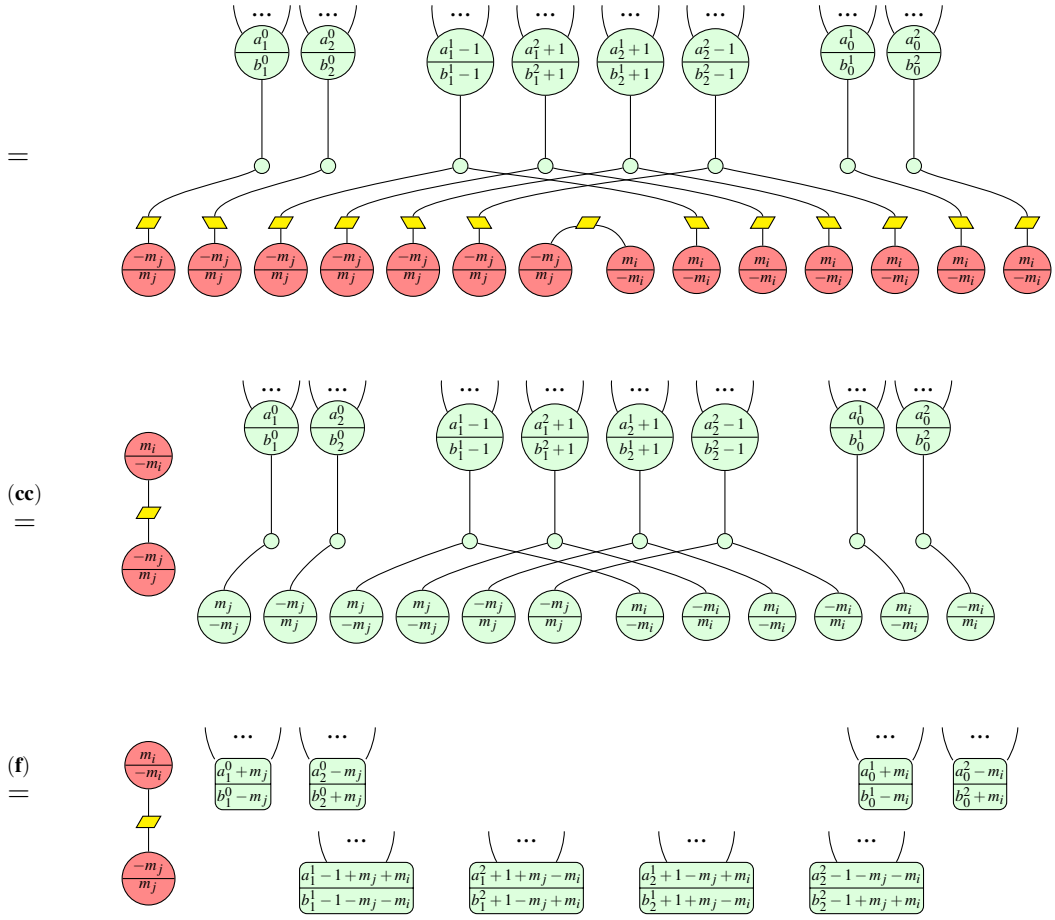


A.3
=



A.4
=

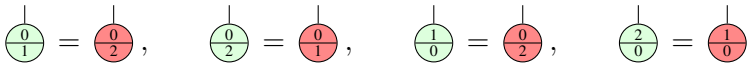




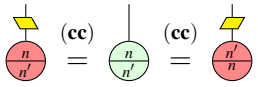
□

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

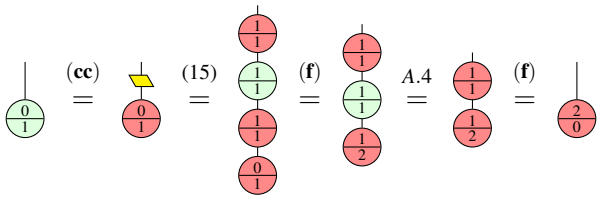
Lemma A.6. The following rules hold in the qutrit ZX-calculus:



Proof. The key observation is that for any green $\mathcal{N}$-spider state with phase $\left(\begin{smallmatrix} n \\ n' \end{smallmatrix}\right)$, 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:

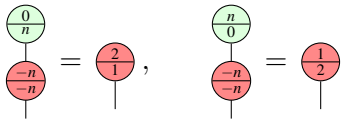


Of these two choices, exactly one has a decomposition of of the H-/$H^\dagger$-box as in (15) that allows the bottom two red spiders to fuse into an $\mathcal{M}$-spider, which we can then move past the green spider above it via A.4. For brevity we only show the case $\left(\begin{smallmatrix} n \\ n' \end{smallmatrix}\right) = \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$:

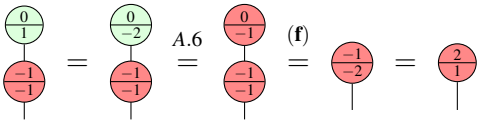


□

Corollary A.7. The following equations hold in the qutrit ZX-calculus:

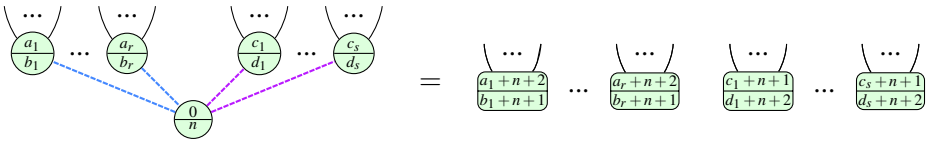


Proof. Again we only prove one case, the rest being analogous. Each case uses A.6 in its adjoint form - recall that the adjoint of a spider is found by swapping inputs and outputs and negating angles.

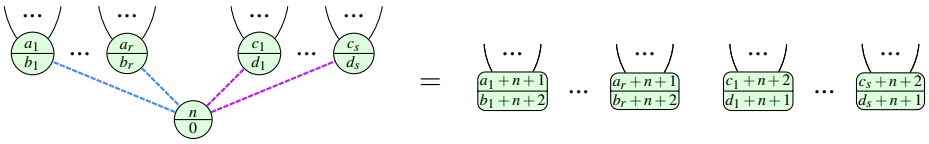


□

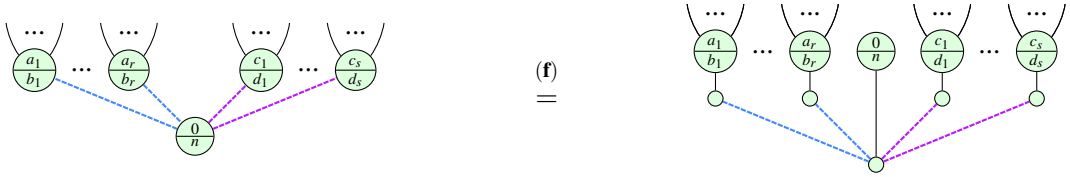
Theorem A.8. / Theorem ??. Given any graph-like ZX-diagram containing an interior $\mathcal{N}$-spider k with phase $\frac{0}{n}$ for $n \in \{1,2\}$, suppose we perform a $(-n)$-local complementation at k. Then the new ZX-diagram is related to the old one by the equality:

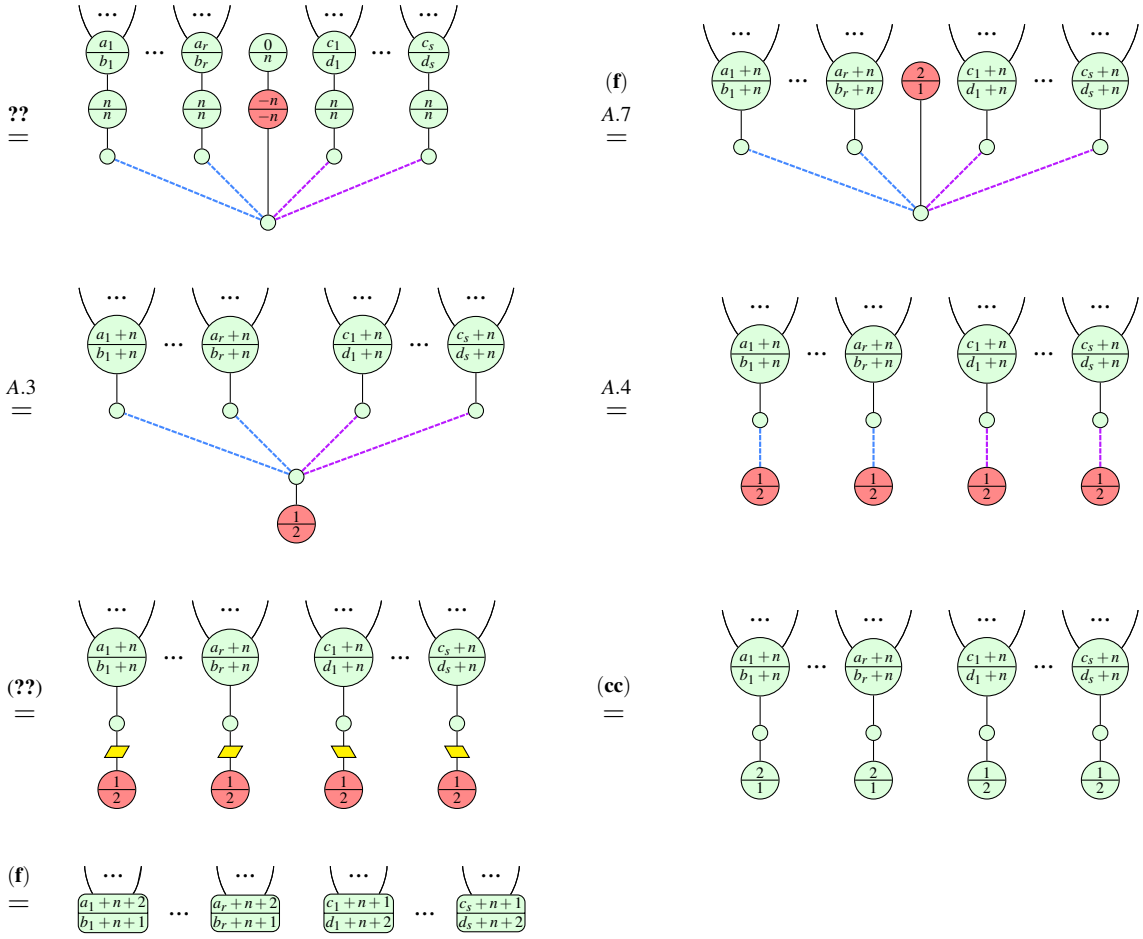


where all changes to weights of edges where neither endpoint is k are omitted. If instead k has phase $\frac{n}{0}$ for $n \in \{1,2\}$, suppose we perform the same $(-n)$-local complementation at k. Then the equality relating the new and old diagrams becomes:



Proof. We prove the case where k has phase $\frac{0}{n}$ for $n \in \{1,2\}$, the other case being near-identical.

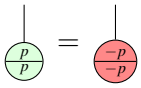




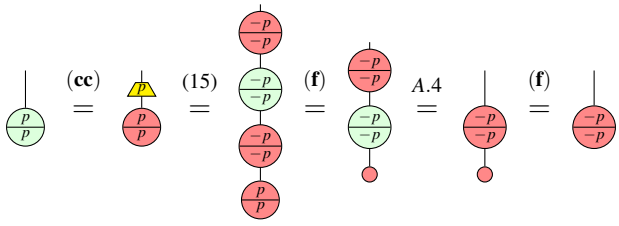
□

Similarly the corresponding $\mathscr{P}$-spider elimination theorem requires a lemma allowing us to turn a $\mathscr{P}$-spider state of one colour into a $\mathscr{P}$-spider state of the other. As above, we will use this lemma its adjoint form in the proof of the main theorem.

Lemma A.9. The following rule holds in the qutrit ZX-calculus:

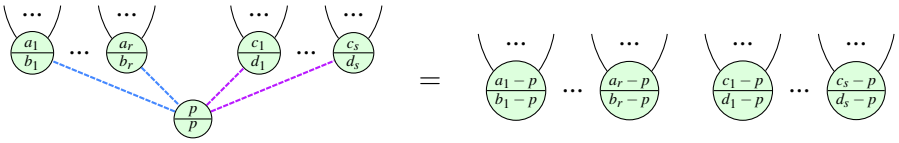


Proof. The proof structure is exactly as in A.6, only for $\mathscr{P}$-spiders it's even simpler:



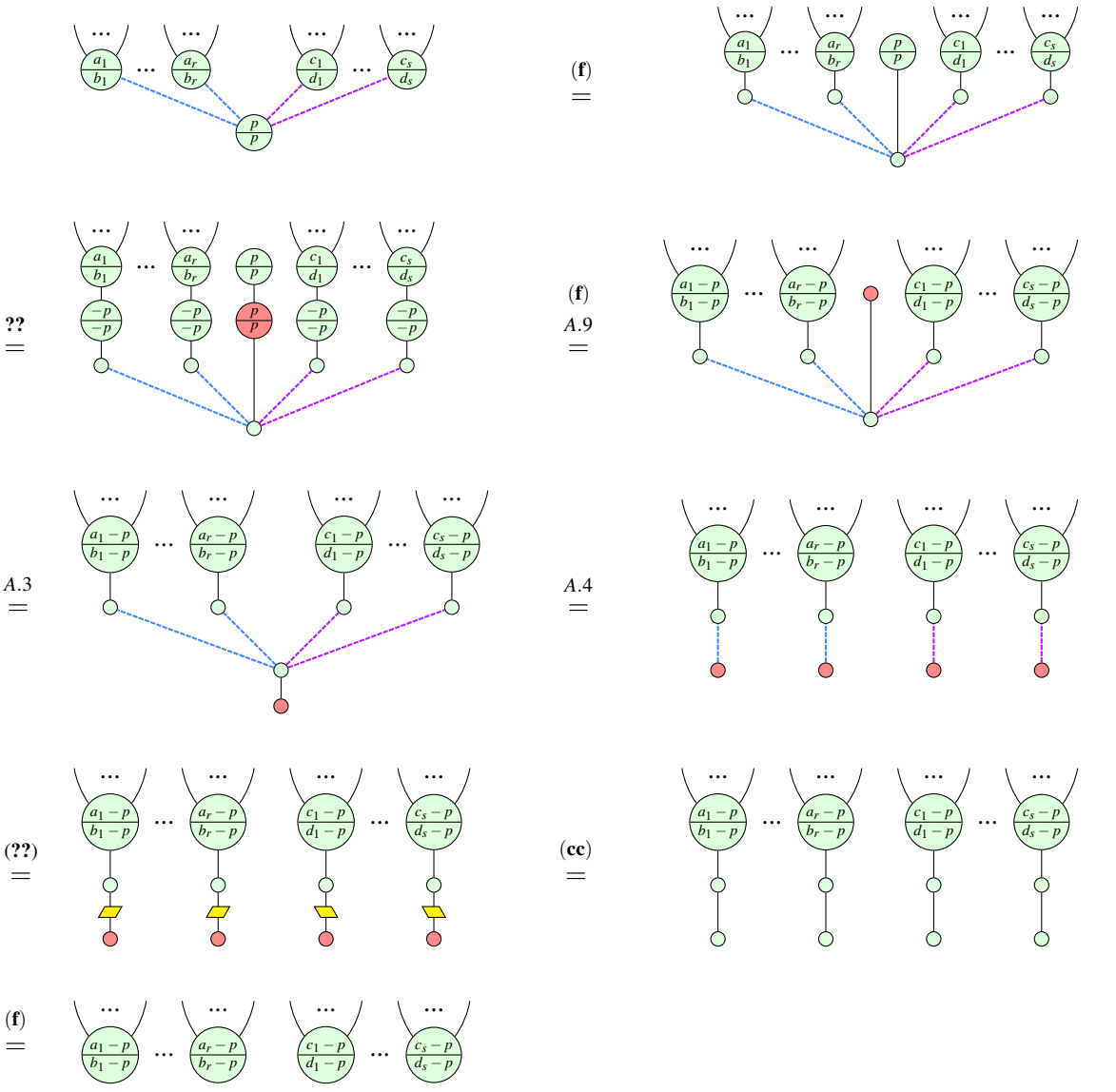
□

Theorem A.10. / Theorem ??. Given any graph-like ZX-diagram containing an interior $\mathscr{P}$-spider k with phase $\frac{p}{p}$ for $p \in \{1,2\}$, suppose we perform a p-local complementation at k. Then the new ZX-diagram is related to the old one by the equality:



where all changes to weights of edges where neither endpoint is k are omitted.

Proof.



□