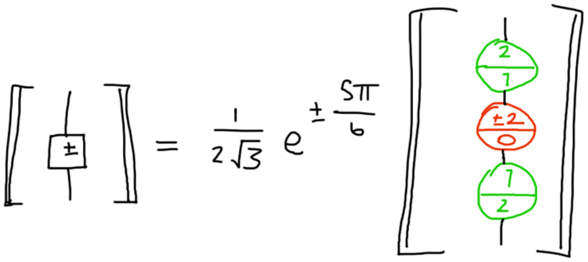
Efficiently Calculating Jones Polynomials at Lattice Roots of Unity with the ZX-Calculus (Draft)

Alex Townsend-Teague   Konstantinos Meichandzitis

1 Introduction

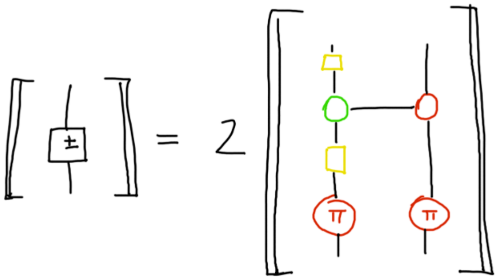
[Use Konstantinos' O.G. notes from the start of this project, and introduce the qubit and qutrit ZX calculi]

For the case $q = 3$ we use the qutrit ZX calculus:



(1)

For the case $q = 4$ we are back again in the usual qubit ZX calculus:



(2)

In each case, the resulting map is in the stabilizer fragment of the ZX calculus.

2 Simplifying Qubit ZX Diagrams

In [cite Aleks' paper] the authors give an efficient algorithm for reducing any qubit stabilizer ZX diagram to one of vastly smaller size [specific bound?]. In our case, since any knot will be transformed by the process above into a ZX diagram with zero inputs and zero outputs, this gives an efficient algorithm for reducing the diagram to a scalar.

[Summarise Aleks' algorithm]

This suffices to prove our main result for the cases $q \in \{2, 4\}$. However, it doesn't yet give us a useful algorithm for explicitly calculating Jones polynomials, because throughout the method above equality is considered only up to a complex scalar multiple. Therefore we now give a scalar-exact version.

[Give scalar-exact version]

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.

3.1 Graph-Like Qutrit ZX Diagrams

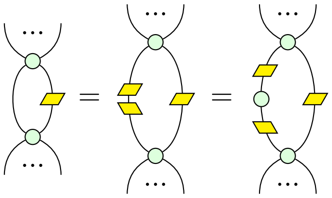
[Make all of the following scalar-exact]

We first define a graph-like diagram in the qutrit ZX calculus.

Definition 3.1. A qutrit ZX diagram is *graph-like* when:

1. Every spider is a Z-spider.
2. Spiders are only connected by Hadamard edges (H-edges) or their adjoints ($H^\dagger$-edges).
3. Every pair of spiders is connected by at most one H-edge or $H^\dagger$-edge.
4. Every input and output is connected to a spider.
5. Every spider is connected to at most one input or output.

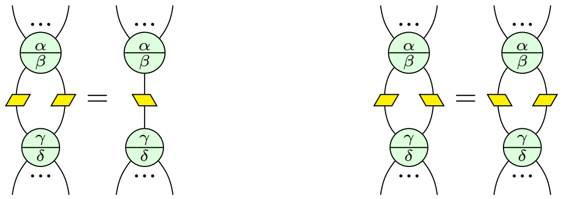
For our specific needs, the last two items above will not be relevant, since ZX-diagrams arising from knots will have no inputs or outputs, but we include them to keep the definition consistent with the qubit case. However, note the difference compared to the qubit case: we need not worry about self-loops 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. The following equation, which holds with the roles of H and $H^\dagger$ reversed, helps us get around this:



(3)

We will shortly show that every qutrit ZX diagram is equivalent to a graph-like one, making use of the following lemmas:

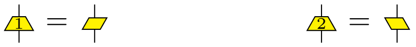
Lemma 3.2. The following two equations hold in the qutrit ZX calculus. Moreover, they hold with the roles of H and $H^\dagger$ interchanged:



(4)

Proof. This is Lemma 3.4 in [cite Harny's local complementation paper]. □

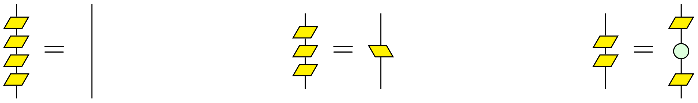
As we will formalise later, the lemma above says we can think of Hadamard edges as 1-weighted edges and their adjoints as 2-weighted edges, then work modulo 3, since every triple of parallel edges disappears. This motivates defining 'parametrised' Hadamard gates, which will come in use later:



(5)

Where the previous lemma relates single H- and H-boxes across multiple edges, the next relates multiple H- and $H^\dagger$- boxes on single edges.

Lemma 3.3. The following three equations hold in the qutrit ZX calculus. Moreover, they hold with the roles of H and $H^\dagger$ interchanged:



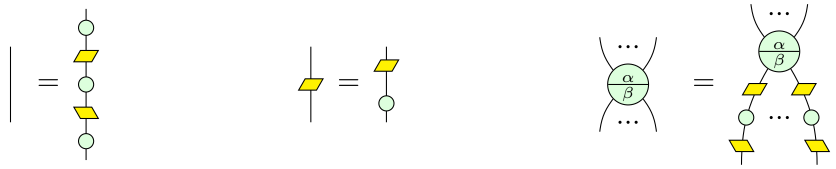
(6)

Proof. □

Again intuitively we can think of Hadamard boxes of having value 1 and their adjoints -1 and then work modulo 4.

Corollary 3.4. 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 Lemma 3.3 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 Equation 3 where fusion is not possible, so that no plain edge connects two spiders. Apply Lemma 3.2 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 again add a few spiders, H- and $H^\dagger$-boxes as needed:



(7)

□

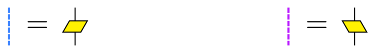
Definition 3.5. A graph-like qutrit ZX diagram is a *graph state* when every spider has zero phase (top and bottom) and is connected to an output.

A graph state is described fully by its underlying multigraph, or equivalently by an adjacency matrix, where edges take weights in $\mathbb{Z}_3$ [reference Harny]. 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 [Harny's completeness paper, Definition 2.6], though the effect is now slightly more complicated. We'll give the intuition after the formal definition:

Definition 3.6. Given $a \in \mathbb{Z}_3$ and a graph state G with adjacency matrix $W = (w_{i,j})$, the *a-local complentation* at node k is the new graph state $G *_a i$, whose adjacency matrix $W' = (w'_{i,j})$ given by:

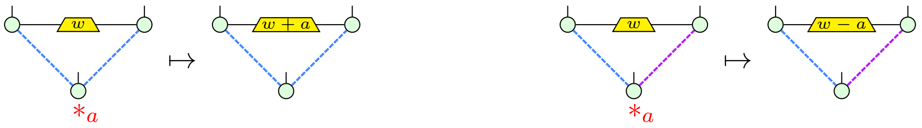
$$w'_{i,j} = w_{i,j} + a w_{i,k} w_{j,k} \tag{8}$$

So only those edges between neighbours of node k are affected, but rather than just having their weight increased by 1 (modulo 2) as in the qubit case, the increase in weight also depends on the weights of the edges from i and j to k. As always, this is best seen graphically. We reintroduce the blue dashed line notation for Hadamard edges, and now also use purple dashed lines for $H^\dagger$-edges:



(9)

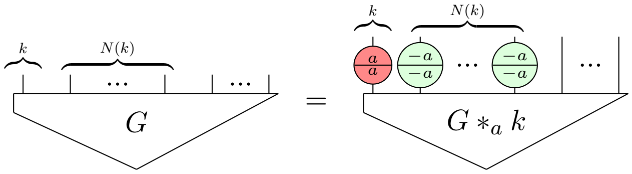
So now for two nodes i and j both connected to k by the same colour edge, a-local complementation at k increases weight $w_{i,j}$ by a. If instead i and j are connected to k by edges of different colour, a-local complementation at k decreases $w_{i,j}$ by a. We show a fragment of a ZX-diagram below under the effect of this operation:



(10)

But the fragment above doesn't give the full picture. As in the qubit case, local complementation gives an equality up to introducing some single qubit phase gates on the outputs.

Theorem 3.7. Given $a \in \mathbb{Z}_3$ and a graph state (G, W) containing a node k, let $N(k)$ denote the neighbours of k - that is, nodes i with weight $w_{i,k} \in \{1, 2\}$. Then the following equality holds:



Proof. This is Theorem 4.4 and Corollary 4.5 in [Harny's completeness paper] □

Composing local complementations gives a local pivot operation.

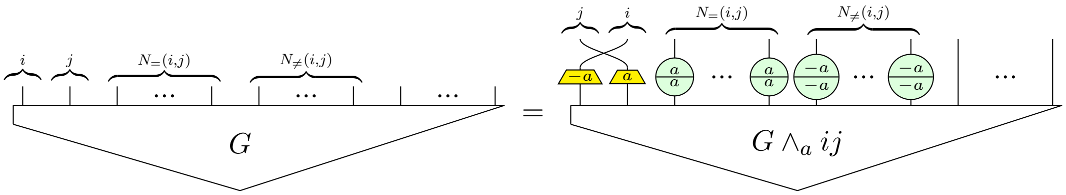
Definition 3.8. Given $a, b, c \in \mathbb{Z}_3$ and a graph state G containing nodes i and j, the (a, b, c)-*local pivot* along ij is the new graph state $G \wedge_{(a,b,c)} ij := ((G *_a i) *_b j) *_c i$.

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

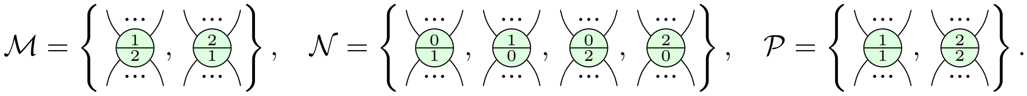
Theorem 3.9. Given $a \in \mathbb{Z}_3$ and a graph state (G, W) containing connected nodes i and j, define the following:

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

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



These two operations are again the drivers behind the simplification procedure. We classify spiders into three families:

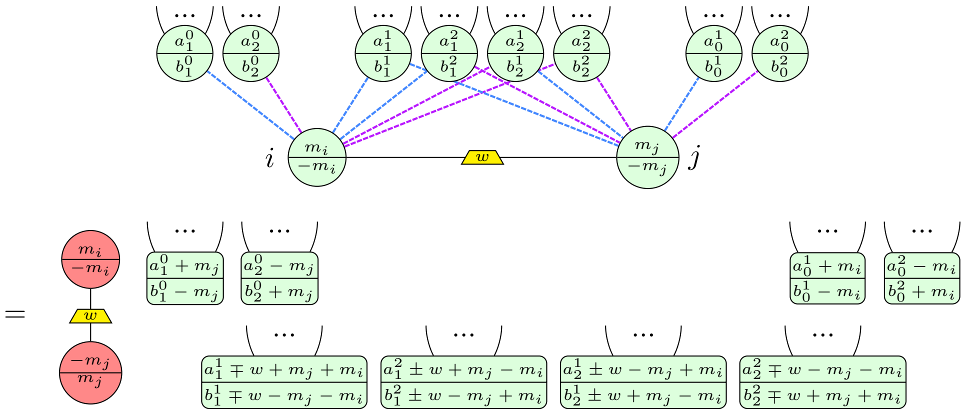


$\mathcal{M} = \{ \frac{1}{2}, \frac{2}{1} \}$,
$\mathcal{N} = \{ \frac{0}{1}, \frac{1}{0}, \frac{0}{2}, \frac{2}{0} \}$,
$\mathcal{P} = \{ \frac{1}{1}, \frac{2}{2} \}$.

(11)

This is as seen in [Wang, 2018, Theorem 3.1] except here we don't class the zero-phase spider as an $\mathcal{M}$-spider. We call a spider in a graph-like ZX-diagram *interior* if it isn't connected to an input or output. Thus for our use case all spiders are interior. Given any graph-like ZX-diagram, we will show that we can eliminate pairs of connected interior $\mathcal{M}$-spiders by local pivoting, and standalone interior $\mathcal{N}$- and $\mathcal{P}$-spiders by local complementation.

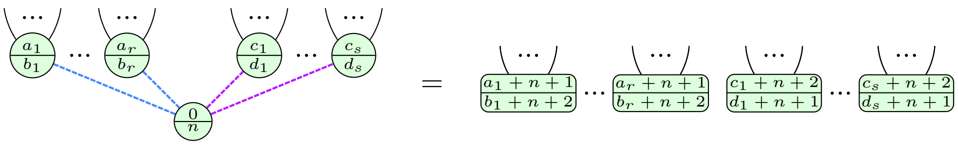
Theorem 3.10. Given any graph-like ZX-diagram containing two interior $\mathcal{M}$-spiders i and j connected by edge ij of weight $w_{i,j} =: w \in \{1, 2\}$, suppose we perform a proper $\pm w$-local pivot along ij. Then the new ZX-diagram is related to the old one by the equality:



a_1^0 / b_1^0
a_2^0 / b_2^0
a_1^1 / b_1^1
a_1^2 / b_1^2
a_2^1 / b_2^1
a_2^2 / b_2^2
a_0^1 / b_0^1
a_0^2 / b_0^2
i
m_i / $-m_i$
w
m_j / $-m_j$
j
=
m_i / $-m_i$
w
$-m_j$ / m_j
$a_1^0 + m_j$ / $b_1^0 - m_j$
$a_2^0 - m_j$ / $b_2^0 + m_j$
$a_0^1 + m_i$ / $b_0^1 - m_i$
$a_0^2 - m_i$ / $b_0^2 + m_i$
$a_1^1 \mp w + m_j + m_i$ / $b_1^1 \mp w - m_j - m_i$
$a_1^2 \pm w + m_j - m_i$ / $b_1^2 \pm w - m_j + m_i$
$a_2^1 \pm w - m_j + m_i$ / $b_2^1 \pm w + m_j - m_i$
$a_2^2 \mp w - m_j - m_i$ / $b_2^2 \mp w + m_j + m_i$

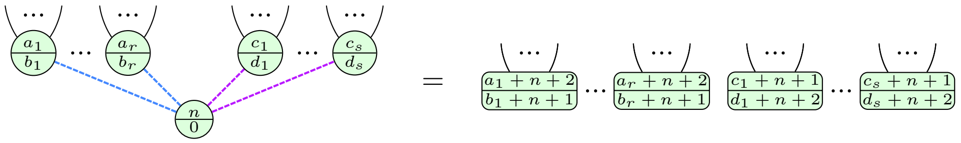
where all changes to weights of edges where neither endpoint is i or j are omitted. Furthermore, in order to save space, each node with phase $\frac{a_x^y}{b_x^y}$ is representative of *all* nodes connected to i by an x-weighted edge and to j by a y-weighted edge.

Theorem 3.11. 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:

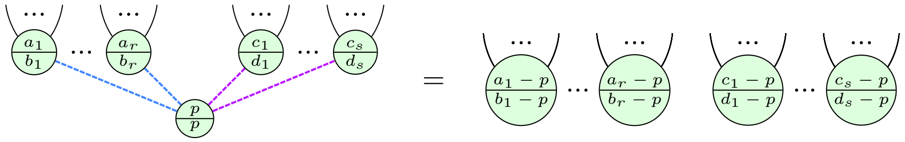


a_1 / b_1
a_r / b_r
c_1 / d_1
c_s / d_s
0 / n
=
$a_1 + n + 1$ / $b_1 + n + 2$
$a_r + n + 1$ / $b_r + n + 2$
$c_1 + n + 2$ / $d_1 + n + 1$
$c_s + n + 2$ / $d_s + n + 1$

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:



Theorem 3.12. Given any graph-like ZX-diagram containing an interior $\mathcal{P}$-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.

We can now combine these three elimination theorems into an algorithm for efficiently simplifying a graph-like ZX-diagram with no inputs or outputs. First note that after applying any one of the three elimination theorems to such a diagram we may end up with a state that is no longer graph-like. Fortunately the only way in which this can happen is if two spiders end up being connected by multiple H- or $H^\dagger$-edges, and we have shown in Lemma 3.2 that these can always be reduced to just one edge.

Theorem 3.13. Given any graph-like ZX-diagram with no inputs or outputs, the following algorithm will always terminate after a finite number of steps, returning an equivalent graph-like ZX-diagram with no $\mathcal{N}$-spiders, $\mathcal{P}$-spiders, or adjacent pairs of $\mathcal{M}$-spiders. Repeat the steps below until no rule matches. After each step, apply Lemma 3.2 as needed until the resulting diagram is graph-like:

1. Eliminate an $\mathcal{N}$-spider via Theorem 3.11.

2. Eliminate a $\mathcal{P}$-spider via Theorem 3.12.

3. Eliminate two adjacent $\mathcal{M}$-spiders via Theorem 3.10.

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 $\mathcal{N}$-spiders, $\mathcal{P}$-spiders, or adjacent pairs of $\mathcal{M}$-spiders. □

Corollary 3.14. In particular, if we start with a stabilizer diagram, we can eliminate all but perhaps 1 $\mathcal{M}$-spider, depending on whether the initial number of $\mathcal{M}$-spiders was odd or even.

Proof. No step introduces any non-stabilizer phases. □

The algorithm above could be extended to graph-like diagrams with inputs or outputs as in the qubit case, but since for our purposes we don't need to do so, we have not gone to the trouble.

References

Q. Wang. Qutrit zx-calculus is complete for stabilizer quantum mechanics. 2018.

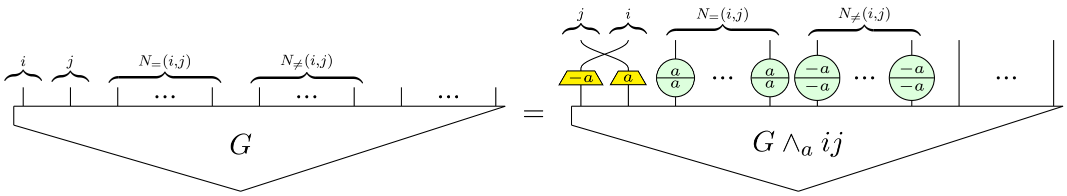
4 Appendix

We begin with a proof of the local pivot equality from Theorem 3.9.

Theorem 3.9. Given $a \in \mathbb{Z}_3$ and a graph state (G, W) containing connected nodes i and j, define the following:

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

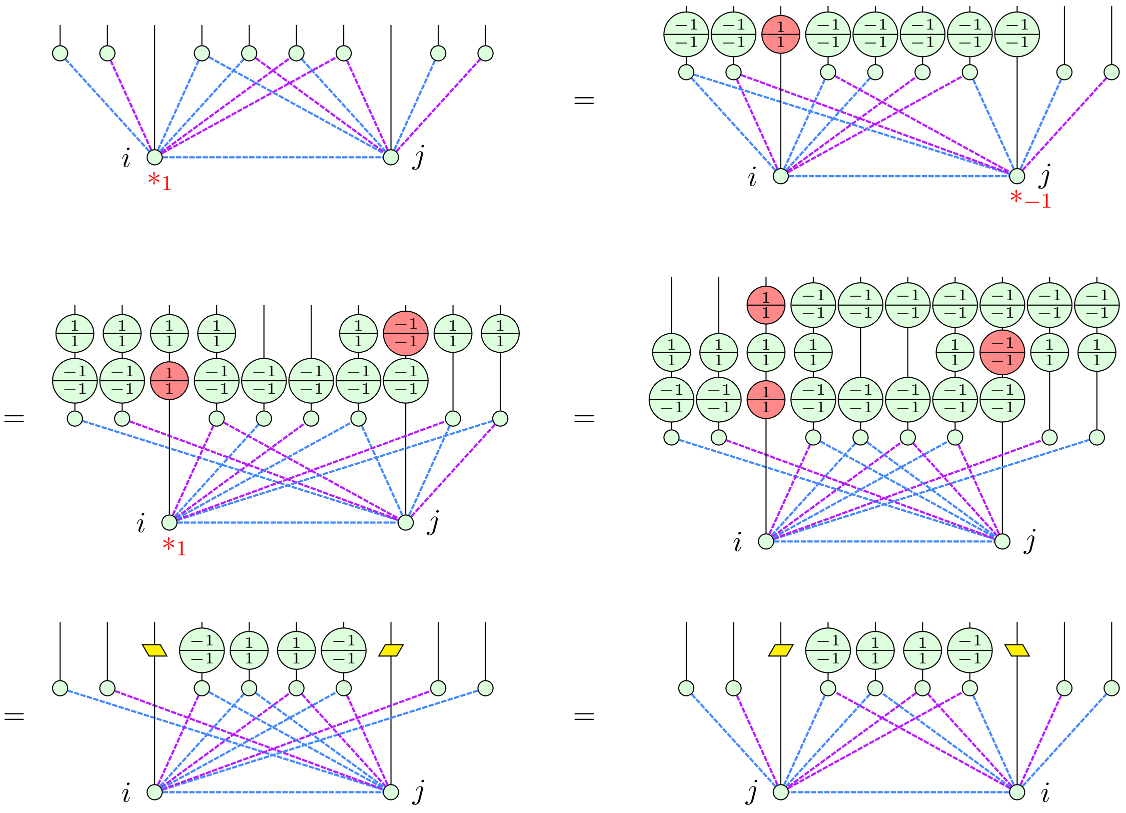
Then the following equation relates G and its proper a-local pivot along ij:



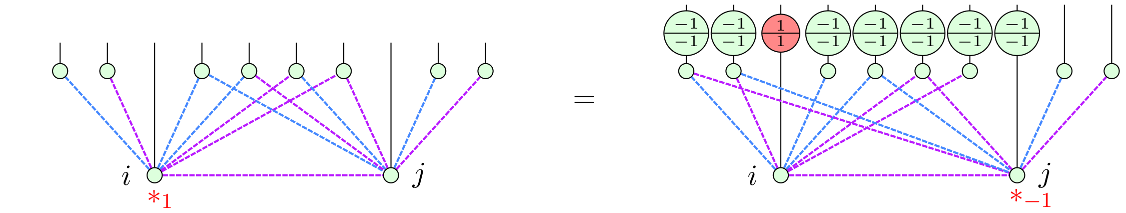
Proof. Annoyingly, proving this in full generality in one go - i.e. for a proper a-local pivot along an edge ij of weight b - becomes a bit tricky diagramatically, because it becomes hard to keep track of all the variable edge weights. Fortunately the four cases ($a, b \in \{1, 2\}$) split into two pairs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

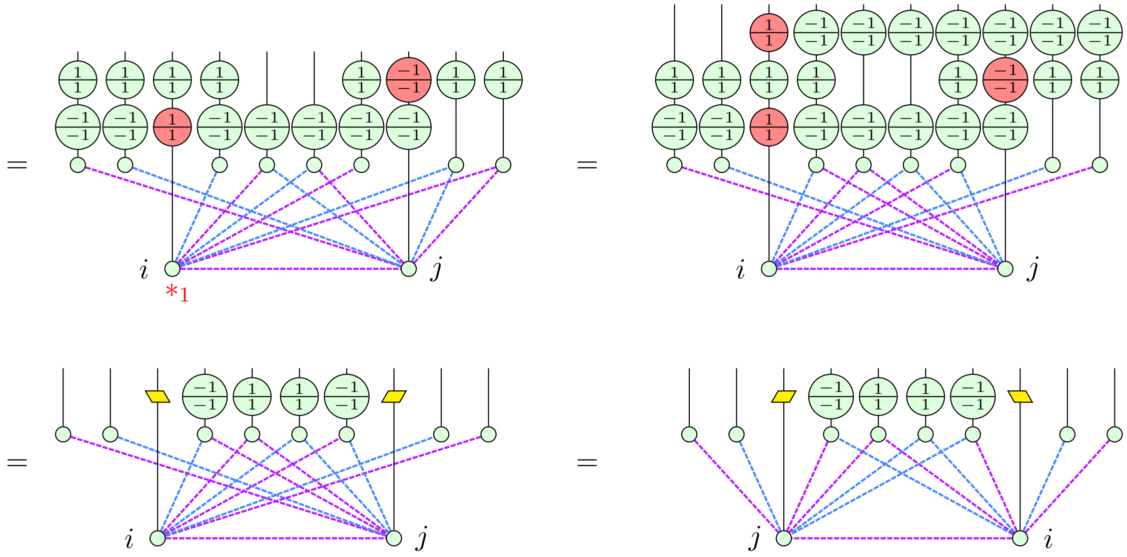
Now, it suffices to only draw a fragment of a graph state. Certainly we consider nodes i and j and the edge ij between them. Then define N_x^y to be the set $\{k \mid w_{k,i} = x, w_{k,j} = y\}$, for $x, y, \in \mathbb{Z}_3$. We will consider a representative node k_x^y from each $N_x^y \neq N_0^0$, 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:



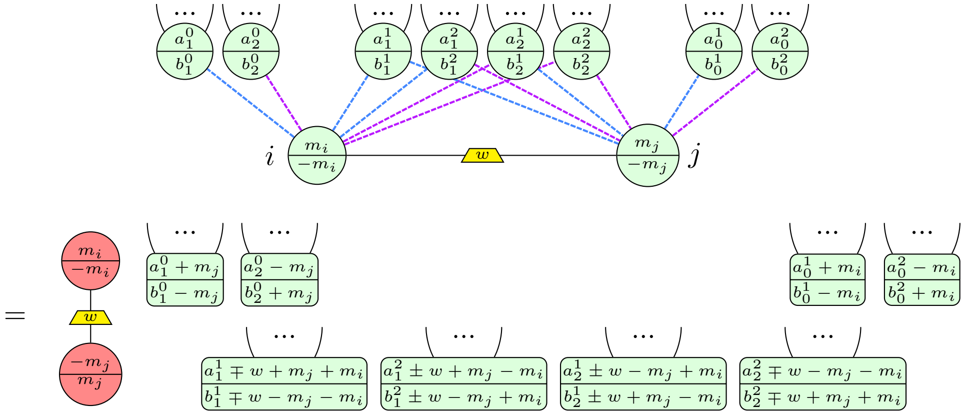


The case $a = b = 2$ is the same as $a = b = 1$, except with the roles of purple and blue edges interchanged, so by symmetry of the diagram the only difference is in the phase gates added to the outputs. Namely, on each phase gate we replace all instances of 1 with a 2. Thus the roles of H and $H^\dagger$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

□

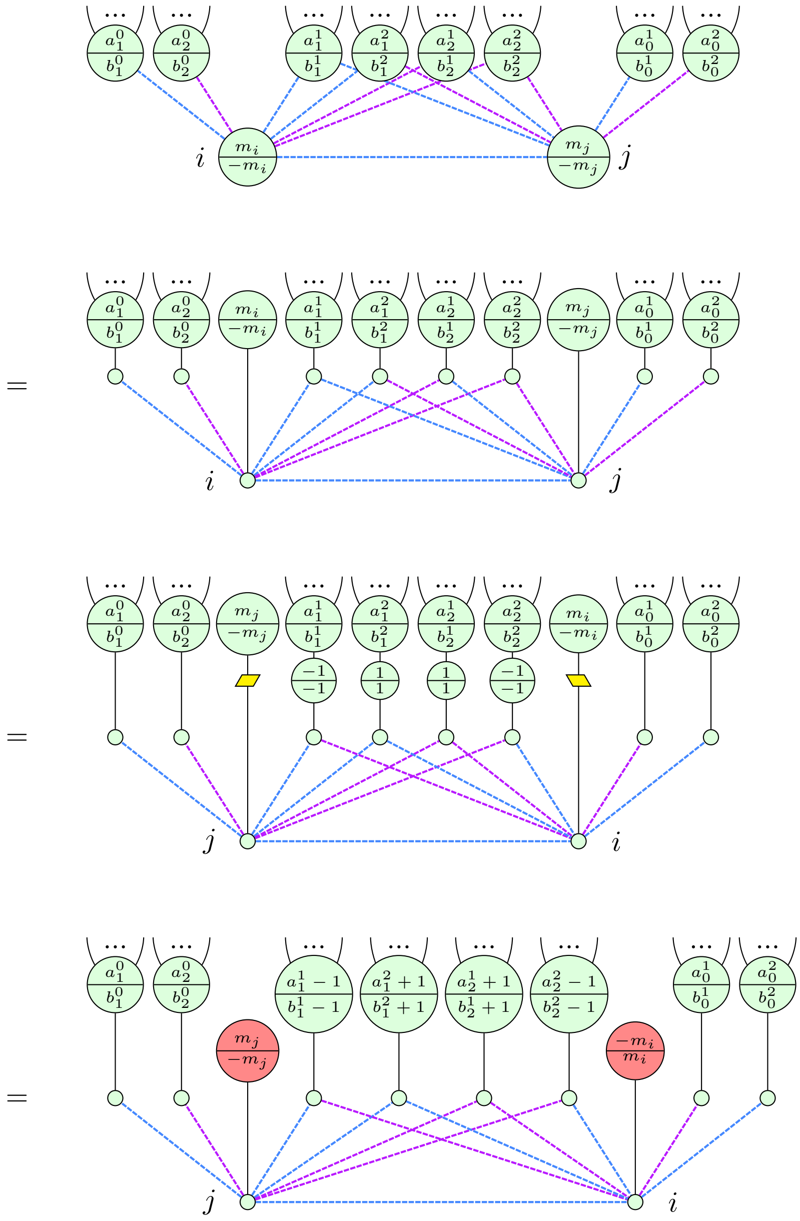
Here we prove the three elimination theorems for $\mathcal{M}$-, $\mathcal{N}$- and $\mathcal{P}$-spiders.

Theorem 3.10. Given any graph-like ZX-diagram containing two interior $\mathcal{M}$-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:

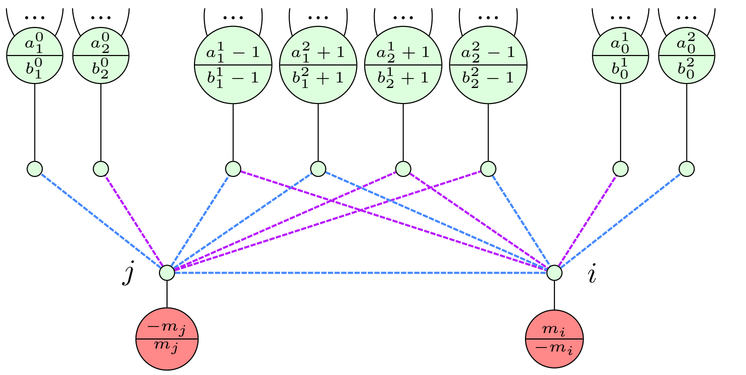

a_1^0 / b_1^0
a_2^0 / b_2^0
a_1^1 / b_1^1
a_1^2 / b_1^2
a_2^1 / b_2^1
a_2^2 / b_2^2
a_0^1 / b_0^1
a_0^2 / b_0^2
i
m_i / $-m_i$
w
m_j / $-m_j$
j
=
m_i / $-m_i$
w
$-m_j$ / m_j
$a_1^0 + m_j$ / $b_1^0 - m_j$
$a_2^0 - m_j$ / $b_2^0 + m_j$
$a_0^1 + m_i$ / $b_0^1 - m_i$
$a_0^2 - m_i$ / $b_0^2 + m_i$
$a_1^1 \mp w + m_j + m_i$ / $b_1^1 \mp w - m_j - m_i$
$a_1^2 \pm w + m_j - m_i$ / $b_1^2 \pm w - m_j + m_i$
$a_2^1 \pm w - m_j + m_i$ / $b_2^1 \pm w + m_j - m_i$
$a_2^2 \mp w - m_j - m_i$ / $b_2^2 \mp w + m_j + m_i$

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:

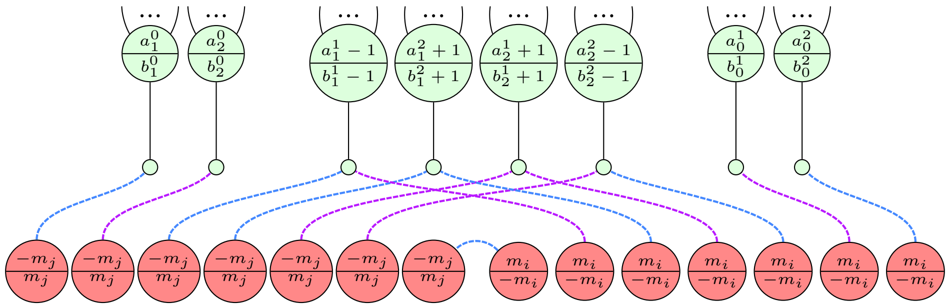


=



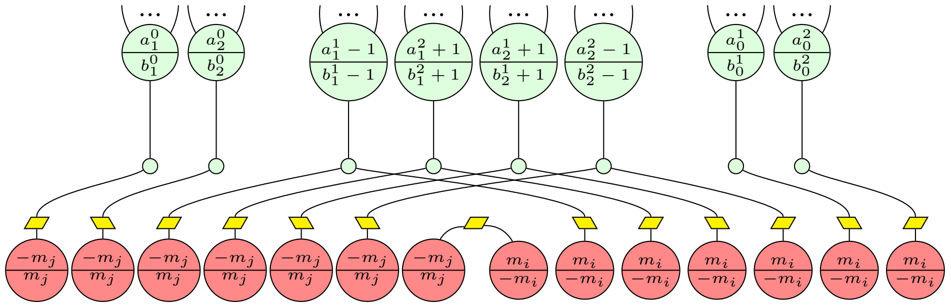
$\cdots$
a_1^0 / b_1^0
a_2^0 / b_2^0
a_1^1-1 / b_1^1-1
a_1^2+1 / b_1^2+1
a_2^1+1 / b_2^1+1
a_2^2-1 / b_2^2-1
a_0^1 / b_0^1
a_0^2 / b_0^2
j
i
$-m_j$ / m_j
m_i / $-m_i$

=



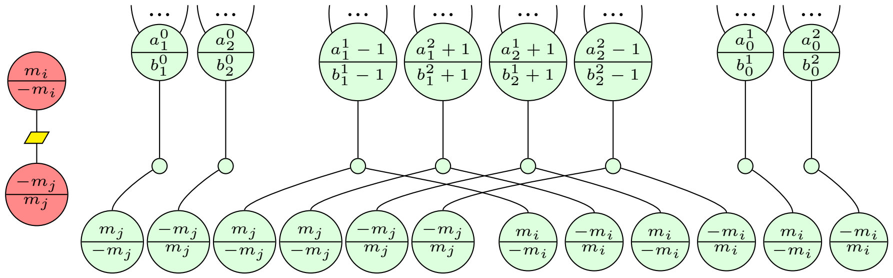
$\cdots$
a_1^0 / b_1^0
a_2^0 / b_2^0
a_1^1-1 / b_1^1-1
a_1^2+1 / b_1^2+1
a_2^1+1 / b_2^1+1
a_2^2-1 / b_2^2-1
a_0^1 / b_0^1
a_0^2 / b_0^2
$-m_j$ / m_j
m_i / $-m_i$

=

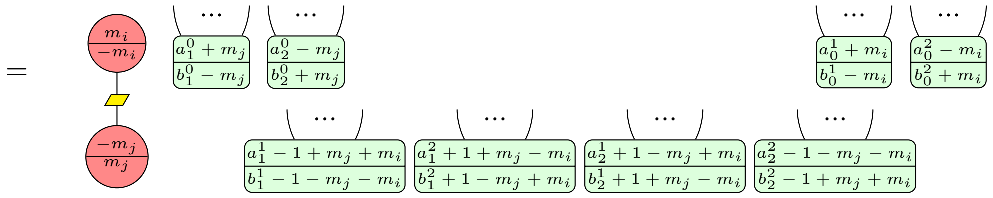


$\cdots$
a_1^0 / b_1^0
a_2^0 / b_2^0
a_1^1-1 / b_1^1-1
a_1^2+1 / b_1^2+1
a_2^1+1 / b_2^1+1
a_2^2-1 / b_2^2-1
a_0^1 / b_0^1
a_0^2 / b_0^2
$-m_j$ / m_j
m_i / $-m_i$

=

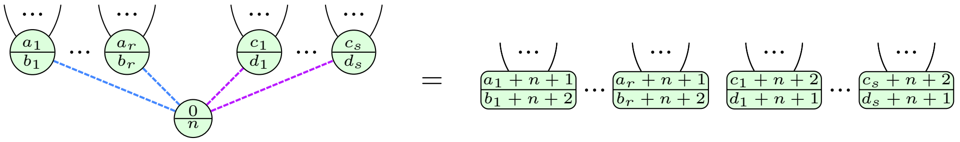


m_i / $-m_i$
$-m_j$ / m_j
$\cdots$
a_1^0 / b_1^0
a_2^0 / b_2^0
a_1^1-1 / b_1^1-1
a_1^2+1 / b_1^2+1
a_2^1+1 / b_2^1+1
a_2^2-1 / b_2^2-1
a_0^1 / b_0^1
a_0^2 / b_0^2
m_j / $-m_j$
$-m_j$ / m_j
m_j / $-m_j$
m_j / $-m_j$
$-m_j$ / m_j
$-m_j$ / m_j
m_i / $-m_i$
$-m_i$ / m_i
m_i / $-m_i$
$-m_i$ / m_i
m_i / $-m_i$
$-m_i$ / m_i

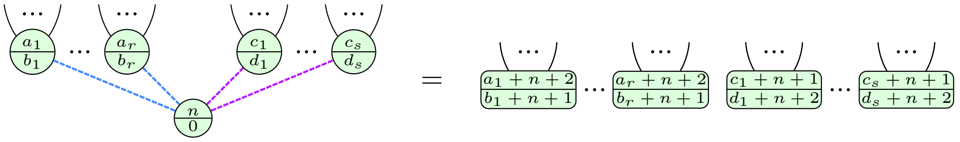

=
m_i / $-m_i$
$-m_j$ / m_j
$a_1^0+m_j$ / $b_1^0-m_j$
$a_2^0-m_j$ / $b_2^0+m_j$
$a_0^1+m_i$ / $b_0^1-m_i$
$a_0^2-m_i$ / $b_0^2+m_i$
$a_1^1-1+m_j+m_i$ / $b_1^1-1-m_j-m_i$
$a_1^2+1+m_j-m_i$ / $b_1^2+1-m_j+m_i$
$a_2^1+1-m_j+m_i$ / $b_2^1+1+m_j-m_i$
$a_2^2-1-m_j-m_i$ / $b_2^2-1+m_j+m_i$

□

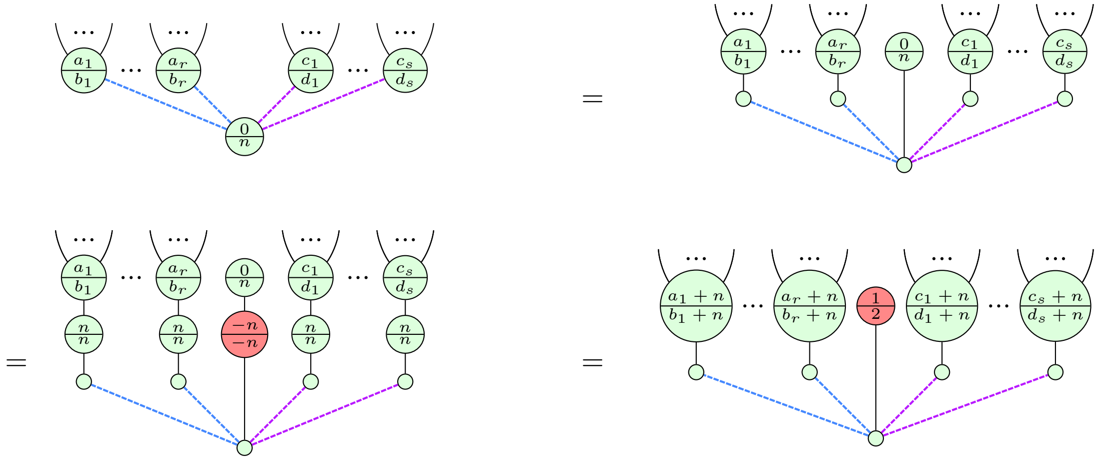
Theorem 3.11. Given any graph-like ZX-diagram containing an interior $\mathcal{N}$-spider k with phase $\frac{0}{n}$ for $n \in \{1,2\}$, suppose we perform a $(-n)$-local complementation at k. Then the new ZX-diagram is related to the old one by the equality:



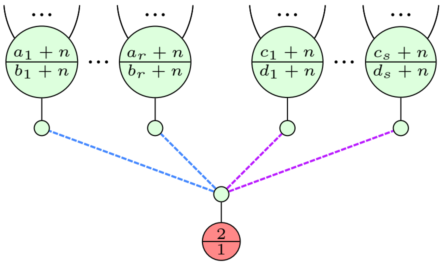
where all changes to weights of edges where neither endpoint is k are omitted. If instead k has phase $\frac{n}{0}$ for $n \in \{1,2\}$, suppose we perform the same $(-n)$-local complementation at k. Then the equality relating the new and old diagrams becomes:



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

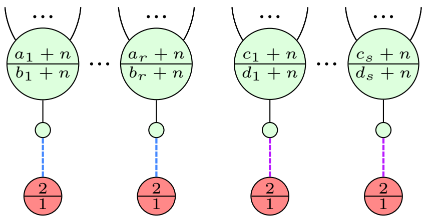


=



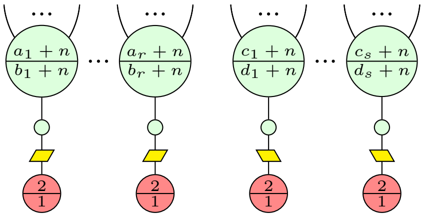
$a_1 + n$
$b_1 + n$
$a_r + n$
$b_r + n$
$c_1 + n$
$d_1 + n$
$c_s + n$
$d_s + n$
2
1

=



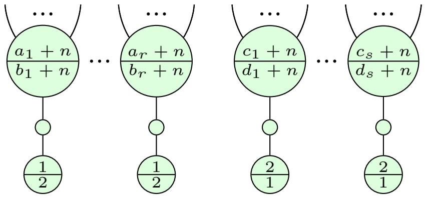
$a_1 + n$
$b_1 + n$
$a_r + n$
$b_r + n$
$c_1 + n$
$d_1 + n$
$c_s + n$
$d_s + n$
2
1
2
1
2
1
2
1

=



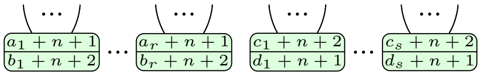
$a_1 + n$
$b_1 + n$
$a_r + n$
$b_r + n$
$c_1 + n$
$d_1 + n$
$c_s + n$
$d_s + n$
2
1
2
1
2
1
2
1

=



$a_1 + n$
$b_1 + n$
$a_r + n$
$b_r + n$
$c_1 + n$
$d_1 + n$
$c_s + n$
$d_s + n$
1
2
1
2
2
1
2
1

=



$a_1 + n + 1$
$b_1 + n + 2$
$a_r + n + 1$
$b_r + n + 2$
$c_1 + n + 2$
$d_1 + n + 1$
$c_s + n + 2$
$d_s + n + 1$

□