Finite Verification of Infinite Families of Diagram Equations

Hector Miller-Bakewell
Department of Computer Science, University of Oxford, Parks Road, Oxford, OX1 3QD
hector.miller-bakewell@cs.ox.ac.uk

The ZX, ZW and ZH calculi are all graphical calculi for reasoning about pure state qubit quantum mechanics. All of these languages use certain diagrammatic decorations, called !-boxes and phase variables, to indicate not just one diagram but an infinite family of diagrams. These decorations are powerful enough to allow complete rulesets for these calculi to be expressed in around ten rules. On the other hand reasoning involving decorated diagrams can be significantly more complicated. We present here a method for constructively reducing these infinite families of equations into finite verifying subsets. The only requirement for this construction is a property of our !-box structure that we call separability. This allows both researchers and proof assistants to reduce infinite families of problems down to undecorated, case-by-case verification, in a way not previously possible. In particular we note the removal of the need to reason directly with !-boxes in verification tasks as something entirely new, as well as extending a previously known result about removal of phase variables in verification tasks. This forms part of larger work in automated verification of quantum circuitry, and diagrammatic languages in general. The methods described here extend to any diagrammatic languages that meet certain simple conditions.

1 Introduction

The investigation of qubit graphical calculi began with Coeke and Duncan in [3], where they created the ZX calculus: A sound, universal calculus that was shown to be complete in [13]. Since then ZX has been used for such things as reasoning about quantum error correction via lattice surgery ([4] and [5],) through to being the basis for taught courses in quantum computing [2]. The $ZW_{\mathbb{C}}$ calculus, invented by Hadzihasanovic [6], presents a different point of view; rather than focus on the Z and X rotations of the Bloch sphere as ZX does, it focusses on the GHZ and W entanglement states. The ZH calculus (Kissinger and Backens, [1]) has another viewpoint again; that of extending the notion of Hadamard and CCZ gates.

Each of these calculi has strength in different areas, and all of them are sound, complete and universal for pure state qubit quantum mechanics. With one exception the rules of these calculi are expressed in a finite manner; that is to say a finite collection of parameterised families of equations. This parameterisation is an expression of two types of regular structure:

- Phase variables: e.g. “This X can be any complex number”
- !-boxes: e.g. “This part of the diagram can be repeated 0 or more times”

The one exception to this being the (EU) rule of the ZX calculus (see [14]) which use what we call *side conditions*, a case we shall not be considering in this paper.

The aim of this paper is to show when these parameterisations admit themselves to finite verification: A process by which a person or computer can verify an entire infinite family by checking a finite number of cases. For phase variables we exploit the properties of the polynomial functions they represent (theorem 1), and for !-boxes we exploit the finite dimensionality of the space in which their repeated structure resides (theorem 2). We show how these two types of parameterisation interact in theorem 3 before finally giving a constructive method for finding a verifying subset in theorem 4.

In essence finite verification allows us to start with an infinite family of equations (indicated by a single diagram decorated with phase variables and !-boxes,) and show that this entire family of equations are true by only checking a finite number of individual equations, none of which are decorated.

Submitted to:
QPL 2019

2 Parameterised families of diagrams

We begin by defining the language with which we discuss decorations, infinite families and so on. We then give a few examples of the way we use this language.

2.1 Types of parameters

Definition 1. *The types of parameterisation we shall consider in this paper (and define below) are:*

- ***Phase variables**, which we will label* $\alpha_1, \dots, \alpha_n$, *and*
- ***!-boxes**, which we will label* $\delta_1, \dots, \delta_m$

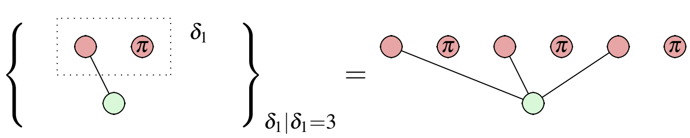
*These parameters can be **instantiated**, i.e. given an explicit value.*

For example setting a phase α_1 to the value of π is an instantiation of α_1. (There is an example explicit instantiation in section 2.2.) This act is so commonplace that there is usually no need to draw attention to it, but the proofs in this paper require explicit treatment.

Definition 2. *A **phase variable** (introduced in [3]) is a variable representing any member of the phase group (the phase group itself being a term for what data is allowed on a given node.) It can appear any number of times inside any node in the diagram, provided all those nodes admit the same phase group. For universal ZX a phase variable* α *can be any angle* $[0, 2\pi)$, *and for ZH and* $ZW_{\mathbb{C}}$ α *can be any complex number.*

Definition 3. *A **!-box** (introduced in [9], discussed in more detail in [12]) describes a region of the graph that may be repeated zero or more times. In this work we give each !-box an unique label* δ_k, *and also use this label when we wish to instantiate the !-box at a specific number of instances. For example by* $\delta_1 = 3$ *we mean to replace the subgraph labeled by* δ_1 *with three copies of that subgraph (removing the label* δ_1 *in the process,) each copy maintaining its connections to the region outside the !-box. For the nuances involving nesting of parameters inside !-boxes see section 4.1.*

Example. *This example shows an explicit instantiation of the !-box labelled by* δ_1 *at the value* $\delta_1 = 3$ *(we will continue to define the notation used after this example:)*



(1)

Definition 4. *A **simple** diagram is one made from the (instantiated) generators of the language, horizontal composition, and sequential composition without any phase variables or !-boxes. We use horizontal composition to indicate* $\otimes$ *and sequential composition to indicate* $\circ$

We include a list of the generators of ZX, ZH, and $ZW_{\mathbb{C}}$, as well as their interpretations into $\mathrm{Mat}_{\mathbb{C}}$, in appendix B.1. In contrast to simple diagrams we will also build **decorated diagrams** directly from the uninstantiated generators (which are allowed to contain variables labeled by α_j), $\otimes$, and $\circ$, and add !-boxes (labeled by δ_k) to indicate repeated elements.

Definition 5. *We write*

$$\{ \quad \mathbb{E} \quad \}_{\alpha_1,\dots,\alpha_n,\delta_1,\dots,\delta_m} \tag{2}$$

*for the **family of equations** between diagrams parameterised with phase variables* α_j *and !-boxes labeled by the* δ_k. *We use the notation* $\alpha | \alpha = a$ *to indicate that* α *has been **instantiated at** value a.*

Example. *We write:*

$$\{\quad \mathbb{E} \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\alpha_2=\pi,\delta_4=12} \tag{3}$$

for the family of equations between parameterised by $\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m$ where we have instantiated α_2 and δ_4. The use of α_j for phase variables and δ_k for !-box labels will be consistent throughout this paper. Note that if we were to instantiate every parameter that appeared in our diagram then the result would be a simple diagram.

Definition 6. *The equation $\mathbb{E}$ (between diagrams $\mathbb{D}_1$ and $\mathbb{D}_2$) **holding**, written as:*

$$\{\quad \mathbb{D}_1 = \mathbb{D}_2 \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m} \quad \textit{holds} \tag{4}$$

will mean that for any choice of instantiated values for the parameters the resulting equation between simple diagrams will hold.

It may be the case that some of the equations only hold for particular values of α or δ. We show restrictions of the parameter values by expressing the instantiated value as belonging to some subset. For example if the family $\mathbb{E}$ holds true for $\alpha_2 = a_2$ whenever $a_2 \in A_2$ we will just write:

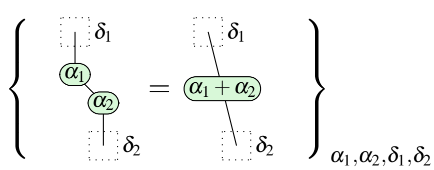
$$\{\quad \mathbb{E} \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\alpha_2\in A_2} \quad \text{holds} \tag{5}$$

2.2 Explicit parameterisation and instantiation

Example. *The (slightly simplified) spider law in universal ZX is parameterised over*

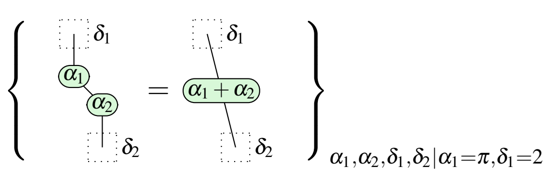
- $\delta_1 \in \mathbb{N}$ *inputs and* $\delta_2 \in \mathbb{N}$ *outputs*
- $\alpha_1, \alpha_2 \in [0, 2\pi)$

And so we write the parameterised family of equations as:

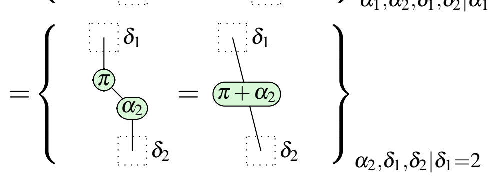


(6)

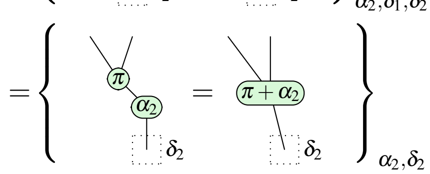
Example. *Here we instantiate some of the parameters of the spider law, resulting in what is still an infinite, parameterised family:*



(7)



(8)



(9)

3 Verifying phase parameters

Our first result will concern diagrams that contain any number of phase variables, but no !-boxes. It relies on a certain property of polynomials: That if you know that value of the polynomial $P(Y)$ at *enough* values of Y then you can determine all the coefficients of P. For example the polynomial $P(Y_1,Y_2) = a + bY_1 + cY_2 + dY_1Y_2$ can have all of its coefficients determined by knowing the values $P(0,0)$, $P(0,1)$, $P(1,0)$, and $P(1,1)$. There are three complications:

- We need need to consistently express our diagrams as matrices of polynomials
- We need to determine the precise value of *enough*
- The ZX calculus requires us to consider Laurent polynomials (polynomials that allow both positive and negative powers)

We will deal with these complications below, starting with some definitions. The idea is to capture some of the phase group structure of the diagram and express it using the polynomial structure of the entries of the matrix. Note that we are using Y to symbolise indeterminates in our polynomials to avoid confusion with the X nodes of ZX (or X Pauli matrix of quantum computing.)

3.1 Matrix interpretations

All of the graphical languages considered in this paper come equiped with a (complex) matrix interpretation, that is a mapping from simple diagrams in that language to morphisms in the category of complex matrices ($\text{Mat}_{\mathbb{C}}$.) In the introduction we mentioned pure state qubit quantum mechanics, meaning that the wires of ZX, ZH and $\text{ZW}_{\mathbb{C}}$ "carry" the information of a vector in $\mathbb{C}^2$ with the Euclidean product (also referred to as Hilbert space $\mathbb{H}$.) A diagram with n input wires and m output wires will be mapped to a matrix with $\dim \mathbb{H}^{\otimes n}$ columns and $\dim \mathbb{H}^{\otimes m}$ rows. It is this interpretation that allows us to use diagrams to represent transformations in quantum computation.

Definition 7. *A complex* ***matrix interpretation*** *for a graphical language, written $[\![\cdot]\!]$, is a monoidal functor from the category of simple diagrams (as a PROP, see e.g. [10, p 97]) to the category of complex matrices. I.e. a functor that preserves the $\otimes$ and $\circ$ products of diagrams:*

$$[\![\cdot]\!] : \textit{Simple diagrams} \to \textit{Mat}_{\mathbb{C}} \tag{10}$$

$$[\![\mathbb{D}_1 \otimes \mathbb{D}_2]\!] = [\![\mathbb{D}_1]\!] \otimes [\![\mathbb{D}_2]\!] \tag{11}$$

$$[\![\mathbb{D}_1 \circ \mathbb{D}_2]\!] = [\![\mathbb{D}_1]\!] \circ [\![\mathbb{D}_2]\!] \tag{12}$$

(We are using $\otimes$ to represent the Kronecker product of matrices, and $\circ$ to represent standard matrix multiplication.)

Appendix B contains matrix interpretations for the ZX, ZH and ZW calculi, or see [1] and [6] for the ZH and ZW interpretations, and either [3] or [14] for the ZX interpretation.

A family of diagrams parameterised over α is a set of simple diagrams, and we could extend our interpretation so that a set of simple diagrams is sent to a set of matrices. This would, however, lose any inherent structure in our family. Instead we try to find a polynomial matrix interpretation; one that sends a family of equations $\{\mathbb{E}\}_\alpha$ to a matrix in $\text{Mat}_{\mathbb{C}[Y]}$ (matrices where each entry is a complex polynomial in Y.)

While this works well for ZH and $\text{ZW}_{\mathbb{C}}$, there is a complication with ZX: An α on a node in a ZX diagram corresponds to an $e^{i\alpha}$ in the matrix interpretation. We can try performing the substitution $Y := e^{i\alpha}$, but run into trouble if there is a node containing, for example, $-\alpha$ (and accordingly Y^{-1} in the matrix,) since polynomials do not normally allow negative powers. Rather than stick with standard polynomials we instead move to Laurent polynomials; polynomials that do allow positive and negative powers, and define all the properties we will need of them below.

ZX also introduces one more sublety: There is an extra relation from the phase group ($2\pi = 0$) that we should take care to reflect in our matrix interpretation. This does not impact any of the calculi that are universal for qubit quantum computing, but does affect the fragments of ZX with a finite phase group (see the Clifford+T example in section 3.3.)

Definition 8. *A **Laurent polynomial interpretation** is a mapping:*

$$[\![\cdot]\!] : \textit{Families of diagrams parameterised by phase variables} \to Mat_{\mathbb{C}[Y_1, Y_1^{-1}, \ldots, Y_n, Y_n^{-1}]} \tag{13}$$

We require this interpretation to have the same restrictions as a normal matrix interpretation on a simple diagram (i.e. respecting $\otimes$ and $\circ$,) and note that any (non-Laurent) polynomial interpretation is automatically also a Laurent polynomial interpretation. This monoidal nature means that we can simply specify how to interpret each of our generators to get an interpretation for any diagram. See appendix B for explicit interpretations of the ZX, ZW and ZH generators into $Mat_{\mathbb{C}[Y_1, Y_1^{-1}, \ldots, Y_n, Y_n^{-1}]}$.

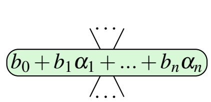
Example. *The Z spider from universal ZX is parameterised by an $\alpha \in [0, 2\pi)$, and the (simple) matrix interpretation of some Z spider (with α instantiated at a) is:*

$$\left[\!\left[\left\{ \cdots \right\} \Big|_{\alpha = a} \right]\!\right] = \begin{bmatrix} 1 & 0 & \dots & & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 0 & 0 \\ 0 & \dots & & 0 & e^{ia} \end{bmatrix} \in Mat_{\mathbb{C}} \tag{14}$$

Rather than instantiate the value of α before we apply the map, we instead make the substitution $Y = e^{i\alpha}$ to find a Laurent polynomial matrix interpretation:

$$[\![\cdots]\!] = \begin{bmatrix} 1 & 0 & \dots & & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 0 & 0 \\ 0 & \dots & & 0 & Y \end{bmatrix} \in Mat_{\mathbb{C}[Y, Y^{-1}]} \tag{15}$$

In general the Z spider with phase $b_0 + b_1\alpha_1 + \ldots + b_n\alpha_n$ will be interpreted as:



$$[\![\cdots]\!] = \begin{bmatrix} 1 & 0 & \dots & & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 0 & 0 \\ 0 & \dots & & 0 & \left(e^{ib_0} \times \prod_j Y_j^{b_j}\right) \end{bmatrix} \in Mat_{\mathbb{C}[Y_1, Y_1^{-1}, \ldots, Y_n, Y_n^{-1}]} \tag{16}$$

3.2 Degree of a matrix

Definition 9. *Laurent polynomial degrees (defined in the same manner as e.g. [15]:)*

- *The 0 polynomial has degree $-\infty$ by convention*

- *the non-zero Laurent polynomial* $a_n Y^n + a_{n-1} Y^{n-1} + \cdots + a_0 + a_{-1} Y^{-1} + \cdots + a_{-m} Y^{-m}$ *with* $a_n \neq 0 \neq a_{-m}$ *has* ***positive degree*** $n \geq 0$ *and* ***negative degree*** $m \geq 0$*.*

Note that we can factorise this Laurent polynomial as Y^{-m} *multiplied by a (non-Laurent) polynomial.*

Definition 10. *We define here a notion of matrix and diagram degrees:*

- *The* Y_j^+***-degree*** *of a matrix in* $Mat_{\mathbb{C}[Y_1, Y_1^{-1}, \ldots, Y_n, Y_n^{-1}]}$ *is the maximum of the positive* Y_j*-degrees of the entries in that matrix*
- *The* Y_j^-***-degree*** *is the maximum of the negative* Y_j*-degrees of the entries in that matrix*
- *The positive degree of a diagram is the positive degree of the matrix interpretation of that diagram (likewise for negative degrees.) When clear from context we will refer to the degree of a phase variable* α_j *in the diagram, meaning the degree of* Y_j *in the interpretation.*

Here are example positive and negative degrees for first a polynomial, and then a 2×2 matrix of polynomials.

Example.

$$deg_Y^+ \left[Y^8 + 1 + Y^{-2} \right] = 8 \tag{17}$$
$$deg_Y^- \left[Y^8 + 1 + Y^{-2} \right] = 2 \tag{18}$$

$$deg_Y^+ \begin{bmatrix} 2 & Y^{-3} \\ Y & Y^2 - 2 \end{bmatrix} = \max\{0, 0, 1, 2\} = 2 \tag{19}$$
$$deg_Y^- \begin{bmatrix} 2 & Y^{-3} \\ Y & Y^2 - 2 \end{bmatrix} = \max\{0, 3, 0, 0\} = 3 \tag{20}$$

Proposition 1. *The (positive or negative)* Y_j*-degree of the interpretation of an entire diagram is bounded above by the sum of the (positive or negative)* Y_j*-degrees of elements of that diagram. Where by elements we mean subdiagrams that are joined by* $\otimes$ *and* $\circ$ *to form the larger diagram.*

Proof. We first note that for Laurent polynomials P and P' in $\mathbb{C}[Y, Y^{-1}]$, and for $\lambda \in \mathbb{C}$:

$$\deg_Y^+ [\lambda P] \leq \deg_Y^+ [P] \tag{21}$$
$$\deg_Y^- [\lambda P] \leq \deg_Y^- [P] \tag{22}$$

$$\deg_Y^+ \left[P \times P' \right] \leq \deg_Y^+ [P] + \deg_Y^+ \left[P' \right] \tag{23}$$
$$\deg_Y^- \left[P \times P' \right] \leq \deg_Y^- [P] + \deg_Y^- \left[P' \right] \tag{24}$$

$$\deg_Y^+ \left[\sum_j P_j \right] \leq \max_j \deg_Y^+ [P_j] \tag{25}$$
$$\deg_Y^- \left[\sum_j P_j \right] \leq \max_j \deg_Y^- [P_j] \tag{26}$$

The composition $A \circ B$ or tensor product $A \otimes B$ of matrices produces a new matrix with entries that are linear combinations of products of the entries of A and B. Therefore:

$$\begin{aligned} \deg_Y^+\left[\, M \circ M' \,\right] &\leq \deg_Y^+\left[\, M \,\right] + \deg_Y^+\left[\, M' \,\right] & (27)\\ \deg_Y^+\left[\, M \otimes M' \,\right] &\leq \deg_Y^+\left[\, M \,\right] + \deg_Y^+\left[\, M' \,\right] & (28)\end{aligned}$$

$$\begin{aligned} \deg_Y^-\left[\, M \circ M' \,\right] &\leq \deg_Y^-\left[\, M \,\right] + \deg_Y^-\left[\, M' \,\right] & (29)\\ \deg_Y^-\left[\, M \otimes M' \,\right] &\leq \deg_Y^-\left[\, M \,\right] + \deg_Y^-\left[\, M' \,\right] & (30)\end{aligned}$$

We use this, along with the monoidal nature of our interpretation, to see that for diagrams $\mathbb{D}$ and $\mathbb{D}'$ (recalling that the degree of a diagram is the degree of its polynomial matrix interpretation:)

$$\begin{aligned} \deg^+\left[\, \mathbb{D} \circ \mathbb{D}' \,\right] &\leq \deg_Y^+\left[\, \mathbb{D} \,\right] + \deg_Y^+\left[\, \mathbb{D}' \,\right] & (31)\\ \deg^+\left[\, \mathbb{D} \otimes \mathbb{D}' \,\right] &\leq \deg_Y^+\left[\, \mathbb{D} \,\right] + \deg_Y^+\left[\, \mathbb{D}' \,\right] & (32)\end{aligned}$$

$$\begin{aligned} \deg^-\left[\, \mathbb{D} \circ \mathbb{D}' \,\right] &\leq \deg_Y^-\left[\, \mathbb{D} \,\right] + \deg_Y^-\left[\, \mathbb{D}' \,\right] & (33)\\ \deg^-\left[\, \mathbb{D} \otimes \mathbb{D}' \,\right] &\leq \deg_Y^-\left[\, \mathbb{D} \,\right] + \deg_Y^-\left[\, \mathbb{D}' \,\right] & (34)\end{aligned}$$

Since all diagrams are built from basic elements via $\otimes$ and $\circ$ we achieve our result. □

3.3 Finite verification

Theorem 1. *For a diagrammatic equation without !-boxes*

$$\{ \quad \mathbb{D}_1 = \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n}$$

that has a Laurent polynomial matrix interpretation, and the interpretations agree on a large enough grid of points $(a_1,\ldots,a_n) \in A_1 \times \cdots \times A_n$ in the parameter space, then the interpretations agree on all values of $(a_1,\ldots,a_n)$.

$$\begin{aligned} &\left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n|\alpha_1=a_1,\ldots,\alpha_n=a_n} \right]\!\right] = \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n|\alpha_1=a_1,\ldots,\alpha_n=a_n} \right]\!\right] \\ &\qquad \forall a_1 \in A_1,\ldots,a_n \in A_n \\ \Longrightarrow &\left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n|\alpha_1=a_1,\ldots,\alpha_n=a_n} \right]\!\right] = \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n|\alpha_1=a_1,\ldots,\alpha_n=a_n} \right]\!\right] \\ &\qquad \forall a_1,\ldots,a_n \end{aligned} \tag{35}$$

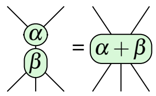
The size of $|A_j|$ (corresponding to α_j) is given by:

$$\begin{aligned} |A_j| = &\max(deg_{Y_j}^+\left[\, \mathbb{D}_1 \,\right], deg_{Y_j}^+\left[\, \mathbb{D}_2 \,\right]) \\ &+ \max(deg_{Y_j}^-\left[\, \mathbb{D}_1 \,\right], deg_{Y_j}^-\left[\, \mathbb{D}_2 \,\right]) \\ &+ 1 \end{aligned} \tag{36}$$

"The maximum positive degree on either side, plus the maximum negative degree on either side, plus one."

The proof can be found in appendix A.1

Example. *The following (universal) ZX diagram contains no !-boxes and two phase variables.*

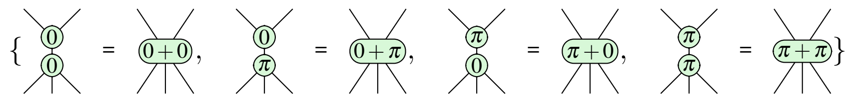


$$\deg^+_\alpha \mathbb{D}_1 = 1 \quad \deg^+_\beta \mathbb{D}_1 = 1 \qquad \deg^+_\alpha \mathbb{D}_2 = 1 \quad \deg^+_\beta \mathbb{D}_2 = 1$$
$$\deg^-_\alpha \mathbb{D}_1 = 0 \quad \deg^-_\beta \mathbb{D}_1 = 0 \qquad \deg^-_\alpha \mathbb{D}_2 = 0 \quad \deg^-_\beta \mathbb{D}_2 = 0 \tag{37}$$

We should therefore construct A_α and A_β such that $|A_\alpha| = \max\{1,1\} + \max\{0,0\} + 1$ and $|A_\beta| = \max\{1,1\} + \max\{0,0\} + 1$. By picking $A_\alpha = A_\beta = \{0,\pi\}$ we therefore know that we can verify this parameterised family of diagram equations for all values of α and β by verifying this equation on the following grid of values:

$$\begin{array}{lll} & \alpha = 0 & \alpha = \pi \\ \beta = 0 & (0,0) & (0,\pi) \\ \beta = \pi & (\pi,0) & (\pi,\pi) \end{array} \tag{38}$$

i.e. by verifying the four equations:



(39)

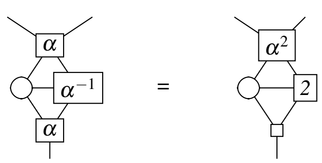
we can assert that the diagram equation in (37) *holds for all values of α and β in* $[0,2\pi)$.

Corollary. *The ZX version of this result was first proved in [7], however the methods used in that paper apply to ZX only: That in the universal ZX calculus it suffices to check $(\alpha_1,\ldots,\alpha_n) \in A_1 \times \ldots A_n$ to prove an equation that is **linear** in the α_j, where the A_j are sets of distinct angles with*

$$\begin{aligned} |A_j| = &\max(deg^+_{\alpha_j}[\,\mathbb{D}_1\,] + deg^+_{\alpha_j}[\,\mathbb{D}_2\,] \\ &+ \max(deg^-_{\alpha_j}[\,\mathbb{D}_1\,] + deg^-_{\alpha_j}[\,\mathbb{D}_2\,]) \\ &+ 1 \end{aligned} \tag{40}$$

[7] uses the symbol μ to count appearances of α_j (with coefficient,) and T_j to denote a large enough set of values, and the result is expressed in their theorem 3. This sidesteps needing to go via Laurent polynomials (instead examining ranks of certain matrices,) but also means the method does not extend to ZH and ZW.

Example. *Note that the ZX result required the variables to be linear, but for ZH and ZW this result applies to diagrams whose phases are Laurent polynomial in the α_j. For example we can verify the following ZH equation by checking 4 distinct values of α:*



(41)

Example. *Consider a Clifford+T ZX diagram that contains at least 8 nodes labeled by a positive α. Our theorem says that for any equation containing this diagram it suffices to try at least 9 distinct values of α, but this is impossible since there are only 8 distinct values of α available in Clifford+T.*

This is because our Laurent polynomial matrix interpretation needs to be viewed not in $\mathbb{C}[Y,Y^{-1}]$ but in $\mathbb{C}[Y]/(Y^8-1)$, reflecting the property $8\times\alpha = 0$ in our phase group. All polynomials in $\mathbb{C}[Y]/(Y^8-1)$ have degree at most 7, and so it is never necessary to check more than 8 points.

4 Verifying !-boxes

We now turn our attention to !-boxes. The aim of this section is to show that if an equation holds for $0,\dots,N$!-box instances, then it continues to hold for any number of instances. It then follows that one only needs to check the first $0,\dots,N$ instances in order to verify the entire family. The method relies on the finite dimensionality of what we call the *join* between the inside and outside of a !-box. Sometimes this join is not finite dimensional, and so we examine a property called *separated* which captures when the method will work. Before we can get to that we first make clear how the nesting of !-boxes and phase variables can work.

4.1 Children, copies, and the nesting order

We begin with some definitions for describing the effect of nesting !-boxes inside a parameterised family of equations. There is a choice to be made when nesting parameters:

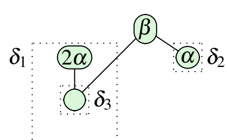
Definition 11. *When a !-box creates new instances of a nested parameter we either* ***copy*** *that variable name, so that all instances are linked by the same name (the approach taken in [12, §4.4.2],) or create new names, each of which is referred to as a* ***child*** *of the original parameter name. When we create child parameters we record the name of its parent, so we can always tell the original ancestor of a parameter.*

Rather than pick one option over the other we will demonstrate our results for both choices. In order to talk about nesting formally we introduce the following definition:

Definition 12. *We define a partial order (which we call the* ***nesting order****) on !-boxes in a diagram:*

$$\delta_1 < \delta_2 \quad \text{if } \delta_1 \text{ is nested inside } \delta_2$$

And use this partial order to draw a nesting diagram. For example this (universal) ZX diagram:



has nesting diagram

$$\begin{array}{cc} \delta_1 & \delta_2 \\ \downarrow & \\ \delta_3 & \end{array}$$

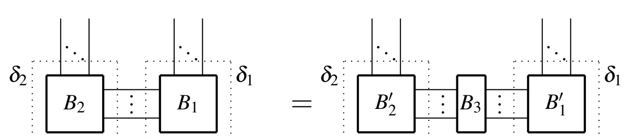
Definition 13. *We say an equation is* ***well nested*** *if the nesting diagrams corresponding to the left and right hand sides of the equation are identical.*

4.2 Separability

We describe a pair of !-boxes as **separated** if either:

- They are nested, or
- They share no edges

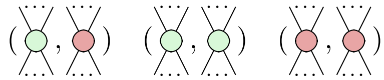
We describe a non-separated pair of !-boxes as **separable** if we can perform the following operation:

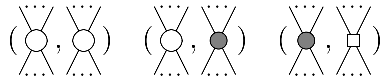


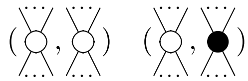
(42)

We define pairs of nodes as separable if we can always separate !-boxes that are joined by edges between these pairs of nodes. Note that we only need to consider nodes that have arbitrary arity, since only they can be connected to !-boxes.

- The following pairs of nodes are separable, by language:

ZX:  (43)

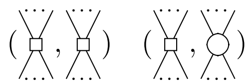
ZH:  (44)

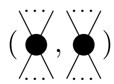
ZW:  (45)

The proofs of these follow immediately from the spider and bialgebra laws in the respective languages.

- The following pairs of nodes are assumed to not be separable, by language:

ZX: Always separable (46)

ZH:  (47)

ZW:  (48)

Note that it is enough to specify the phase-free versions of these interactions, because phases can always be moved away from the critical nodes.

4.3 Verification

Definition 14. *Given a parameterised equation* $\mathbb{E}$ *we say that* $\{\mathbb{E}_1, \ldots, \mathbb{E}_n\}$ ***verifies*** $\mathbb{E}$ *if:*

$$\forall j\ (\mathbb{E}_j \text{ holds for all parameters in } \mathbb{E}_j) \implies \mathbb{E} \text{ holds for all parameters in } \mathbb{E}$$

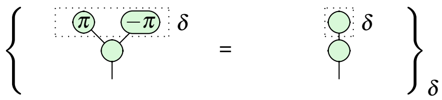
Theorem 2. *Given a family* $\mathbb{E} = \{\quad \mathbb{D}_1 = \mathbb{D}_2 \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m}$ *of diagrammatic equations, parameterised by a !-box* δ_1 *where* δ_1 *is separated from all other !-boxes and is nested in no other !-box; then E is verified by the finite family* $\{\mathbb{E}|_{\delta_1=0}, \ldots, \mathbb{E}|_{\delta_1=N}\}$ *where N is the sum of the dimensions of the joins between* δ_1 *and the rest of the diagram. That is:*

$$\begin{aligned}
&\left[\!\left[\{\quad \mathbb{D}_1 \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \\
&= \left[\!\left[\{\quad \mathbb{D}_2 \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \qquad \forall d_1 \leq N \\
\\
\implies &\left[\!\left[\{\quad \mathbb{D}_1 \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \\
&= \left[\!\left[\{\quad \mathbb{D}_2 \quad\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \qquad \forall d_1
\end{aligned}$$

$$\begin{aligned}
N &:= \dim\left(\mathbb{H}^{\otimes n_1} \oplus \mathbb{H}^{\otimes n_2}\right) \\
n_1 &:= \textit{number of wires between } \delta_1 \textit{ and the rest of } \mathbb{D}_1 \\
n_2 &:= \textit{number of wires between } \delta_1 \textit{ and the rest of } \mathbb{D}_2
\end{aligned}$$

The proof is presented in appendix A.2. Note that the the resulting finite family of equations need not be simple, and that if δ_1 is not separated from the other !-boxes then we cannot put a bound on N.

Example. *Consider the following (qubit) ZX family of equations, parameterised by a single !-box:*

$\mathbb{E} :=$ 

(49)

The join between the !-box and the rest of the diagram on the left is two wires (dimension 2^2), and on the right is one wire (dimension 2^1.) These sum to have dimension 6, and we therefore need only to check the !-box instances $(0,\dots,6)$ to be sure that (the matrix interpretation of) this equation holds for all instances of δ. Since ZX is sound and complete we know that the matrix interpretation holding is enough to imply that the equation is derivable in ZX.

5 Interacting Parameters

Theorem 1 and theorem 2 deal with equations containing multiple phase variables and nested !-boxes respectively. This section will put together the necessary results such that we can combine these approaches to deal with equations containing multiple !-boxes and phase variables, any of which could potentially be nested inside other !-boxes.

Theorem 3. *Given an equation $\mathbb{E}$ and finite verifying sets D_k for the δ_k we can construct finite verifying sets A_j for α_j, such that we may verify the entire family $\mathbb{E}$ by checking the (finite) set given by the cartesian product of all the A_j and D_k.*

Proof. We define:

$$\bar{D} := D_1 \times D_2 \times \cdots \times D_m \tag{50}$$

$$\bar{\delta} := (\delta_1, \dots, \delta_m) \tag{51}$$

$$A_j(\bar{\delta}) := \text{The verifying set for } \alpha_j \text{ once } E \text{ has had !-boxes instantiated at } \bar{\delta} \tag{52}$$

$$\tag{53}$$

Construct A_j by choosing as many points as there are in $\max_{\bar{\delta}}\{|A_j(\bar{\delta})|\}$. This is finite because the D_i and the $A_j(\bar{\delta})$ are finite. A_j therefore contains enough points to be a verifying set for $A_j(\bar{\delta}) \quad \forall \bar{\delta} \in \bar{D}$.

We show that $A_1 \times \cdots \times A_n \times D_1 \times \cdots \times D_m$ is a verifying set for the parameterised equation $\mathbb{E}$:

$$\mathbb{E}_{\alpha_1,\dots,\alpha_n,\delta_1,\dots,\delta_m | \forall i,j\ \alpha_j \in A_j,\, \delta_k \in D_k} \quad \text{holds} \tag{54}$$

$$= \mathbb{E}_{\alpha_1,\dots,\alpha_n,\delta_1,\dots,\delta_m | \forall i\ \alpha_j \in A_j,\, \bar{\delta} \in \bar{D}} \quad \text{holds} \quad \text{(rewrite using } \bar{\delta} \text{ notation)} \tag{55}$$

$$\Longrightarrow \mathbb{E}_{\alpha_1,\dots,\alpha_n,\delta_1,\dots,\delta_m | \forall i\ \alpha_j \in A_j(\bar{\delta}),\, \bar{\delta} \in \bar{D}} \quad \text{holds} \quad \text{(construction of } A_j) \tag{56}$$

$$\Longrightarrow \mathbb{E}_{\alpha_1,\dots,\alpha_n,\delta_1,\dots,\delta_m | \bar{\delta} \in \bar{D}} \quad \text{holds} \quad \text{(theorem 1)} \tag{57}$$

$$\Longrightarrow \mathbb{E}_{\alpha_1,\dots,\alpha_n,\delta_1,\dots,\delta_m} \quad \text{holds} \quad \text{(theorem 2)} \tag{58}$$

□

Theorem 4. *Given a parameterised family of equations $\mathbb{E}$, where the !-boxes are separated and well nested, we can construct a finite set of simple equations $\{\mathbb{E}_\kappa\}_{\kappa \in K}$, such that:*

$$\{\mathbb{E}_\kappa\}_{\kappa \in K} \quad \textit{holds} \implies \mathbb{E} \quad \textit{holds} \tag{59}$$

The proof can be found in appendix A.3. The idea of the proof is to iteratively remove dependencies on !-boxes via theorem 2, each time generating a larger set of verifying equations. Once we have removed all !-box dependence we then use theorem 3 to remove phase variable dependence; using the "largest" equation in $\{\mathbb{E}_\kappa\}$ to determine the sizes of the A_j. We argue by the finiteness of all the parts involved that this process terminates.

We did not specify in the statement of our theorem which method of !-box expansion we were following, and indeed both methods work and are covered in the proof.

6 Summary

We have shown how to construct finite sets of equations that verify certain classes of infinite families of equations. In some situations this will make verifying theorems significantly easier for people, but also paves the way for proof assistants to verify these theorems. Further work would be to implement such methods into proof assistants, such as Quantomatic [8], so that the verifying set could be generated (and ideally checked) automatically.

Although we gave examples from ZX, $ZW_{\mathbb{C}}$ and ZH the techniques apply to any language satsifying the conditions given alongside theorems 1 and 2. Our requirement of the diagrams being separated (or separable) is linked to the bialgebraic interactions between nodes in the graphical language. As such it appears to be a fundamental aspect of the language, and further work could be to show whether there are inseparable equations beyond the bialgebra law that non-trivially hold true.

One final avenue is to develop methods to deal with the side conditions of the (EU) rule of the ZX calculus ([14],) either by extending these results or finding a presentation of the ZX calculus that does not require side conditions. It should be noted that [7] gives some evidence that such a presentation may not exist.

Acknowledgements: The author would like to thank their supervisors Aleks Kissinger and Bob Coecke as well as Miriam Backens for their help in writing this paper, and the EPSRC for providing the funding grant.

References

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

[2] B. Coecke & A. Kissinger (2017): *Picturing Quantum Processes*. Cambridge University Press.

[3] Bob Coecke & Ross Duncan (2008): *Automata, Languages and Programming: 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II*, chapter Interacting Quantum Observables, pp. 298–310. Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-540-70583-3_25. Available at `http://dx.doi.org/10.1007/978-3-540-70583-3_25`.

[4] Niel de Beaudrap & Dominic Horsman (2017): *The ZX calculus is a language for surface code lattice surgery*. arXiv e-prints:arXiv:1704.08670.

[5] Craig Gidney & Austin G. Fowler (2018): *Efficient magic state factories with a catalyzed* $|CCZ >$ *to* $2|T >$ *transformation*. arXiv e-prints:arXiv:1812.01238.

[6] Amar Hadzihasanovic (2015): *A Diagrammatic Axiomatisation for Qubit Entanglement*. arXiv e-prints:arXiv:1501.07082.

[7] Emmanuel Jeandel, Simon Perdrix & Renaud Vilmart (2018): *Diagrammatic Reasoning beyond Clifford+T Quantum Mechanics*. arXiv e-prints:arXiv:1801.10142.

[8] A. Kissinger & V. Zamdzhiev (2015): *Quantomatic: A Proof Assistant for Diagrammatic Reasoning*. ArXiv e-prints:1503.01034.

[9] Aleks Kissinger, Alex Merry & Matvey Soloviev (2012): *Pattern Graph Rewrite Systems*. In: DCM.

[10] Saunders MacLane (1965): *Categorical algebra*. Bull. Amer. Math. Soc. 71(1), pp. 40–106. Available at `https://projecteuclid.org:443/euclid.bams/1183526392`.

[11] Thomas Sauer Mariano Gasca (2000): *On the history of multivariate polynomial interpolation.* Journal of Computational and Applied Mathematics 122, pp. 23–35, doi:https://doi.org/10.1016/S0377-0427(00)00353-8. Available at `https://www.sciencedirect.com/science/article/pii/S0377042700003538`.

[12] Alexander Merry (2014): *Reasoning with !-Graphs*. CoRR abs/1403.7828. Available at `http://arxiv.org/abs/1403.7828`.

[13] K. F. Ng & Q. Wang (2017): *A universal completion of the ZX-calculus*. ArXiv e-prints:1706.09877.

[14] Renaud Vilmart (2018): *A Near-Optimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics*. arXiv e-prints:arXiv:1812.09114.

[15] Eric W. Weisstein: *Laurent Polynomial.* Available at `http://mathworld.wolfram.com/LaurentPolynomial.html`. From MathWorld–A Wolfram Web Resource.

A Proofs

We show here any proofs that were not included in the main text.

A.1 Proof of theorem 1

Theorem 1. *For a diagrammatic equation without !-boxes*

$$\{ \quad \mathbb{D}_1 = \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n}$$

that has a Laurent polynomial matrix interpretation, and the interpretations agree on a large enough grid of points $(a_1,\ldots,a_n) \in A_1 \times \cdots \times A_n$ *in the parameter space, then the interpretations agree on all values of* $(a_1,\ldots,a_n)$*.*

$$\begin{aligned} &\left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n|\alpha_1=a_1,\ldots,\alpha_n=a_n} \right]\!\right] = \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n|\alpha_1=a_1,\ldots,\alpha_n=a_n} \right]\!\right] \\ &\qquad \forall a_1 \in A_1,\ldots,a_n \in A_n \\ \Longrightarrow &\left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n|\alpha_1=a_1,\ldots,\alpha_n=a_n} \right]\!\right] = \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n|\alpha_1=a_1,\ldots,\alpha_n=a_n} \right]\!\right] \\ &\qquad \forall a_1,\ldots,a_n \end{aligned} \tag{35}$$

The size of $|A_j|$ *(corresponding to* α_j*) is given by:*

$$\begin{aligned} |A_j| = &\max(deg^+_{Y_j}[\,\mathbb{D}_1\,], deg^+_{Y_j}[\,\mathbb{D}_2\,]) \\ &+ \max(deg^-_{Y_j}[\,\mathbb{D}_1\,], deg^-_{Y_j}[\,\mathbb{D}_2\,]) \\ &+ 1 \end{aligned} \tag{36}$$

"The maximum positive degree on either side, plus the maximum negative degree on either side, plus one."

Proof. We are seeking the multivariate complex polynomials that populate the matrices $[\![\,\mathbb{D}_1\,]\!]$ and $[\![\,\mathbb{D}_2\,]\!]$. We begin by combining the two matrices of Laurent polynomials into one matrix of (not-Laurent) polynomials and a scale factor of the form $Y_1^{m_1}\ldots Y_n^{m_n}$.

- We define:

$$M_1 := [\![\,\mathbb{D}_1\,]\!] \tag{60}$$
$$M_2 := [\![\,\mathbb{D}_2\,]\!] \tag{61}$$

  and wish to show $M_1 = M_2$.

- First we pull enough copies of $Y_1^{-1},\ldots,Y_n^{-1}$ out of each side so that we have an equation of the form:

$$M_1' \prod_j (Y_j^{-1})^{\deg^-_{Y_j}[\,M_1\,]} = M_2' \prod_j (Y_j^{-1})^{\deg^-_{Y_j}[\,M_2\,]} \tag{62}$$

  Where M_1' and M_2' are matrices of (not-Laurent) polynomials.

- Let $m_j := \max(\deg^-_{Y_j}[\,M_1\,], \deg^-_{Y_j}[\,M_2\,])$ and multiply both sides by $\prod_j Y_j^{m_j}$ to clear any negative powers of Y_j.

$$M_1' \prod_j Y^{m_j - \deg^-_{Y_j}[\,M_1\,]} = M_2' \prod_j Y^{m_j - \deg^-_{Y_j}[\,M_2\,]} \tag{63}$$

- Then subtract the right hand side from the left:

$$M_1' \prod_j Y_j^{m_j - \deg^-_{Y_j}[\,M_1\,]} - M_2' \prod_j Y^{m_j - \deg^-_{Y_j}[\,M_2\,]} = 0 \tag{64}$$

  And define $\mathbb{M} := M_1' \prod_j Y^{m_j - \deg^-_{Y_j}[\,M_1\,]} - M_2' \prod_j Y^{m_j - \deg^-_{Y_j}[\,M_2\,]}$, which is a matrix of (not-Laurent) polynomials. The statement $\mathbb{M} = 0$ can be viewed as a stating that each of its polynomial entries are equal to the 0 polynomial.

- We will use the notation $\deg_{Y_j}[\,\cdot\,]$ for the degree of a (not-Laurent) polynomial, or matrix of not-Laurent polynomials. This is the same as its positive degree, but since people are more familiar with (not-Laurent) polynomials we want to make it clear when we are in the more familiar setting.

- We wish to find a bound for the maximum degree of any polynomial in $\mathbb{M}$:

$$\deg_{Y_j}[\,\mathbb{M}\,] = \deg_{Y_j}\left[M_1' \prod_j Y^{m_j - \deg^-_{Y_j}[\,M_1\,]} - M_2' \prod_j Y^{m_j - \deg^-_{Y_j}[\,M_2\,]} \right] \tag{65}$$

$$\leq \max(\deg_{Y_j}\left[\,M_1'\,\right] + m_j - \deg^-_{Y_j}[\,M_1\,], \quad \deg_{Y_j}\left[\,M_2'\,\right] + m_j - \deg^-_{Y_j}[\,M_2\,]) \tag{66}$$

$$= \max(\deg^+_{Y_j}[\,M_1\,] + \deg^-_{Y_j}[\,M_1\,] + m_j - \deg^-_{Y_j}[\,M_1\,], \quad \deg^+_{Y_j}[\,M_2\,] + \deg^-_{Y_j}[\,M_2\,] + m_j - \deg^-_{Y_j}[\,M_2\,]) \tag{67}$$

$$= \max(\deg^+_{Y_j}[\,M_1\,] + m_j, \quad \deg^+_{Y_j}[\,M_2\,] + m_j) \tag{68}$$

$$= \max(\deg^+_{Y_j}[\,M_1\,], \quad \deg^+_{Y_j}[\,M_2\,]) \quad + m_j \tag{69}$$

$$= \max(\deg^+_{Y_j}[\,M_1\,], \deg^+_{Y_j}[\,M_2\,]) + \max(\deg^-_{Y_j}[\,M_1\,], \deg^-_{Y_j}[\,M_2\,]) \tag{70}$$

- Suppose we know that our diagram equation held on a large enough *regular grid* of values for the α_j.

(A technique that appears to date to before the 20th century, according to [11].)

$$\left[\!\left[\{\quad \mathbb{D}_1 \quad\}_{\alpha_1,\ldots,\alpha_n|\alpha_1=a_1,\ldots,\alpha_n=a_n}\right]\!\right] = \left[\!\left[\{\quad \mathbb{D}_2 \quad\}_{\alpha_1,\ldots,\alpha_n|\alpha_1=a_1,\ldots,\alpha_n=a_n}\right]\!\right] \tag{71}$$

$$\begin{aligned} \text{for} \quad & (a_1,\ldots,a_n) \in A_1 \times \cdots \times A_n \\ \text{where} \quad & |A_j| = \deg_{Y_j}(\mathbb{M}) + 1 \\ A_j \quad & := \{a_{j,0}\ ,\ldots,\ a_{j,\deg Y_j}\} \end{aligned}$$

$$\begin{aligned} \text{i.e.} \quad & \forall P \text{ a polynomial entry of } \mathbb{M} \\ & P(a_1,\ldots,a_n) = 0 \end{aligned}$$

By picking a polynomial entry P of $\mathbb{M}$, expressing P using the multi-index β as $P = \sum_\beta c_\beta Y_j^\beta$, and then evaluating P at every point in $A_1 \times \cdots \times A_n$ we construct the system of equations:

$$\begin{bmatrix} c_{0,\ldots,0}a_{0,\ldots,0}^{0,\ldots,0} & c_{1,\ldots,0}a_{0,\ldots,0}^{1,\ldots,0} & \cdots & c_{\deg Y_1,\ldots,\deg Y_n}a_{0,\ldots,0}^{\deg Y_1,\ldots,\deg Y_n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{0,\ldots,0}a_{|A_1|,\ldots,|A_n|}^{0,\ldots,0} & c_{1,\ldots,0}a_{|A_1|,\ldots,|A_n|}^{1,\ldots,0} & \cdots & c_{\deg Y_1,\ldots,\deg Y_n}a_{|A_1|,\ldots,|A_n|}^{\deg Y_1,\ldots,\deg Y_n} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{72}$$

Which we view as:

$$\begin{bmatrix} V \end{bmatrix} \begin{bmatrix} c_{0,\ldots,0} \\ \vdots \\ c_{\deg Y_1,\ldots,\deg Y_n} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{73}$$

Where V contains all the products $a_1^{\beta_1} \times \cdots \times a_n^{\beta_n}$, β ranging from $(0,\ldots,0)$ to $(\deg Y_1,\ldots,\deg Y_n)$. Thankfully V decomposes as:

$$V = V_1 \otimes \ldots \otimes V_n \tag{74}$$

$$V_j = \begin{bmatrix} a_{j,0}^0 & a_{j,0}^1 & \cdots & a_{j,0}^{\deg Y_j} \\ a_{j,1}^0 & a_{j,1}^1 & \cdots & a_{j,1}^{\deg Y_j} \\ \vdots & \vdots & & \vdots \\ a_{j,\deg Y_j}^0 & a_{j,\deg Y_j}^1 & \cdots & a_{j,\deg Y_j}^{\deg Y_j} \end{bmatrix} \tag{75}$$

- Since $\det(A \otimes B) \neq 0$ if and only if $\det(A) \neq 0$ and $\det(B) \neq 0$, and since $\det(V_j) \neq 0$ because each V_j is a Vandermonde matrix, we know that $\det(V) \neq 0$. Since V is therefore invertible we know that all the coefficiencts c_β must be 0, and therefore P is the 0 polynomial.

- In the presence of a regular grid on which $\mathbb{D}_1$ and $\mathbb{D}_2$ agree we know:

$$
\begin{aligned}
& \left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n | \alpha_1 \in A_1,\ldots,\alpha_n \in A_n} \right]\!\right] = \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n | \alpha_1 \in A_1,\ldots,\alpha_n \in A_n} \right]\!\right] && (76)\\
\implies \quad & P = 0 \quad \text{for any entry } P \text{ of } M && (77)\\
\implies \quad & \mathbb{M} = 0 && (78)\\
\implies \quad & M_1' \prod_j Y^{m_j - \deg^-_{Y_j}[\, M_1 \,]} - M_2' \prod_j Y^{m_j - \deg^-_{Y_j}[\, M_2 \,]} = 0 && (79)\\
\implies \quad & M_1' \prod_j Y^{m_j - \deg^-_{Y_j}[\, M_1 \,]} = M_2' \prod_j Y^{m_j - \deg^-_{Y_j}[\, M_2 \,]} && (80)\\
\implies \quad & M_1 = M_2 && (81)\\
\implies \quad & \left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n} \right]\!\right] = \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n} \right]\!\right] && (82)
\end{aligned}
$$

- By setting d_j to $\deg_{Y_j}[\, \mathbb{M} \,] + 1$ (which we can calculate using equation (70)), and $|A_j| = d_j$ we attain our result; that if we know that the interpretations of the families of diagrams agree on the regular grid described by the sizes d_j then the interpretations agree on all points in the phase group.

□

A.2 Proof of theorem 2

Theorem 2. *Given a family* $\mathbb{E} = \{ \quad \mathbb{D}_1 = \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m}$ *of diagrammatic equations, parameterised by a !-box* δ_1 *where* δ_1 *is separated from all other !-boxes and is nested in no other !-box; then E is verified by the finite family* $\{\mathbb{E}|_{\delta_1=0}, \ldots, \mathbb{E}|_{\delta_1=N}\}$ *where N is the sum of the dimensions of the joins between* δ_1 *and the rest of the diagram. That is:*

$$
\begin{aligned}
& \left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m | \alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \\
= & \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m | \alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \qquad \forall d_1 \leq N \\
\implies & \left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m | \alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \\
= & \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m | \alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \qquad \forall d_1
\end{aligned}
$$

$$
\begin{aligned}
N &:= \dim\left(\mathbb{H}^{\otimes n_1} \oplus \mathbb{H}^{\otimes n_2}\right) \\
n_1 &:= \textit{number of wires between } \delta_1 \textit{ and the rest of } \mathbb{D}_1 \\
n_2 &:= \textit{number of wires between } \delta_1 \textit{ and the rest of } \mathbb{D}_2
\end{aligned}
$$

The idea of the proof is:

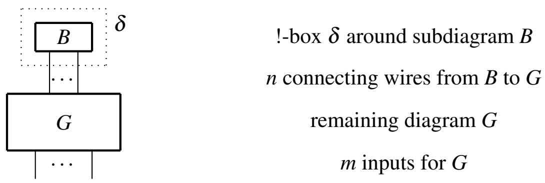
1. Manipulate the diagrams into what we call **series !-box form**
2. Move to the matrix interpretation
3. Manipulate the equation between two matrices into an expression on a single vector space of dimension N
4. Demonstrate the required property as a condition on subspaces

We will require the "only topology matters" meta-rule for our diagrams, suitable spider laws, a matrix interpretation, and finite dimensionality of $\mathbb{H}$.

Proof. We begin by showing series !-box form on a diagram containing a single !-box:

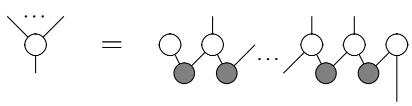
$$\{ \quad \mathbb{D} \quad \}_\delta \tag{83}$$

First we will manipulate the diagram (via "only topology matters") until it is in the following form:

$\mathbb{D} \quad =$ 

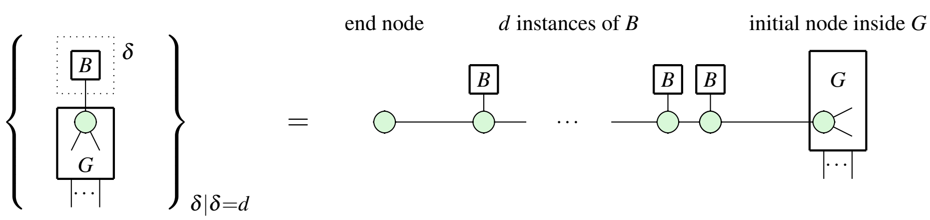
(84)

We note that the nodes inside G that join with B must be spiders; since there can be arbitrary many instances of B the node in G must be able to have arbitrary arity. There may also be p boundaries that are internal to B (and therefore δ,) which we will deal with momentarily. We will now rely on the existence of a spider law such that we may do the following:


(85)

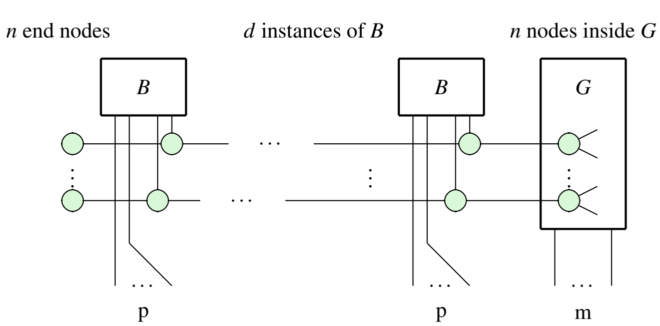
Which is the ability to "spread" a spider with n outputs into n repeated copies of a spider with 1 output, with suitable initial, terminal, and joining subdiagrams. We will also insist that such a spider has a non-zero interpretation. We do this so that each instance of the !-box is connected to its own copy of the spider, and these copies are joined in sequence.

This is possible in ZX, ZH and ZW. To give a ZX example (where the spider law is simple:)



(86)

Note that although we have only used one example node and joining wire we can perform this action on all nodes and joining wires. Where two wires travel from the !-box to the same spider inside G we first spread out that spider so each wire from the !-box connects to a different spider in G. Here is an example for n wires between B and G, and p boundaries inside B (which we stretch down to be below each copy of B in this representation, just for visibility.)



(87)

From here it is easy to see that we have the diagram $G : m \to n$, beside d copies of a diagram we call $B : p+n \to n$ (containing the p boundary nodes and the new connecting spiders), and finally an ending diagram $C : n \to 0$. We call this the **series !-box form**.

Definition 15. ***Series !-box form*** *for a given (non-nested, separated) !-box* $\delta_1 = d$ *in a diagram is a presentation (as in equation 87) of each the* δ_1*-instantiated diagrams as*

$C :=$ *the end cap of spiders*

$B :=$ *repeated element, which may contain* $\alpha_1, \ldots, \alpha_n, \delta_2, \ldots, \delta_m$*, and some boundary nodes*

$G :=$ *the rest of the diagram outsie of B, which may contain* $\alpha_1, \ldots, \alpha_n, \delta_2, \ldots, \delta_m$*,*
and some boundary nodes

Such that the d instances of δ_1 *are spread out as d instances of B. In the case where you consider a !-box to create child instances of parameters then B will contain children of the* α_j *and* $\delta_{k>1}$*, rather than copies.*

Claim: For any value of d we can put $\mathbb{D}\Big|_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\delta_1=d}$ into series !-box form as in the example above.

We will just use the variable names $\alpha_1, \ldots, \alpha_n, \delta_1, \ldots, \delta_m$ here and assume we are copying parameters, but the technique is identical for when one is creating child parameters and we will point out the different intricacies along the way. We aim to show

$$\begin{aligned} & \left[\!\left[\left\{ \quad \mathbb{D}_1 \quad \right\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \\ = & \left[\!\left[\left\{ \quad \mathbb{D}_2 \quad \right\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \qquad \forall d_1 \leq N \\ \Longrightarrow & \left[\!\left[\left\{ \quad \mathbb{D}_1 \quad \right\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \\ = & \left[\!\left[\left\{ \quad \mathbb{D}_2 \quad \right\}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\alpha_1=a_1,\ldots,\delta_1=d_1,\ldots\delta_m=d_m} \right]\!\right] \qquad \forall d_1 \end{aligned} \tag{88}$$

$N := \dim\left(\mathbb{H}^{\otimes n_1} \oplus \mathbb{H}^{\otimes n_2}\right)$

$n_1 :=$ number of wires between δ_1 and the rest of $\mathbb{D}_1$

$n_2 :=$ number of wires between δ_1 and the rest of $\mathbb{D}_2$

We would like to move directly to the matrix interpretation of diagram (87), but we have the following problems:

- the p dangling wires from every copy of B
- the parameters inside every copy of B (either linked copies or discrete children)

We solve these problems (and justify these solutions below) by considering B as parameterised by:

- Instances of $\alpha_j \; \forall j$
- Instances of $\delta_k \; \forall k > 1$
- Input vectors $v \in \mathbb{H}^{\otimes p}$ that "plug" the inputs inside B.

Since we can show equivalence of complex matrices by showing that they perform the same operation on any input, we need to show that for any choice of α_j, $\delta_{k>1}$ and for any input that equation (88) holds. (And that if we are creating child instances of parameters then these equations hold for any choice of each of those independently.) Once we have specified values of α_j, $\delta_{k>1}$ we may use our matrix interpretation to get a complex matrix, but we need to do this for every possible choice of α_j, $\delta_{k>1}$.

Assuming we have made choices for the α_j and $\delta_{k>1}$, we wish to justify that we can choose the input vector for the p inputs of B independently. To do this we note that we can determine equality of complex matrices by showing they perform the same operation on all basis elements.

Claim: The set of all vectors of the form $\{v_d \otimes \ldots \otimes v_1 \otimes x\}$ where $v_j \in \mathbb{H}^{\otimes p}$ and $x \in \mathbb{H}^{\otimes m}$ contains a basis for $\mathbb{H}^{\otimes(m+dp)}$.

We show this by noting that we may form a basis for $V \otimes V'$ by taking the tensor products of the bases of V and V', and therefore the above set contains all the basis elements of $\mathbb{H}^{\otimes p} \otimes \ldots \mathbb{H}^{\otimes p} \otimes \mathbb{H}^{\otimes m} \cong \mathbb{H}^{\otimes(m+dp)}$.

Given a choice of values for the α_j, $\delta_{k>1}$ and v we denote this choice by q and use B_q to mean "the sub-diagram B from the series !-box form with this choice of variables". If one is copying variable names then these values must be the same in each copy of B_q, but we will show the more general case of when you cannot assume that each instance of B_q contains the same choices of values for variables. (Even when creating children of the α_j and $\delta_{k>1}$ these children are contained entirely inside each instance of B so specifying a choice for q for each B determines all the values for parameters inside B.)

The last thing to define here is G_q. G_q is the choice of parameter values inside G, same as for B_q, but we ignore the vector v component of q. Once we have chosen values for q we may consider the matrix interpretation of the diagram:

$$\llbracket C \rrbracket \llbracket B_{q_d} \rrbracket \ldots \llbracket B_{q_1} \rrbracket \llbracket G_{q_o} \rrbracket \tag{89}$$
$$G_q : \mathbb{H}^{\otimes m} \to \mathbb{H}^{\otimes n} \tag{90}$$
$$B_q : \mathbb{H}^{\otimes n} \to \mathbb{H}^{\otimes n} \tag{91}$$
$$C : \mathbb{H}^{\otimes n} \to \mathbb{C} \tag{92}$$

Given an equation $\mathbb{D}_1 = \mathbb{D}_2$ of two families of diagrams, both parameterised by a (non-nested, separated) !-box δ_1 (among other parameters) we wish to remove our dependence on δ_1 by instead verifying a finite set of equations, each of which has a fixed value $\delta_1 = d$. Note that for this to be the case we require the number of inputs to be equal; i.e. $m_1 = m_2 = m$ and $p_1 = p_2 = p$, but we do not require $n_1 = n_2$ in equation 87. With the aim of reducing notational clutter we instantiate $\delta_1 = d$ and express D_1 and D_2 in series !-box form, with matrix interpretations:

$$\llbracket C_1 \rrbracket \llbracket B_{1,q_d} \rrbracket \ldots \llbracket B_{1,q_1} \rrbracket \llbracket G_{1,q_0} \rrbracket \tag{93}$$
$$\llbracket C_2 \rrbracket \llbracket B_{2,q_d} \rrbracket \ldots \llbracket B_{2,q_1} \rrbracket \llbracket G_{2,q_0} \rrbracket \tag{94}$$

And we wish to know when these two interpretations are equal. Rather than consider the matrices acting on two independent spaces we view them as acting on the direct sum of those two spaces and represent these maps as block matrices. (We drop the $\llbracket \cdot \rrbracket$ notation when it would appear inside a matrix.)

$$
\begin{aligned}
&\llbracket C_1 \rrbracket \llbracket B_{1,q_d} \rrbracket \dots \llbracket B_{1,q_1} \rrbracket \llbracket G_{1,q_0} \rrbracket \\
&= \llbracket C_2 \rrbracket \llbracket B_{2,q_d} \rrbracket \dots \llbracket B_{2,q_1} \rrbracket \llbracket G_{2,q_0} \rrbracket && \forall d,q \qquad (95) \\
\iff &\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \dots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0 && \forall d,q \qquad (96)
\end{aligned}
$$

Think of this as copying an input vector $x \in \mathbb{H}^m$ as $x :: x$ in $\mathbb{H}^m \oplus \mathbb{H}^m$, then applying

$$\llbracket C_1 \rrbracket \llbracket B_{1,q_d} \rrbracket \dots \llbracket B_{1,q_1} \rrbracket \llbracket G_{1,q_0} \rrbracket$$

and

$$\llbracket C_2 \rrbracket \llbracket B_{2,q_d} \rrbracket \dots \llbracket B_{2,q_1} \rrbracket \llbracket G_{2,q_0} \rrbracket$$

to the left and right copies respectively. After that we apply a minus sign to the right hand result and add that to the left hand result, effectively comparing them and demanding the difference to be 0. We seek to prove:

$$
\begin{aligned}
&\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \dots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0 \quad \forall d \leq N, q \\
\implies &\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \dots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0 \qquad \forall d,q \quad (97)
\end{aligned}
$$

Recalling that q is the choice of values for α_j, $\delta_{k>1}$ and $v \in \mathbb{H}^{\otimes p}$, we use Q to denote the set of all possible choices. We use B'_q for the matrix that acts as the direct sum of $B_{1,q}$ and $B_{2,q}$:

$$B'_q := \begin{bmatrix} B_{1,q} & 0 \\ 0 & B_{2,q} \end{bmatrix} \tag{98}$$

and inductively define the spaces:

$$V_0 := \mathrm{span}\{\bigcup_{q \in Q} \mathrm{Im}(\begin{bmatrix} G_{1,q} & 0 \\ 0 & G_{2,q} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix})\} \tag{99}$$

$$V_j := \mathrm{span}\{V_{j-1} \cup \bigcup_{q \in Q} B'_q V_{j-1}\} \tag{100}$$

The V_j form an ascending sequence of subspaces, each containing the potential images of up to j applications of B'_q:

$$V_j \geq \mathrm{Im}(\, B'_{q_k} \dots B'_{q_1} V_0 \,) \quad \forall k \leq j \, \forall q_k, \dots, q_1 \in Q \tag{101}$$

Claim: There is a number b such that

- if $j < b$ then $V_j > V_{j-1}$
- if $j \geq b$ then $V_j = V_{j-1}$
- $b \leq \dim(\, \mathbb{H}^{n_1} \oplus \mathbb{H}^{n_2} \,)$

We define $N := \dim(\mathbb{H}^{n_1} \oplus \mathbb{H}^{n_2})$ and show this by noting.

- $V_{j-1} \leq V_j \quad \forall j$
- if $V_j = V_{j-1}$ then $V_{j+1} = V_j$
- $V_j \leq \mathbb{H}^{n_1} \oplus \mathbb{H}^{n_2} \quad \forall j$
- $\dim V_{j-1} \leq \dim V_j \leq N \quad \forall j$
- The strictly increasing section of the sequence of the $\dim V_j$ must have length less than N
- We declare b to be the number such that $V_c = V_b \; \forall c \geq b$, and note $b \leq N$

Let W be the kernal of the map $\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}$

Claim:

$$\begin{aligned} & V_j \leq W && \forall j \leq N \\ \Longrightarrow & V_j \leq W && \forall j \end{aligned} \tag{102}$$

Since $V_c = V_N$ when $c \geq N \geq b$ it is enough to show that this is the case for all V_j when $j \leq N$. This is implied by the assumption in our theorem; that for $d \leq N$ our diagrammatic equation holds, and so for any choice of d and $q_1, \ldots, q_d$ this matrix equation holds:

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \cdots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0 \tag{103}$$

We have shown that for any choice of inputs $v \in \mathbb{H}^p \otimes \ldots \otimes \mathbb{H}^p \otimes \mathbb{H}^m$, and parameters α_j, $\delta_{k>1}$ our matrix equations hold, and by extension they hold on all elements of the space $\mathbb{H}^{\otimes(m+dp)}$, i.e. that:

$$\begin{aligned} & \forall d \leq N \, \forall q_d \ldots q_1 \\ & \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \cdots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0 && (104) \\ \Longrightarrow & \forall d \forall q_d \ldots q_1 \\ & \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} B_{1,q_d} & 0 \\ 0 & B_{2,q_d} \end{bmatrix} \cdots \begin{bmatrix} B_{1,q_1} & 0 \\ 0 & B_{2,q_1} \end{bmatrix} \begin{bmatrix} G_{1,q_0} & 0 \\ 0 & G_{2,q_0} \end{bmatrix} \begin{bmatrix} \mathrm{id}_m \\ \mathrm{id}_m \end{bmatrix} = 0 && (105) \end{aligned}$$

And therefore:

$$[\![C_1]\!] [\![B_{1,q_d}]\!] \ldots [\![B_{1,q_1}]\!] [\![G_{1,q_0}]\!] = [\![C_2]\!] [\![B_{2,q_d}]\!] \ldots [\![B_{2,q_1}]\!] [\![G_{2,q_0}]\!] \qquad \forall d \leq N \, \forall q_d \ldots q_1 \tag{106}$$

$$\Longrightarrow [\![C_1]\!] [\![B_{1,q_d}]\!] \ldots [\![B_{1,q_1}]\!] [\![G_{1,q_0}]\!] = [\![C_2]\!] [\![B_{2,q_d}]\!] \ldots [\![B_{2,q_1}]\!] [\![G_{2,q_0}]\!] \qquad \forall d \, \forall q_d \ldots q_1 \tag{107}$$

And therefore for any choice of value for α_j and $\delta_{k>1}$:

$$\left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\delta_1=d} \right]\!\right] = \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\delta_1=d} \right]\!\right] \qquad \forall d \leq N \tag{108}$$

$$\Longrightarrow \left[\!\left[\{ \quad \mathbb{D}_1 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\delta_1=d} \right]\!\right] = \left[\!\left[\{ \quad \mathbb{D}_2 \quad \}_{\alpha_1,\ldots,\alpha_n,\delta_1,\ldots,\delta_m|\delta_1=d} \right]\!\right] \qquad \forall d \tag{109}$$

□

A.3 Proof of theorem 4

Theorem 4. *Given a parameterised family of equations $\mathbb{E}$, where the !-boxes are separated and well nested, we can construct a finite set of simple equations $\{\mathbb{E}_\kappa\}_{\kappa\in K}$, such that:*

$$\{\mathbb{E}_\kappa\}_{\kappa\in K} \quad \textit{holds} \Longrightarrow \mathbb{E} \quad \textit{holds} \tag{59}$$

Proof. We first show the existence of an ordered list of the !-boxes present in E, compatible with the nesting order on both of the nesting diagrams of E.

Definition 16. *We construct the ordered list $\delta_{k_1} \succ \delta_{k_2} \succ \ldots$ by:*

We choose a !-box nested in no other !-boxes, add that !-box to the end of our list, then consider the nesting diagram with that !-box removed to pick the next entry in our list.

Definition 17. *!-Removal:*

- ***If we are copying variable names:*** *Given a set of equations $\{\mathbb{E}_\kappa\}_{\kappa\in K}$, and a !-box δ_k nested in no other !-boxes present in the $\mathbb{E}_\kappa$:*

  We define !-Remove(δ_k) as the process described in theorem 2. It acts on the set $\{\mathbb{E}_\kappa\}_{\kappa\in K}$ by acting on each of the $\mathbb{E}_\kappa$ in turn, finding the value N_κ, and creating the new verification set:

$$\{\mathbb{E}_\kappa\}_{\kappa\in K'} := \bigcup_{\kappa\in K} \{\mathbb{E}_\kappa\Big|_{\delta_k=1}, \ldots, \mathbb{E}_\kappa\Big|_{\delta_k=N_\kappa}\} \tag{110}$$

- ***If we are creating child instances:*** *We do as above, but we pick the !-box δ_k such that all its child instances are nested in no other !-boxes present in the $\mathbb{E}_\kappa$, and we act not only on each of the $\mathbb{E}_\kappa$ in turn but also on each of the child instances of δ_k in turn.*

Claims:

- !-Remove(δ_k) removes any dependency on δ_k in the verifying set $\{\mathbb{E}_\kappa\}_{\kappa\in K'}$
- $\{\mathbb{E}_\kappa\}_{\kappa\in K'}$ verifies $\{\mathbb{E}_\kappa\}_{\kappa\in K}$ (by theorem 2)
- !-Remove(δ_k) does not alter the nesting ordering of any remaining !-boxes and phase variables in the verification pair
- The ordered list $\delta_{k_1} \succ \delta_{k_2} \succ \ldots$ provides us with a sequence of !-boxes such that we can apply !-Remove($\delta_{k_{n+1}}$) to the output of !-Remove(δ_{k_n}).
- This ordered sequence of !-Removes results in a finite verifying set that has no dependence on any !-box.

The third and fourth of these claims are easy when we are copying variable names, but when creating child instances of !-boxes below δ_k one should view instantiation as creating (distinctly named) copies of the nesting structure that exists below δ_k.

Removing phase variables is trickier than removing !-boxes, because we know by theorem 3 that to do so in a manner compatible with !-box removal we need to know the largest number of !-box instances we are going to instantiate for each !-box. Since we have already constructed the $\{\mathbb{E}_\kappa\}_{\kappa\in K}$ we can find the equation that resulted from every !-box δ_k being instantiated at its largest amount, N_k, and use that equation.

Definition 18. *We use $\mathbb{E}'$ to denote the !-box free equation that is the result of instantiating each !-box δ_k at its largest required amount N_k*

Claim: The equation $\mathbb{E}'$ contains the largest number of instances of α_j for any α_j.

Definition 19. *α-Removal: We construct the set A_j for variable α_j by considering the degree of α_j in $\mathbb{E}'$, and then choosing enough valid values of α_j to reach the amount dictated by theorem 1. We then form:*

$$\{\mathbb{E}_\kappa\}_{\kappa\in K'} := \bigcup_{\kappa\in K, a\in A_j} \{\mathbb{E}_\kappa\big|_{\alpha_j=a}\} \tag{111}$$

Claim: By theorem 3 $\{\mathbb{E}_\kappa\}_{\kappa\in K'}$ verifies $\{\mathbb{E}_\kappa\}_{\kappa\in K}$

And finally:

Claim: After applying !-Removes in the order dictacted by $\prec$ and then applying all possible α-Removes we construct a finite set $\{\mathbb{E}_\kappa\}$ of simple equations that verifies $\mathbb{E}$.

□

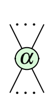
B Generators and degrees

We show here the generators of our three example languages (ZX, ZH and $\mathrm{ZW}_\mathbb{C}$), and then show the matrix degrees of these interpretations.

B.1 Generators

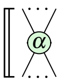
B.1.1 ZX

- The Z spider has interpretation in $\mathrm{Mat}_\mathbb{C}$:



$$\left[\!\left[\left\{ \quad \right\}\Big|_{\alpha=a} \right]\!\right] = \begin{bmatrix} 1 & 0 & \dots & & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 0 & 0 \\ 0 & \dots & & 0 & e^{ia} \end{bmatrix} \tag{112}$$

Making the substitution $Y := e^{i\alpha}$ (and therefore $Y^n = e^{in\alpha}$) the Z spider has this interpretation in $\mathrm{Mat}_{\mathbb{C}[Y,Y^{-1}]}$:



$$\left[\!\left[\; \right]\!\right] = \begin{bmatrix} 1 & 0 & \dots & & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 0 & 0 \\ 0 & \dots & & 0 & Y \end{bmatrix} \tag{113}$$

- The Hadamard node has interpretation:

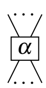
$$\left[\!\left[\; \right]\!\right] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{114}$$

  And admits no parameters.

- The X spider interpretation is found by applying Hadamards nodes on all inputs and outputs of the corresponding Z spider.

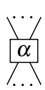
B.1.2 ZH

- The H box in ZH has interpretation in $\mathrm{Mat}_{\mathbb{C}}$ as:



$$\left[\!\left[\left\{ \; \right\}\Big|_{\alpha=a} \right]\!\right] = \begin{bmatrix} 1 & 1 & \dots & & 1 \\ 1 & 1 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 1 & 1 \\ 1 & \dots & & 1 & a \end{bmatrix} \tag{115}$$

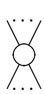
  We can simply equate $Y := \alpha$ and get the interpretation in $\mathrm{Mat}_{\mathbb{C}[Y,Y^{-1}]}$:



$$\left[\!\left[\; \right]\!\right] = \begin{bmatrix} 1 & 1 & \dots & & 1 \\ 1 & 1 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 1 & 1 \\ 1 & \dots & & 1 & Y \end{bmatrix} \tag{116}$$

  in $\mathrm{Mat}_{\mathbb{C}[Y,Y^{-1}]}$.

- The Z spider in ZH admits no parameters, and has interpretation:

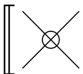


$$\left[\!\left[\; \right]\!\right] = \begin{bmatrix} 1 & 0 & \dots & & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 0 & 0 \\ 0 & \dots & & 0 & 1 \end{bmatrix} \tag{117}$$

B.1.3 ZW

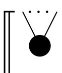
We are explicitly using $ZW_{\mathbb{C}}$ here for compatibility with the other languages.

- The Crossing x has interpretation



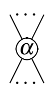
$$\llbracket \; \rrbracket \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{118}$$

- The W spider in bra-ket notation has interpretation



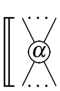
$$\llbracket \; \rrbracket = \sum_{k=1}^{n} |\underbrace{0\ldots0}_{k-1}1\underbrace{0\ldots0}_{n-k}\rangle \tag{119}$$

- The Z spider is parameterised by $\alpha \in \mathbb{C}$ and has interpretation



$$\left\llbracket \left\{ \; \right\}\Big|_{\alpha=a} \right\rrbracket = \begin{bmatrix} 1 & 0 & \ldots & & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 0 & 0 \\ 0 & \ldots & & 0 & a \end{bmatrix} \tag{120}$$

We can again simply equate $Y := \alpha$ and get the interpretation in $\mathrm{Mat}_{\mathbb{C}[Y,Y^{-1}]}$:



$$\llbracket \; \rrbracket = \begin{bmatrix} 1 & 0 & \ldots & & 0 \\ 0 & 0 & & & \vdots \\ \vdots & & \ddots & & \\ & & & 0 & 0 \\ 0 & \ldots & & 0 & Y \end{bmatrix} \tag{121}$$

B.2 degrees

We will consider nodes parameterised by a single variable α, and express their degree with respect to Y following the convention of appendix B.1.

B.2.1 ZX

Using $Y^n := e^{ni\alpha}$, the degrees in Y of the generators (for $n \geq 0$) are:

$$\deg^+_\alpha\left[\, n\alpha \,\right] = n \qquad \deg^-_\alpha\left[\, n\alpha \,\right] = 0 \tag{122}$$

$$\deg^+_\alpha\left[\, -n\alpha \,\right] = 0 \qquad \deg^-_\alpha\left[\, -n\alpha \,\right] = n \tag{123}$$

$$\deg^+_\alpha\left[\, n\alpha \,\right] = n \qquad \deg^-_\alpha\left[\, n\alpha \,\right] = 0 \tag{124}$$

$$\deg^+_\alpha\left[\, -n\alpha \,\right] = 0 \qquad \deg^-_\alpha\left[\, -n\alpha \,\right] = n \tag{125}$$

$$\deg^+_\alpha\left[\, \square \,\right] = 0 \qquad \deg^-_\alpha\left[\, \square \,\right] = 0 \tag{126}$$

B.2.2 ZH

We equate $Y := \alpha$, and for P any Laurent polynomial:

$$\deg^+_\alpha\left[\, \boxed{P(\alpha)} \,\right] = \deg^+_Y\left[\, P \,\right] \qquad \deg^-_\alpha\left[\, \boxed{P(\alpha)} \,\right] = \deg^-_Y\left[\, P \,\right] \tag{127}$$

$$\deg^+_\alpha\left[\, \circ \,\right] = 0 \qquad \deg^-_\alpha\left[\, \circ \,\right] = 0 \tag{128}$$

B.2.3 ZW

We equate $Y := \alpha$, and for P any Laurent polynomial:

$$\deg^+_\alpha\left[\, \bullet \,\right] = 0 \qquad \deg^-_\alpha\left[\, \bullet \,\right] = 0 \tag{129}$$

$$\deg^+_\alpha\left[\, \otimes \,\right] = 0 \qquad \deg^-_\alpha\left[\, \otimes \,\right] = 0 \tag{130}$$

$$\deg^+_\alpha\left[\, P(\alpha) \,\right] = \deg^+_Y\left[\, P \,\right] \qquad \deg^-_\alpha\left[\, P(\alpha) \,\right] = \deg^-_Y\left[\, P \,\right] \tag{131}$$

(132)