Hypergraph simplification: Linking the path-sum approach to the ZH-calculus

Louis Lemonnier
École Normale Supérieure Paris-Saclay
louis.lemonnier@ens-paris-saclay.fr

John van de Wetering
Radboud Universiteit Nijmegen
wetering@cs.ru.nl

Aleks Kissinger
Oxford University
aleks.kissinger@cs.ox.ac.uk

The ZH-calculus is a complete graphical calculus for linear maps between qubits that admits, for example, a straightforward encoding of hypergraph states and circuits arising from the Toffoli+Hadamard gate set. In this paper, we establish a correspondence between the ZH-calculus and the path-sum formalism, a technique recently introduced by Amy to verify quantum circuits. In particular, we find a bijection between certain canonical forms of ZH-diagrams and path-sum expressions. We then introduce and prove several new simplification rules for the ZH-calculus, which are in direct correspondence to the simplification rules of the path-sum formalism. The relatively opaque path-sum rules are shown to arise naturally as the convergence of two consequences of the ZH-calculus. The first is the extension of the familiar graph-theoretic simplifications based on local complementation and pivoting to their hypergraph-theoretic analogues: hyper-local complementation and hyper-pivoting. The second is the graphical Fourier transform introduced by Kuijpers et al., which enables effective simplification of ZH-diagrams encoding multi-linear phase polynomials with arbitrary real coefficients.

1 Introduction

The very nature of quantum computation makes it hard to verify classically that a given quantum circuit implements the desired computation without incurring exponential space or time costs. It is however still possible to develop smart heuristics that can verify that quantum circuits indeed implement the right unitary. One such heuristic is the *path-sum* approach [2]. It represents each quantum gate in the circuit by the action it has on the computational basis states, given by a Boolean function determining the output basis state and a semi-Boolean function giving relative phases of outputs, each of which can depend on inputs as well as auxiliary Boolean variables, or 'paths', that are summed over. Amy developed a set of simplification rules for these path-sums that were powerful enough to completely simplify a set of benchmark quantum circuits that implemented classical reversible functions to their classical specification. Each of these rewrite rules eliminates a variable from the path-sum, but beyond that, their interpretation is quite opaque.

In this paper we will see that path-sum expressions and the rewrite rules from [2] can be represented in a natural way using the *ZH-calculus*. The ZH-calculus is a graphical language recently introduced by Backens and Kissinger [4] that can easily represent computations involving Hadamard and Toffoli gates, and generalisations thereof. It comes with a set of graphical rewrite rules that are *complete*, meaning that any two diagrams representing equal linear maps can be graphically transformed into one another.

There are two key ingredients in our translation of the path-sum rewrite rules into the ZH-calculus. The first is based on the realisation that ZH-diagrams can easily represent *hypergraph states* [21, 19].

Submitted to:
QPL 2020

Graph states are a type of stabiliser state widely used in a variety of quantum protocols as well as the one-way model of measurement-based quantum computation [20]. Interestingly, the graph operations of *local complementation* and *pivoting* can be performed on the underlying graph of a graph state by applying local Cliffords [18]. We will show in this paper that the graph-theoretic simplifications for ZX-diagrams from [9] based on local complementation and pivoting extend naturally to hypergraph-theoretic simplifications on ZH-diagrams, which we call *hyper-local complementation* and *hyper-pivoting*.

The second ingredient is producing a ZH-calculus analogue to the operation of *lifting* of a Boolean polynomial Q to a related polynomial $\overline{Q}$ to enable substitution into a semi-boolean function. This operation plays an important role in the path-sum reductions, and it turns out to correspond to the *graphical Fourier transform* introduced by Kuijpers and two of the authors in [16].

By introducing a new connection between path-sums and hypergraph states, via the ZH-calculus, we provide not only a new perspective on verification for quantum circuits, but also new techniques that can be applied to measurement-based quantum computing models based on hypergraph states [12, 22, 11]. Computations could be analysed and verified using either the ZH-calculus or path-sums, similar to how the ZX-calculus can be used to analyse the one-way model of MBQC [5, 10].

Related work. This work is an extension of a technical report by one of the authors in 2019 [17]. Since then, parallel work of Vilmart has drawn a similar correspondence between the path-sum approach and the ZH-calculus [24]. That work differs from ours in two ways. First, its correspondence is semantical: i.e. it defines categories of ZH-diagrams and sum-over-paths expressions, modulo the appropriate laws and gives functors in either direction. Ours is syntactic: we establish a direct bijection between certain universal families of ZH-diagrams and path-sum expressions. Second, at the level of rewriting Ref. [24] focuses on the Clifford fragments of each of the two languages and proves completeness, whereas we aim to capture the full power the two languages in reasoning beyond Clifford computations, but like [2], our goal is to develop useful heuristics for diagram simplification rather than a complete procedure for deciding equality (which is a QMA-hard problem [6]).

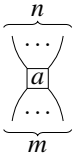
Much as the ZX-calculus is closely connected to the theory of graph states and the one-way model, the ZH-calculus is closely connected to the emerging theory of hypergraph states. In that context, a notion of local complementation for hypergraph states has been introduced by Gachechildaze et al. [13] which is closely related to the simplification we introduce in Section 4.1. In addition, a restricted version of the graphical Fourier transform was described for *weighted* hypergraph states in [23].

2 ZH-calculus

In this section we will recall the ZH-calculus, together with (annotated) !-box notation and the Fourier transform rule of [16].

The ZH-calculus is a diagrammatic language introduced by Backens and Kissinger [4] that represents linear maps as *ZH-diagrams*. These are string diagrams based on two generators: Z-spiders, depicted as white dots; and H-boxes, depicted as white boxes with a complex parameter a:

$:= |0\ldots0\rangle\langle 0\ldots0| + |1\ldots1\rangle\langle 1\ldots1|$

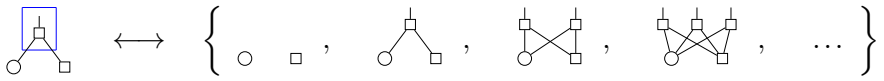


$:= \sum a^{i_1\ldots i_m j_1\ldots j_n} |j_1\ldots j_n\rangle\langle i_1\ldots i_m|$

Here in the right-hand equation the sum runs over all $i_1,\ldots,i_m,j_1,\ldots,j_n \in \{0,1\}$. Hence, an H-box represents a matrix with a as its $|1\ldots1\rangle\langle 1\ldots1|$ entry, and ones everywhere else. By convention, we omit

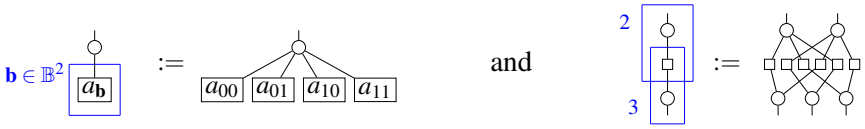
the parameter a when $a = -1$, and hence an unlabeled H-box with 1 input and 1 output is the conventional Hadamard gate (up to normalisation). More complex diagrams are constructed by composing these generators either by stacking them or by joining the outputs of the first with the inputs of the second, which correspond respectively to the tensor product and regular composition of linear maps.

Our calculations will greatly benefit from the usage of !-box – pronounced as "bang box" – notation [14]. A !-box, drawn as a blue square around a piece of a ZH-diagram, represents a part of the diagram that may be replicated an arbitrary number of times, and hence allows one to express a whole family of diagrams at once:

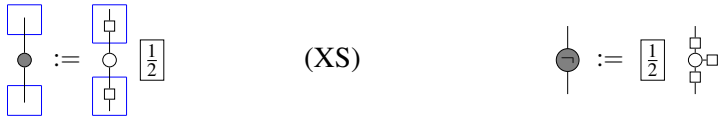


When used in equations, corresponding !-boxes on either side of the equation should be understood to be replicated an equal number of times.

As in [4] we will also use *annotated* !-boxes that are labelled by a set or a natural number to denote the number of copies of the diagram. For example, letting $\mathbb{B} = \{0,1\}$ denote the set of Booleans we have:

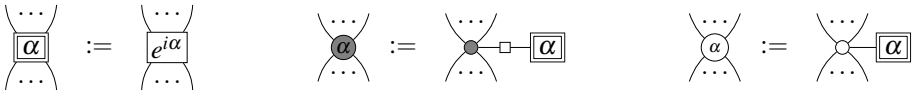


Note that in the right-hand diagram we had overlapping !-boxes resulting in a fully-connected bipartite graph of connectivity. Also following [4] we use some derived generators: the X-spider and the NOT gate.



The power of the ZH-calculus comes from the set of graphical rewrite rules associated to it. First of all, ZH-diagrams are considered equal when they can be topologically deformed into one another, as long as the order of in- and outputs is preserved [7, 8]. Second, there are a set of rewrite rules that can be applied to parts of a diagram. We present these standard rules in Figure 2 in Appendix A. These rules are *complete* meaning that if two ZH-diagrams represent the same linear map, then those diagrams can be transformed into one another using some application of these rules [4].

Some of our results will require the *Fourier transform* of a ZH-diagram as constructed in [16]. Before we introduce this Fourier transform, we recall some of the notation of [16]. First, there are the *exponentiated* H-boxes and the associated phase spiders, that allow us to make the connection between ZH-diagrams and path-sums more direct:



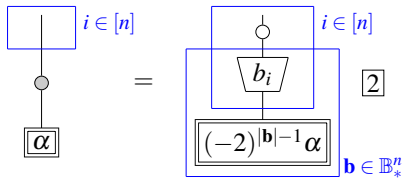
(1)

Second, there are the *disconnect boxes* from [16] that combine nicely with annotated !-boxes:

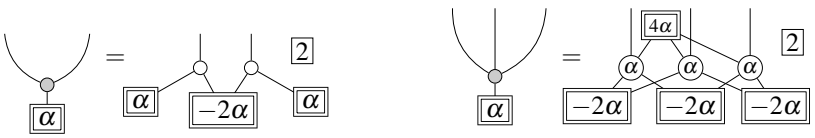
(2)

Let $[n] = \{1,\ldots,n\}$ denote the n element set and let $|\mathbf{b}|$ denote the *weight* of a bitstring $\mathbf{b} \in \mathbb{B}^n$, i.e. the number of 1's in $\mathbf{b}$. Denote by $\mathbb{B}^n_*$ the set of all non-zero bitstrings, i.e. $\mathbb{B}^n \setminus (0,\ldots,0)$.

Proposition 2.1. [16] The following Fourier-transform rule holds.



For example, in the $n = 2$ and $n = 3$ cases we have:



Remark 2.2. The reason we call this rule a Fourier transform is because it is closely related to the Fourier transform of semi-Boolean functions. See [16] for the details. We presented the rule using exponentiated H-boxes. A more general version using regular H-boxes (as long as the label is non-zero) also holds.

Before we continue onto our introduction of path-sums, it will be useful to introduce a new class of ZH-diagrams that generalises the definition of graph-like ZX-diagrams from [9].

Definition 2.3. We say a ZH-diagram is *hypergraph-like* when

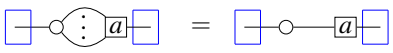
- all spiders are Z-spiders,
- every in- and output wire is connected only to a spider (so no connections directly to H-boxes),
- the only wires are between H-boxes and spiders,
- there is at most one wire between any given H-box and spider (so no parallel edges),
- and there are no H-boxes connected to exactly the same set of spiders.

We call these diagrams hypergraph-like, because most of their structure is captured by the underlying hypergraph that has as vertices the spiders and as hyperedges the H-boxes.

Definition 2.4. A *hypergraph* $G = (V, E)$ consists of a set of *vertices* V and a set of *hyperedges* E. Each hyperedge $e \in E$ is a non-empty set of vertices $e \subseteq V$. We call a hyperedge *simple* when it contains exactly two vertices. A *simple graph* is a hypergraph where every hyperedge is simple.

The underlying hypergraph of a hypergraph-like ZH-diagram is a simple graph iff all H-boxes have arity 2. Such diagrams are *graph-like* as defined in [9]. Every ZH-diagram can be reduced to a hypergraph-like ZH-diagram representing the same linear map. Before we prove this, we need a lemma.

Lemma 2.5. Multiple parallel edges between an H-box and a Z-spider can be reduced to a single edge.



(3)

Proof. Follows easily from the rule (DC) in Lemma 3.4 of [16]. □

Lemma 2.6. Every ZH-diagram can be efficiently transformed into a hypergraph-like ZH-diagram.

Proof. First transform all grey spiders and NOT gates into white spiders using the definitions (XS) and (N). Remove all double Hadamard gates this introduces with (HS2) — the equation label refers to Figure 2. Fuse all the spiders by applying (ZS1) repeatedly. Disconnect in- and outputs from H-boxes by introducing identities with (ZS2), and similarly introduce identities between H-boxes that are connected. Remove parallel edges between H-boxes and spiders with Lemma 2.5. Finally, fuse H-boxes that have the same set of neighbours using (M). □

Remark 2.7. The *weighted* hypergraphs of [23] associate phases to each hyperedge. The resulting weighted hypergraph states precisely correspond to hypergraph-like ZH-diagrams where every spider has exactly one output wire and there are no inputs.

3 Path-sums and pure path-sums

Path-sums give a compact way of representing the action of a linear map U on computational basis states in terms of two polynomial functions f and ϕ:

$$U :: |\boldsymbol{x}\rangle \mapsto \lambda \cdot \sum_{\boldsymbol{y}} e^{2\pi i \cdot \phi(\boldsymbol{x},\boldsymbol{y})} |f(\boldsymbol{x},\boldsymbol{y})\rangle \tag{4}$$

In this description $\lambda \in \mathbb{C}$ is a (typically irrelevant) global scalar factor; $\boldsymbol{x} \in \mathbb{B}^n$, $\boldsymbol{y} \in \mathbb{B}^k$ are bit strings which we will typically, by minor abuse of notation, treat as lists of variables; $f = (f_1, \ldots, f_m)$ where each $f_i \in \mathbb{B}[\boldsymbol{x},\boldsymbol{y}]$ is a Boolean (i.e. $\mathbb{F}_2$-valued) polynomial describing the i-th output basis state in terms of $\boldsymbol{x}$ and $\boldsymbol{y}$; and $\phi \in \mathbb{R}[\boldsymbol{x},\boldsymbol{y}]$ is a polynomial valued in the real numbers (or some sub-ring thereof) describing the phase of each summand, which is often referred to as the *phase polynomial*.

Example 3.1. The path-sum representation of the CNOT gate is $|x_1x_2\rangle \mapsto |x_1(x_1 \oplus x_2)\rangle$. Hence, we have $\phi = 1$, no path variables $\mathbf{y}$, and $f = (f_1, f_2)$ where $f_1 = \mathrm{id}$ and $f_2(x_1,x_2) = x_1 \oplus x_2$.

While there are some practical benefits for keeping the data f and ϕ separate, we will consider a slight variation on path-sum expressions that we call *pure path-sum expressions*, which keeps all of the relevant data about U in ϕ and treats inputs and outputs on the same footing. This will enable us to make an exact correspondence with ZH-diagrams in the sequel.

For a set S, let S^* be the set of finite lists of elements of S.

Definition 3.2. A *pure path-sum expression* consists of:

- A set of *path variables* $\boldsymbol{x} = (x_1, \ldots, x_k)$,
- an *input signature* $\mathbf{i} \in \{1, \ldots k\}^*$,
- an *output signature* $\mathbf{o} \in \{1, \ldots k\}^*$,
- and a *phase polynomial* $\phi(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$.

The *associated linear operator* of a pure path-sum expression is $U := \sum_{\boldsymbol{x}} e^{2\pi i \phi(\boldsymbol{x})} |\boldsymbol{x}_\mathbf{o}\rangle \langle \boldsymbol{x}_\mathbf{i}|$ where $\boldsymbol{x}_\mathbf{o} = x_{o_1}x_{o_2}\cdots x_{o_{|o|}}$ and $\boldsymbol{x}_\mathbf{i} = x_{i_1}x_{i_2}\cdots x_{i_{|i|}}$.

We don't lose any expressiveness by using pure path-sum expressions. In fact, we can translate a path-sum expression in the form of (4) to a pure path-sum expression, at the cost of introducing some dummy variables:

$$U = \lambda \sum_{\boldsymbol{x},\boldsymbol{y}} e^{2\pi i \cdot \phi(\boldsymbol{x})} |f(\boldsymbol{x},\boldsymbol{y})\rangle \langle \boldsymbol{x}| = \frac{\lambda}{2^m} \sum_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{x},\boldsymbol{y}} e^{2\pi i \cdot \left[\phi(\boldsymbol{x}) + \frac{1}{2}\sum_j v_j (w_j + f_j(\boldsymbol{x},\boldsymbol{y}))\right]} |\boldsymbol{w}\rangle\langle \boldsymbol{x}| \tag{5}$$

Here we used the identity $\frac{1}{2}\sum_{v_j} e^{i\pi v_j (w_j + f_j(\boldsymbol{x},\boldsymbol{y}))} = \delta_{w_j, f_j(\boldsymbol{x},\boldsymbol{y})}$ to obtain the RHS above.

Remark 3.3. It will be convenient in the sequel for ϕ to have a certain canonical form. Namely that $\phi(\boldsymbol{x}) = \sum_i \lambda_i \phi_i(\boldsymbol{x})$ for $\lambda_i \in \mathbb{R}$ where each ϕ_i is a *Boolean monomial*, i.e. $\phi_i(\boldsymbol{x}) = \prod_j x_j \cdot y_j^i$ for some $\mathbf{y}^i \in \mathbb{B}^k$. The procedure above introduces terms in ϕ that are not of this form, but are $e^{2\pi i \alpha f(\boldsymbol{x})}$ where $f : \mathbb{B}^k \to \mathbb{B}$ is some Boolean function. Such functions f can however always be written as an XOR of monomials. Using the identity $x \oplus y = x + y - 2x \cdot y$ repeatedly we can then write any Boolean function f as $f = \sum_j \alpha_j f_j$ where $\lambda_j \in \mathbb{R}$ and f_j are monomials. Hence $e^{2\pi i \alpha f(\boldsymbol{x})} = e^{2\pi i \sum_j \alpha \lambda_j f_j}$ is of the desired form.

Example 3.4. Applying the transformation of Eq. (5) to the path-sum representation of Example 3.1 yields

$$\text{CNOT} \;=\; \frac{1}{4} \sum_{\substack{v_1,v_2,\\ w_1,w_2,\\ x_1,x_2}} e^{2\pi i \cdot \frac{1}{2}\left[v_1(w_1+x_1)+v_2(w_2+(x_1\oplus x_2))\right]} \left|w_1, w_2\right\rangle \left\langle x_1, x_2\right| .$$

Further applying $x_1 \oplus x_2 = x_1 + x_2 - 2x_1 \cdot x_2$, and eliminating monomials with coefficient 1, we can simplify this to

$$\text{CNOT} \;=\; \frac{1}{2} \sum_{\substack{v_1,v_2,\\ w_1,w_2,\\ x_1,x_2}} e^{2\pi i \cdot \left[\frac{1}{2}v_1 w_1 + \frac{1}{2}v_1 x_1 + \frac{1}{2}v_2 w_2 + \frac{1}{2}v_2 x_1 + \frac{1}{2}v_2 x_2\right]} \left|w_1, w_2\right\rangle \left\langle x_1, x_2\right| , \tag{6}$$

which is indeed a pure path-sum. It can actually be further simplified to the path-sum of Eq. 8 later on by using the identity $\sum_{v_1,w_1} (-1)^{v_1 w_1 + v_1 x_1} \left|w_1\right\rangle = 2 \left|x_1\right\rangle$.

A pure path-sum allows easy repetition of variables in the input as well as the output, so that we can succinctly write linear maps such as the following one representing a Z-spider with 2 inputs and 1 output:

$$M \;:=\; \sum_x e^{2\pi i \cdot 0} \left|x\right\rangle \left\langle xx\right|$$

While it is possible to write these maps using a standard path-sum, this requires additional dummy variables, e.g. $M \;::\; \left|x_0 x_1\right\rangle \mapsto \frac{1}{2} \sum_v e^{2\pi i \cdot \left[\frac{1}{2}(v(x_0+x_1))\right]} \left|x_0\right\rangle$.

In [2], several reduction rules were presented for path-sum expressions. Each of these 4 rules removes at least one path-variable from the expression. We present these rules, translated into pure path-sum expressions, in Figure 1. Of these rules, [Elim] is the easiest to understand: a single variable that does not occur in any other part of the expression can be safely removed. The other rules are however considerably more opaque in their interpretation. In the next section we will see that translated into the ZH-calculus, they gain an intuitive meaning.

4 Translating path-sums into ZH-diagrams

In this section we will see that pure path-sums can be straightforwardly represented by hypergraph-like ZH-diagrams and vice versa. We will use this correspondence to translate the path-sum reduction rules into rules for the ZH-calculus, and prove them diagrammatically. In the process we will see that these rules correspond to quite canonical hypergraph-theoretic operations.

First, let us describe the translation between ZH-diagrams and pure path-sum expressions.

Definition 4.1. A hypergraph-like ZH-diagram D is given by the following data:

1. Finite sets $[k]$ and $[l]$ of *spiders* and *H-boxes*, respectively,

$$\lambda \sum_{y_0,\boldsymbol{x}} e^{2\pi i R(\boldsymbol{x})} |\boldsymbol{x_o}\rangle \langle \boldsymbol{x_i}| \longrightarrow 2\lambda \sum_{\boldsymbol{x}} e^{2\pi i R(\boldsymbol{x})} |\boldsymbol{x_o}\rangle \langle \boldsymbol{x_i}| \qquad \text{[Elim]}$$

$$\lambda \sum_{y_0,\boldsymbol{x}} e^{2\pi i\left(\frac{1}{4}y_0 + \frac{1}{2}y_0 Q(\boldsymbol{x}) + R(\boldsymbol{x})\right)} |\boldsymbol{x_o}\rangle \langle \boldsymbol{x_i}| \longrightarrow \sqrt{2}\lambda \sum_{\boldsymbol{x}} e^{2\pi i\left(\frac{1}{8} - \frac{1}{4}\overline{Q}(\boldsymbol{x}) + R(\boldsymbol{x})\right)} |\boldsymbol{x_o}\rangle \langle \boldsymbol{x_i}| \qquad [\omega]$$

$$\lambda \sum_{y_0,y_1,\boldsymbol{x}} e^{2\pi i\left(\frac{1}{2}y_0(y_1 + Q(\boldsymbol{x})) + R(y_1,\boldsymbol{x})\right)} |\boldsymbol{x_o}\rangle \langle \boldsymbol{x_i}| \longrightarrow 2\lambda \sum_{\boldsymbol{x}} e^{2\pi i\left(R[y_1 \leftarrow \overline{Q}]\right)(\boldsymbol{x})} |\boldsymbol{x_o}\rangle \langle \boldsymbol{x_i}| \qquad \text{[HH]}$$

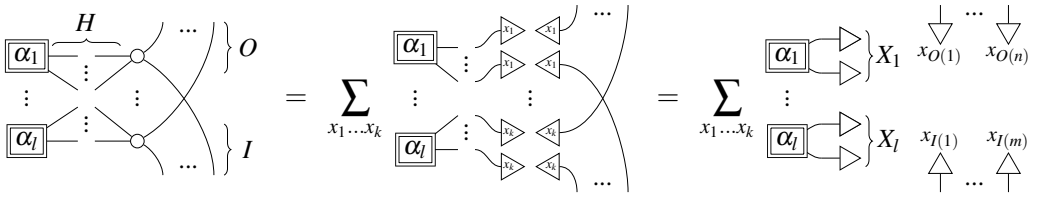
$$\lambda \sum_{y_0,y_1,\boldsymbol{x}} e^{2\pi i\left(\alpha y_0 X(\boldsymbol{x}) + \frac{1}{2}y_0 Q(\boldsymbol{x}) + \frac{1}{2}y_0 y_1 + \frac{1}{2}y_1 Q'(\boldsymbol{x}) + \beta y_1(1 - X(\boldsymbol{x})) + R(\boldsymbol{x})\right)} |\boldsymbol{x_o}\rangle \langle \boldsymbol{x_i}|$$
$$\longrightarrow 2\lambda \sum_{\boldsymbol{x}} e^{2\pi i\left(\frac{1}{2}Q(\boldsymbol{x})Q'(\boldsymbol{x}) + \alpha X(\boldsymbol{x})\overline{Q'}(\boldsymbol{x}) + \beta(1 - X(\boldsymbol{x}))\overline{Q}(\boldsymbol{x}) + R(\boldsymbol{x})\right)} |\boldsymbol{x_o}\rangle \langle \boldsymbol{x_i}| \qquad \text{[Case]}$$

Figure 1: Pure path-sum reduction rules. Here $\lambda \in \mathbb{C}$, $\alpha, \beta \in \mathbb{R}$, $Q, Q', X : \mathbb{B}^k \to \mathbb{B}$ are Boolean polynomials and $R : \mathbb{B}^k \to \mathbb{R}$ is any semi-Boolean function. The symbol $\overline{P}$ for a Boolean polynomial $P : \mathbb{B}^n \to \mathbb{B}$ represents its *lifting* $\overline{P} : \mathbb{B}^n \to \mathbb{R}$ that is defined inductively by $\overline{x} = x$, $\overline{PQ} = \overline{P} \cdot \overline{Q}$ and $\overline{P \oplus Q} = \overline{P} + \overline{Q} - 2\overline{PQ}$. Hence, it maps a Boolean polynomial to an integer polynomial which has the same value over the Booleans. Finally, $R[y_1 \leftarrow \overline{Q}]$ indicates that every occurrence of y_1 in the function R is replaced by the value of $\overline{Q}(\mathbf{x})$.

2. a function $H : [l] \to \mathscr{P}([k])$ where $H(i) = J$ when the i-th H-box is connected to all of the spiders in J,
3. a set of phase angles $\{\alpha_1, \ldots, \alpha_l\}$ where $\alpha_j \in [0, 2\pi)$ is the label on H-box j,
4. an input function $I : [m] \to [k]$ where $I(i) = j$ when the i-th input of D is connected to spider j and similarly $O(i) = j$ when the i-th output is connected to spider j, and
5. an output function $O : [n] \to [k]$ defined similarly.

Note that the first two items in Definition 4.1 specify an undirected hypergraph, whence the name.

Following [8] let us introduce notation for the computational basis states and effects: $\overset{x}{\triangledown} := |x\rangle$ and $\overset{x}{\triangle} := \langle x|$. Then starting with a generic hypergraph-like ZH-diagram, we can expand each of the Z-spiders as a sum over basis elements as follows:



where the sets $X_1, \ldots, X_l$ are defined as $X_j := \{x_i \,|\, i \in H(j)\}$. Note we can write basis elements 'sideways' without ambiguity as $(|x\rangle)^T = \langle x|$. In the RHS above, each H-box contributes a phase α_j when all of the variables in X_j are 1 (i.e. when $\prod X_j = 1$) and a phase of 0 otherwise. So the whole ZH-diagram evaluates to:

$$\sum_{x_1 \ldots x_k} e^{i\left[\sum_j \alpha_j \cdot \prod X_j\right]} \left|x_{O(1)} \ldots x_{O(n)}\right\rangle \left\langle x_{I(1)} \ldots x_{I(m)}\right| \tag{7}$$

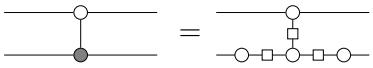
Letting $\phi(\boldsymbol{x}) := (\sum_j \alpha_j \cdot \prod X_j)/2\pi$, $\mathbf{i} := [I(1), ..., I(m)]$, and $\mathbf{o} := [O(1), ..., O(n)]$, we see that (7) is a generic pure path-sum expression.

Conversely, from any pure path-sum expression e, representing the phase polynomial $\phi(\boldsymbol{x})$ as a sum of monomials (cf. Remark 3.3), we can reconstruct H and $\{\alpha_j\}_j$ from the phase polynomial $\phi(\boldsymbol{x})$ and the

functions I and O from the lists $\mathbf{i}$ and $\mathbf{o}$, such that evaluating the ZH-diagram as in (7) gives e. If we ensure $\phi(\boldsymbol{x})$ has no repeated monomials, this reconstruction is furthermore unique, up to re-indexing of H-boxes.

Let $[.]_{\text{zh}\to\text{ps}}$ be the operation of translating a ZH-diagram to a pure path-sum expression by evaluation, and let $[.]_{\text{ps}\to\text{zh}}$ by the process of reconstructing a ZH-diagram from a pure path-sum expression. By construction these satisfy: $[[e]_{\text{ps}\to\text{zh}}]_{\text{zh}\to\text{ps}} = e$ and $[[D]_{\text{zh}\to\text{ps}}]_{\text{ps}\to\text{zh}} \cong D$, for any pure path-sum expressions e and hypergraph-like ZH-diagrams D, where '$\cong$' means equal up to permutation of the sets of spiders and H-boxes. In [24] it is shown that this construction is actually functorial and leads to an equivalence of categories.

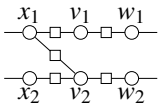
Example 4.2. The CNOT gate in the ZH-calculus is the following diagram.



Here the RHS is the hypergraph-like diagram resulting from applying Lemma 2.6 to the LHS. Translating it into a path-sum (and recalling that by convention unlabelled H-boxes have a label of -1) gives

$$\sum_{x_1,x_2,x_3,x_4\in\mathbb{B}} e^{2\pi i\frac{1}{2}(x_2x_3+x_1x_3+x_3x_4)}\,|x_1x_4\rangle\langle x_1x_2|\,. \tag{8}$$

Translating the CNOT path-sum of Eq. (6) into a ZH-diagram gives:



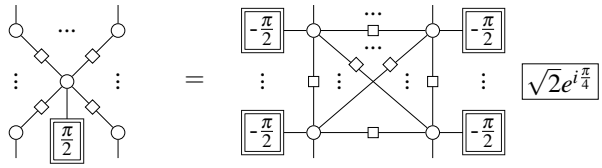
By applying $[.]_{\text{ps}\to\text{zh}}$ to both sides of the rules of Figure 1, we get an equation between (families of) ZH-diagrams. For [Elim] it is easy to see that the corresponding ZH-calculus rule is the simple removal of an arity-0 spider representing the scalar 2. For the other rules it is harder to see directly what the translation should be. We cover each of the rules [ω], [HH] and [Case] in the next subsections.

4.1 Hyper-local complementation

In this section, we will look at the [ω] rule from Figure 1 and show it is equivalent to a new simplification we can derive using the ZH-calculus, which we call *hyper-local complementation*.

Definition 4.3. Let $G = (V,E)$ be a simple graph and $u \in V$ a vertex. The *local complementation of G about the vertex u*, written as $G\star u$, is the graph (V,E') where $\{v,w\} \in E'$ iff $\{v,w\} \notin E$ when v and w are both neighbours of u in V, and $\{v,w\} \in E'$ iff $\{v,w\} \in E$ otherwise. In other words: $G\star u$ is the same graph as G except that neighbours of u are connected iff they are not connected in G.

Local complementation has featured in quantum information theory as it can be used to combinatorially capture local Clifford equivalence of certain stabiliser states called *graph states* [18]. It has also been used in the context of circuit simplification with the ZX-calculus [9]. There, it was shown that a Z-spider labelled by a phase of $\pm\frac{\pi}{2}$ can be deleted from a ZX-diagram without changing the linear map, as long as one first performs a local complementation about the vertex. Translating the rule from [9] in ZH notation, we obtain:

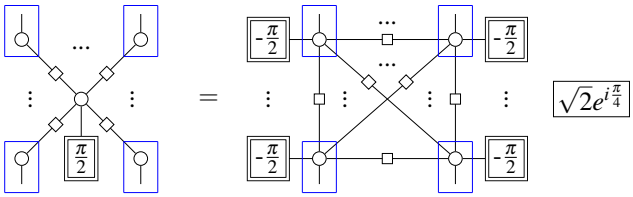


(9)

where the right-hand side is a totally connected graph of Z spiders, connected via H-boxes. This rule can be proven in the ZX-calculus, and hence by completeness, also in the ZH-calculus.

We can extend this to *hyperlocal complementation* by introducing a !-box on each of the Z-spiders at the boundary:

Proposition 4.4. The following hyperlocal complementation rule holds in the ZH-calculus.



(10)

Proof. See Appendix A. □

Remark 4.5. That a version of hyperlocal complementation can be implemented on a hypergraph state using local operations was first found in [13].

Remark 4.6. As in [4], we adopt a 'hybrid' notation mixing !-boxes with ellipses to express the hyperlocal complementation rule. This is due to a limitation of !-box notation, which is not rich enough to capture complete graphs of unbounded size [25].

Let us consider $[\omega]$ in Figure 1 in more detail to find the connection between it and Eq. (10). The phase-polynomial on the LHS of $[\omega]$ is $\phi(y_0,\mathbf{x}) = \frac{1}{4}y_0 + \frac{1}{2}y_0 Q(\boldsymbol{x}) + R(\boldsymbol{x})$ where Q is a Boolean polynomial and R is an irrelevant semi-Boolean function. The variable y_0 corresponds to the central spider of the LHS of Eq. (10) while the $\frac{1}{4}y_0$ term in the phase polynomial corresponds to the $\frac{\pi}{2}$-labelled H-box. We can write Q as $Q(\boldsymbol{x}) = \oplus_{j=1}^{n} m'_j(\boldsymbol{x})$ for some monomials m'_j. Note that we have the identity $e^{2\pi i \frac{1}{2} y_0 Q(\boldsymbol{x})} = e^{2\pi i \frac{1}{2}\sum_j y_0 m'_j(\boldsymbol{x})}$. Hence, the term $\frac{1}{2}y_0 Q(\mathbf{x})$ contributes an H-box to the ZH-diagram for each monomial m'_j, which are the neighbours of the central spider in Eq. (10).

The RHS phase polynomial in $[\omega]$ is $\phi(\mathbf{x}) = \frac{1}{8} - \frac{1}{4}\overline{Q}(\boldsymbol{x}) + R(\boldsymbol{x})$. The function R is again irrelevant, and the term $\frac{1}{8}$ becomes the $e^{i\pi/4}$ scalar in the RHS ZH-diagram. Note that the lifting of Q is given by

$$\overline{Q} = \sum_{r=1}^{n} \sum_{E \in \mathscr{P}_r([n])} (-2)^{r-1} \prod_{i \in E} m'_i \tag{11}$$

where $\mathscr{P}_r([n]) \subseteq \mathscr{P}([n])$ is the set of subsets of $[n]$ that contain exactly r elements. Hence,

$$e^{2\pi i(-\frac{1}{4}\overline{Q})} = e^{2\pi i(-\frac{1}{4}\sum_j m'_j + \frac{1}{2}\sum_{j<k} m'_j m'_k + f(\boldsymbol{x}))}$$

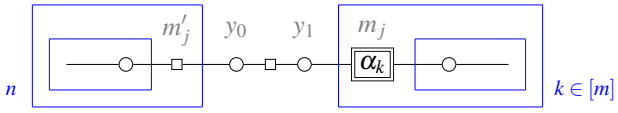
where $f(\boldsymbol{x})$ is the remainder, and is valued in the integers so that it does not contribute to the phase. Each of the $e^{-i\frac{\pi}{2}m'_j}$ terms contributes an $-\frac{\pi}{2}$-labelled H-box connected to the spiders of the monomial m'_i, while each $e^{i\pi m'_j m'_k}$ term contributes an unlabeled H-box connected to all the spiders of both m'_j and m'_k. This indeed results in the fully connected graph in the RHS of Eq. (10).

4.2 Fourier Hyper pivot

On the LHS of [HH] in Figure 1 the phase polynomial is $\phi(y_0,y_1,\boldsymbol{x}) = \frac{1}{2}y_0 y_1 + \frac{1}{2}y_0 Q(\boldsymbol{x}) + R(y_1,\boldsymbol{x})$. Hence, in the corresponding ZH-diagram we see that y_0 and y_1 are connected by an arity-2 exponentiated H-box with a phase of $2\pi\frac{1}{2} = \pi$, and hence is a regular Hadamard gate. Writing the Boolean polynomial Q as

$Q = \bigoplus_j^n m'_j$ where the m'_j are monomials as in the previous section we see that each monomial introduces an H-box to the ZH-diagram that is connected to the spider of y_0.

We can separate the action of y_1 in R as $R(\boldsymbol{x}, y_1) = S(\boldsymbol{x})y_1 + T(\boldsymbol{x})$ for some functions S and T where we can furthermore expand S as $S(\mathbf{x}) = \sum_j \frac{\alpha_j}{2\pi} m_j(\mathbf{x})$ for some monomials m_j. Hence, in the translated ZH-diagram y_1 shares an α_j-valued H-box with each of the spiders of the monomials m_j. Combining these observations we see that the relevant part of the pure path-sum on the LHS of [HH] is:



To write the RHS of [HH] we need to represent $R[y_1 \leftarrow \overline{Q}] = S(\boldsymbol{x}) \cdot \overline{Q}(\boldsymbol{x}) + T(\boldsymbol{x})$. By representing elements of the powerset $\mathscr{P}([n])$ as bitstrings in $\mathbb{B}^n$ we can rewrite the lifting of Q in Eq. (11) as

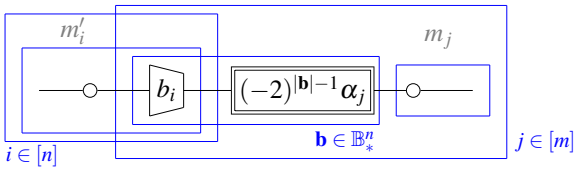
$$\overline{Q} = \sum_{\mathbf{b} \in \mathbb{B}^n_*} (-2)^{|\mathbf{b}|-1} \prod_{i \in [n]} (m'_i)^{b_i}.$$

Here $\prod_{i \in [n]} (m'_i)^{b_i}$ is again a Boolean monomial that is 1 precisely when m'_i is 1 for all i for which $b_i = 1$.

Hence, the releveant term in the RHS phase polynomial becomes

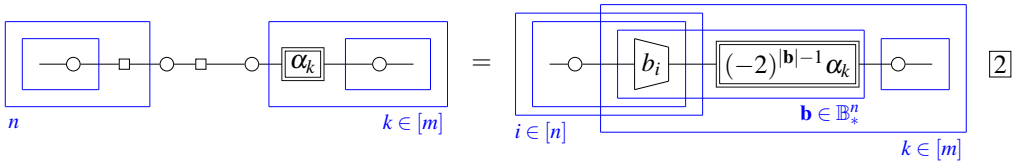
$$S \cdot \overline{Q} = \sum_j \sum_{\mathbf{b} \in \mathbb{B}^n_*} (-2)^{|\mathbf{b}|-1} \frac{\alpha_j}{2\pi} m_j \prod_{i \in [n]} m'^{b_i}_i$$

In the translation to the ZH-calculus we hence get H-boxes with a phase of $(-2)^{|\mathbf{b}|-1}\alpha_j$ that are connected to all the spiders of m_j and to all the spiders of the monomial $\prod_{i \in [n]} (m'_i)^{b_i}$. We can represent this using the disconnect box of Eq. (2), so that the corresponding ZH-diagram is:



Hence, [HH] is equivalent to the following result for ZH-diagrams, that we call the *Fourier hyper-pivot*.

Theorem 4.7 (Fourier hyper-pivot)**.** The following equation holds in the ZX-calculus for any set of real numbers $\alpha_1, \ldots, \alpha_m$.



(12)

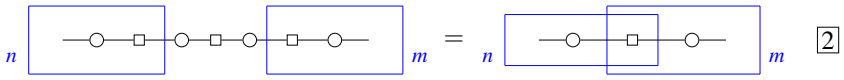
To see why we call this a Fourier hyper-pivot, let us introduce the notion of a regular pivot.

Definition 4.8. Let $G = (V, E)$ be a simple graph and $\{u, v\} \in E$ a connected pair of vertices. The *pivot of G along the edge uv* is the graph $G \star u \star v \star u$.

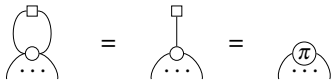
Whereas local complementation complements the connectivity of the neighbours of a vertex, a pivot along uv complements the connectivity between three groups of vertices: those connected to u and not to v, those connected to v but not to u, and those connected to both u and v.

In the ZX-calculus, pivoting is implemented by the bialgebra rule between the Z- and X-spider. In fact, an application of the bialgebra rule to two connected spiders u and v can be seen as a pivot along uv followed by a deletion of the vertices u and v. This can be represented particularly elegantly using !-boxes:

Proposition 4.9. The following pivoting rule holds in the ZH-calculus for any $n, m \in \mathbb{N}$.

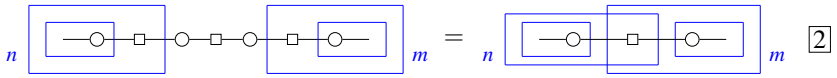


This is indeed a pivot (followed by vertex deletions) as every vertex (spider) connected to the left pivoted vertex becomes connected via a 2-ary H-box to every neighbour of the right pivoted vertex. The reason we only distinguish two groups of vertices, instead of three, is because spiders that are connected to both the vertices also belong to both !-boxes, and hence also get the appropriate connectivity. Since these vertices belong to both !-boxes, they furthermore get connected to *themselves*, a connection that can simplified to a simple phase:



By allowing each H-box in Proposition 4.9 to have arbitrary arity, we can generalise the above rule to a *hyper-pivot* followed by two vertex deletions.

Proposition 4.10. The following hyper-pivot rule holds in the ZH-calculus for any $n, m \in \mathbb{N}$.



(RHP)

We see that Proposition 4.10 is a special case of Theorem 4.7 where all the α_j are equal to π (which corresponds to R being a Boolean polynomial in [HH]). To prove Theorem 4.7 we combine the hyper-pivot rule with the Fourier transform of Proposition 2.1, hence the name. The proofs of Theorem 4.7 and Proposition 4.10 are given in Appendix A.

Remark 4.11. If we assume (RHP) as a rule of the ZH-calculus together with (ZS1), (ZS2) and (HS2) of Figure 2, then we can prove the rules (BA1), (BA2), and (HS1). Hence, hyperpivoting supersedes these separate rules, resulting in a more 'compact' calculus.

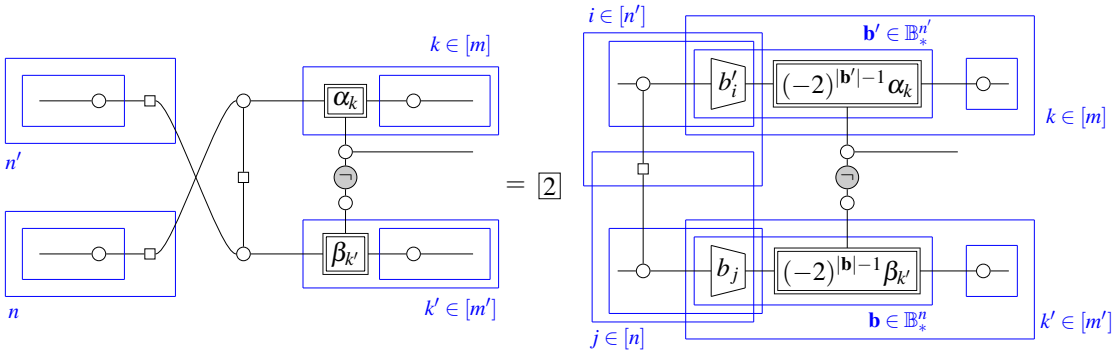
Remark 4.12. Similar to how a pivot is implemented by a combination of three local complementations, our hyperpivot rule can be implemented by three of the hyperlocal complementations of [13].

Using the hyperpivot rule we can straightforwardly prove identities that would be hard to show using the ZX-calculus. We give an example of this in Appendix B.

4.3 Case hyper pivot

Like the [HH] rule, the [Case] rule enables the elimination of a pair of variables y_0, y_1 from a path-sum expression. However, unlike the case rule, *both* variables are allowed to occur in monomials that have coefficients other than $\frac{1}{2}$, as long as they are 'orthogonal' in a certain sense. That is, the non-$\frac{1}{2}$ monomials containing y_0 must be multiplied by some boolean function $X(\mathbf{x})$, whereas those containing y_1 must be multiplied by its negation. This can be expressed as the following ZH-calculus rule:

Theorem 4.13 (Case hyper-pivot)**.** The following rewrite rule holds in the ZH-calculus.



We present the details of its proof and its exact relation to the [Case] rule in Appendix C.

This rule seems less canonical than the others, and it is in fact omitted in a later presentation of the path-sum formalism by Amy [1]. However, it is interesting to note that this rule essentially arises from two different, incompatible simplifications using hyper-pivoting starting from the same ZH-diagram (diagram (15) in Appendix C). To adopt terminology from rewrite theory, the Case hyper-pivot rule arises from closing a *critical pair* of the hyper-pivot rule with itself. This shows firstly that simplification using the other two laws is not confluent, which is unsurprising given its heuristic nature. More interestingly, it suggests that standard automated techniques for dealing with critical pairs, namely *Knuth-Bendix completion*, could yield useful new simplification rules.

5 Conclusion and Future Work

We have found a bijective correspondence between path-sum expressions and ZH-diagrams and we gave ZH-calculus versions of each of the path-sum simplification rules. Furthermore, the derivation of the Case hyper-pivot rule suggests many more such rules could be recovered automatically by studying overlapping applications of the existing simplification rules.

The natural next step is to cash in these new structural insights to develop new techniques to simplify and verify circuits. ZH-diagrams and the Fourier hyper-pivot have been implemented using the PyZX library [15] and seem to be effective at reducing many families of circuits to a compact form analogous to the GSLC form of ZX-diagrams [3], however it is a topic of ongoing research to characterise exactly when this succeeds and when it succeeds efficiently.

Amy showed that the path-sum reduction rules suffice to verify the functional interpretation of many quantum circuits. By casting his rules in the ZH-calculus we extend this to arbitrarily constructed linear maps based on hypergraph states. In particular, we can use our results to verify and analyse MBQC schemes based on hypergraph states, such as those of Refs. [22, 12]. Using graphical languages to study MBQC schemes has already resulted in several new results [10, 5] and so this seems like a promising approach to new results in the burgeoning field of hypergraph MBQC.

Finally, it does not seem like the simplification rules here have yet captured the full power of the ZH-calculus. It would be interesting to see if other ZH-calculus rules, such as the ortho rule from Ref. [4], can be translated into useful, and previously unconsidered simplification rules for path-sum expressions and/or ZH-diagrams.

Acknowledgements: We wish to thank Matthew Amy and Neil J. Ross for valuable discussions regarding path-sums, and Matt for help in understanding his software Feynman. AK and JvdW gratefully acknowledge support of AFOSR grant FA2386-18-1-4028.

References

[1] Matthew Amy (2019): *Formal Methods in Quantum Circuit Design*. Ph.D. thesis, University of Waterloo.

[2] Matthew Amy (2019): *Towards Large-scale Functional Verification of Universal Quantum Circuits*. In Peter Selinger & Giulio Chiribella, editors: Proceedings of the 15th International Conference on *Quantum Physics and Logic*, Halifax, Canada, 3-7th June 2018, *Electronic Proceedings in Theoretical Computer Science* 287, Open Publishing Association, pp. 1–21, doi:10.4204/EPTCS.287.1.

[3] Miriam Backens (2014): *The ZX-calculus is complete for stabilizer quantum mechanics*. *New Journal of Physics* 16(9), p. 093021, doi:10.1088/1367-2630/16/9/093021.

[4] Miriam Backens & Aleks Kissinger (2018): *ZH: A Complete Graphical Calculus for Quantum Computations Involving Classical Non-linearity*. doi:10.4204/EPTCS.287.2.

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

[6] Adam D Bookatz (2012): *QMA-complete problems*. *arXiv preprint arXiv:1212.6312*.

[7] Bob Coecke & Ross Duncan (2009): *Interacting Quantum Observables: Categorical Algebra and Diagrammatics*. *CoRR*, doi:10.1088/1367-2630/13/4/043016.

[8] Bob Coecke & Aleks Kissinger (2018): *Picturing Quantum Processes - A First Course on Quantum Theory and Diagrammatic Reasoning*. doi:10.1007/978-3-319-91376-6_6.

[9] Ross Duncan, Aleks Kissinger, Simon Perdrix & John van de Wetering (2019): *Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus*.

[10] Ross Duncan & Simon Perdrix (2010): *Rewriting measurement-based quantum computations with generalised flow*. In: *International Colloquium on Automata, Languages, and Programming*, Springer, pp. 285–296, doi:10.1007/978-3-642-14162-1_24.

[11] Mariami Gachechiladze (2019): *Quantum hypergraph states and the theory of multiparticle entanglement*. Ph.D. thesis, Universität Siegen. `http://141.99.19.133//bitstream/ubsi/1509/2/Dissertation_Mariami_Gachechiladze.pdf`.

[12] Mariami Gachechiladze, Otfried Gühne & Akimasa Miyake (2019): *Changing the circuit-depth complexity of measurement-based quantum computation with hypergraph states*. *Physical Review A* 99(5), p. 052304, doi:10.1103/PhysRevA.99.052304.

[13] Mariami Gachechiladze, Nikoloz Tsimakuridze & Otfried Gühne (2017): *Graphical description of unitary transformations on hypergraph states*. *Journal of Physics A: Mathematical and Theoretical* 50(19), p. 19LT01, doi:10.1088/1751-8121/aa676a.

[14] Aleks Kissinger, Alex Merry & Matvey Soloviev (2012): *Pattern graph rewrite systems*. doi:10.4204/EPTCS.143.5.

[15] Aleks Kissinger & John van de Wetering (2019): *PyZX: Large Scale Automated Diagrammatic Reasoning*. *arXiv:1904.04735*.

[16] Stach Kuijpers, John van de Wetering & Aleks Kissinger (2019): *Graphical Fourier Theory and the Cost of Quantum Addition*.

[17] Louis Lemonnier (2019): *Relating high-level frameworks for quantum circuits*. Master's thesis, Radboud University Nijmegen. Available at `https://www.cs.ox.ac.uk/people/aleks.kissinger/papers/lemonnier-high-level.pdf`.

[18] M. Van den Nest, J. Dehaene & B. De Moor (2004): *Graphical description of the action of local Clifford transformations on graph states*. *Physical Review A* 69(2), p. 9422, doi:10.1103/physreva.69.022316.

[19] Ri Qu, Juan Wang, Zong-shang Li & Yan-ru Bao (2013): *Encoding hypergraphs into quantum states*. *Physical Review A* 87(2), p. 022311, doi:10.1103/PhysRevA.87.022311.

[20] R. Raussendorf, D.E. Browne & H.J. Briegel (2003): *Measurement-based quantum computation on cluster states*. *Physical Review A* 68(2), p. 22312, doi:10.1103/physreva.68.022312.

[21] Matteo Rossi, Marcus Huber, Dagmar Bruß & Chiara Macchiavello (2013): *Quantum hypergraph states*. *New Journal of Physics* 15(11), p. 113022, doi:10.1088/1367-2630/15/11/113022.

[22] Yuki Takeuchi, Tomoyuki Morimae & Masahito Hayashi (2019): *Quantum computational universality of hypergraph states with Pauli-X and Z basis measurements*. *Scientific reports* 9(1), pp. 1–14, doi:10.1038/s41598-019-49968-3.

[23] Nikoloz Tsimakuridze & Otfried Gühne (2017): *Graph states and local unitary transformations beyond local Clifford operations*. *Journal of Physics A: Mathematical and Theoretical* 50(19), p. 195302.

[24] Renaud Vilmart (2020): *The Structure of Sum-Over-Paths, its Consequences, and Completeness for Clifford*. *arXiv preprint arXiv:2003.05678*.

[25] Vladimir Zamdzhiev (2016): *Rewriting Context-free Families of String Diagrams*. Ph.D. thesis, University of Oxford.

A Proofs of graphical rewrite rules

Let us first state the rules of the ZH-calculus. See Figure 2.

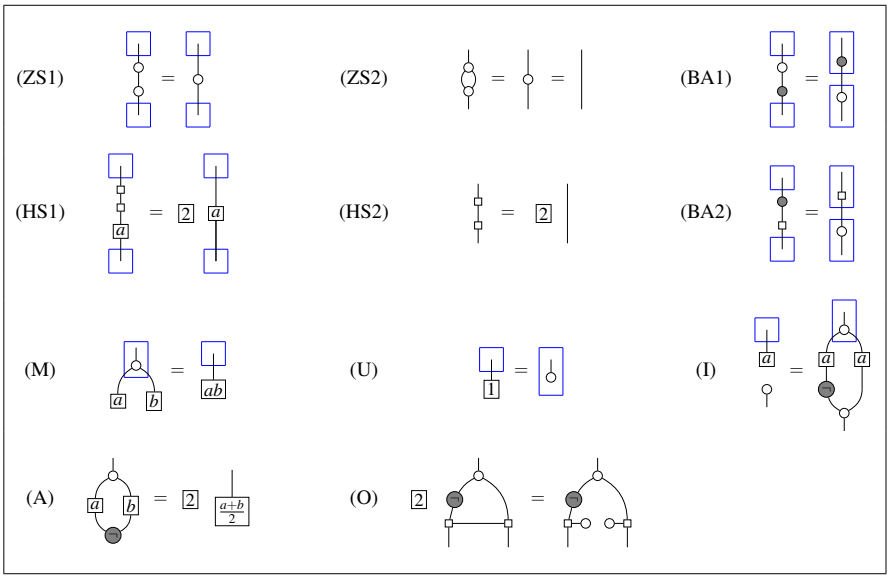
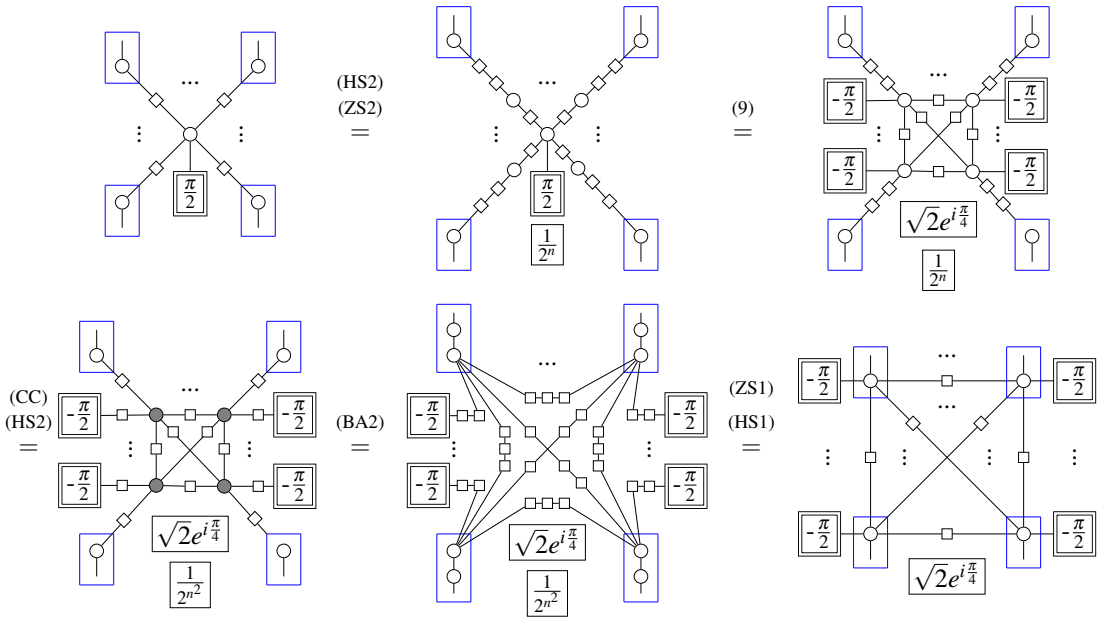


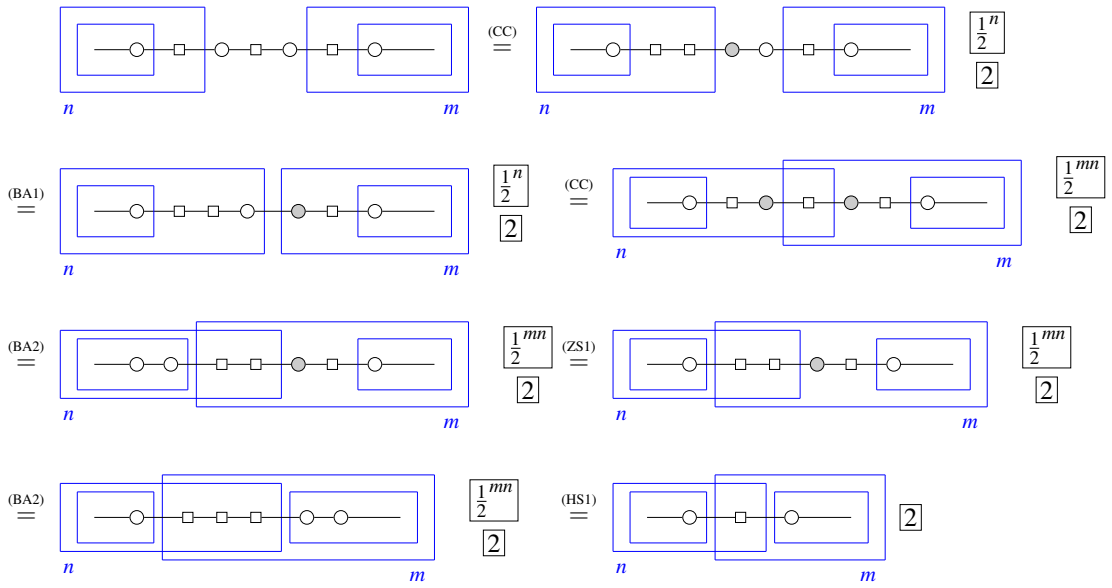
Figure 2: The rules of the ZH-calculus. Throughout, a, b are arbitrary complex numbers. These are the !-boxed versions of the rules as presented in [4].

Proof of Proposition 4.4.



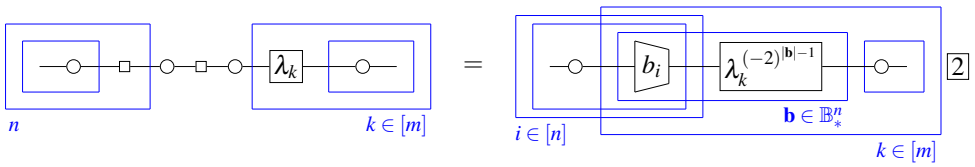
□

Proof of Proposition 4.10.

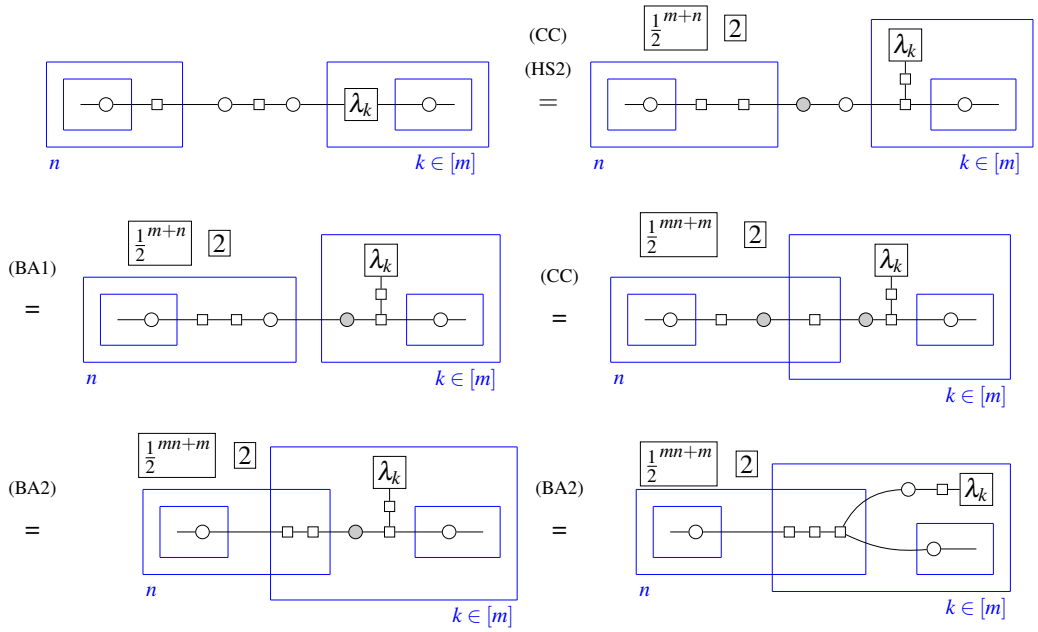


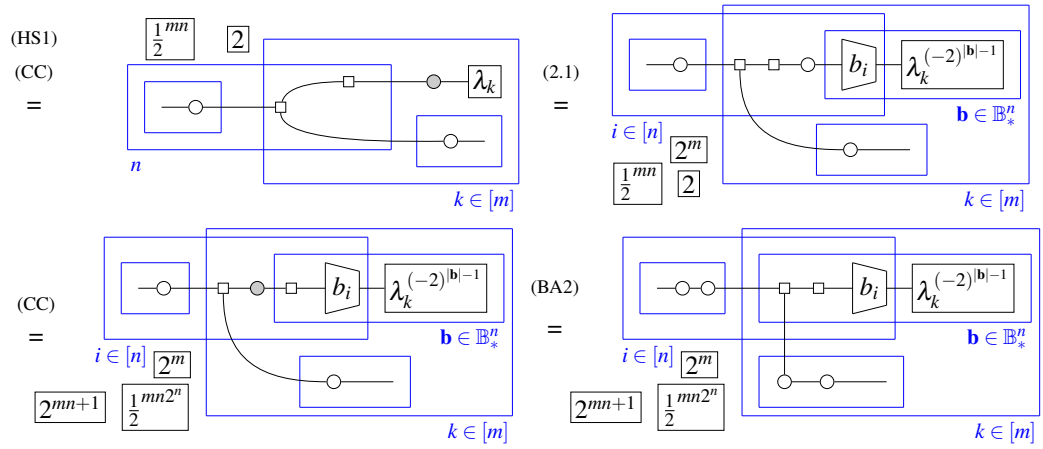
□

Proof of Theorem 4.7. Instead of proving Theorem 4.7 exactly, we will prove the slightly more general following equation, that has regular instead of exponentiated H-boxes:

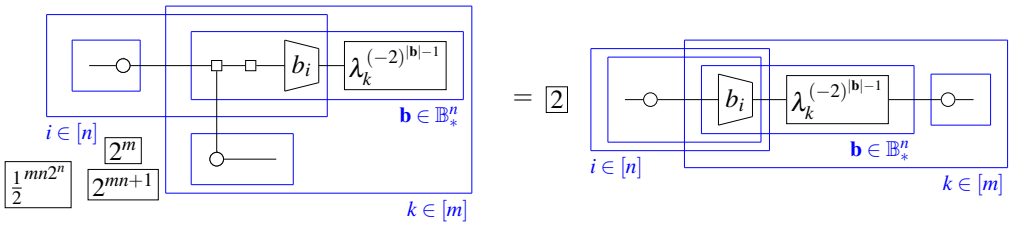


Let us prove this equation:



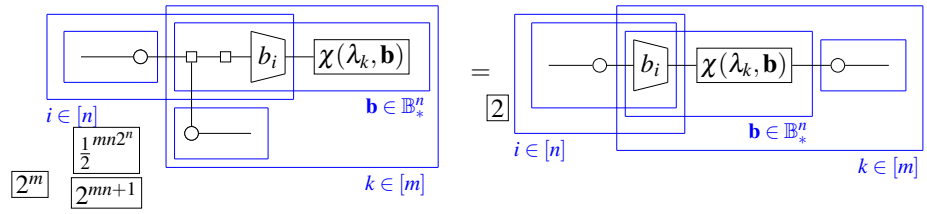


Fusing the connected white spiders, we need one final step, the correctness of which we prove in Lemma A.1:

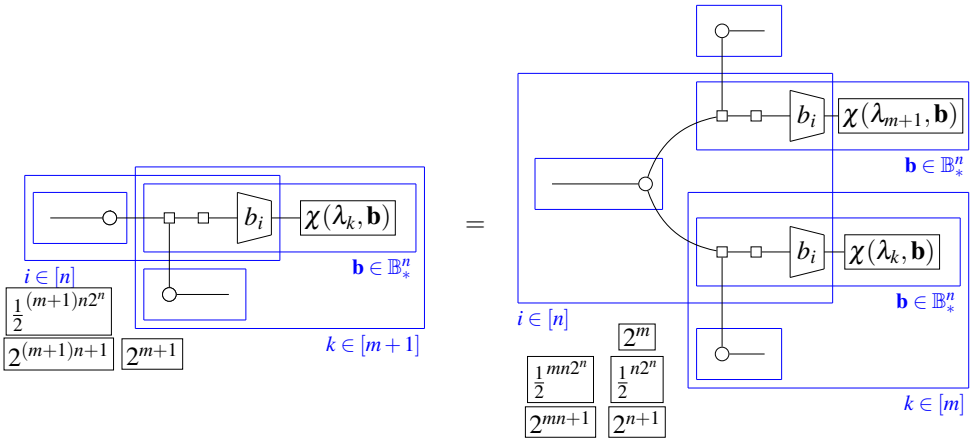


□

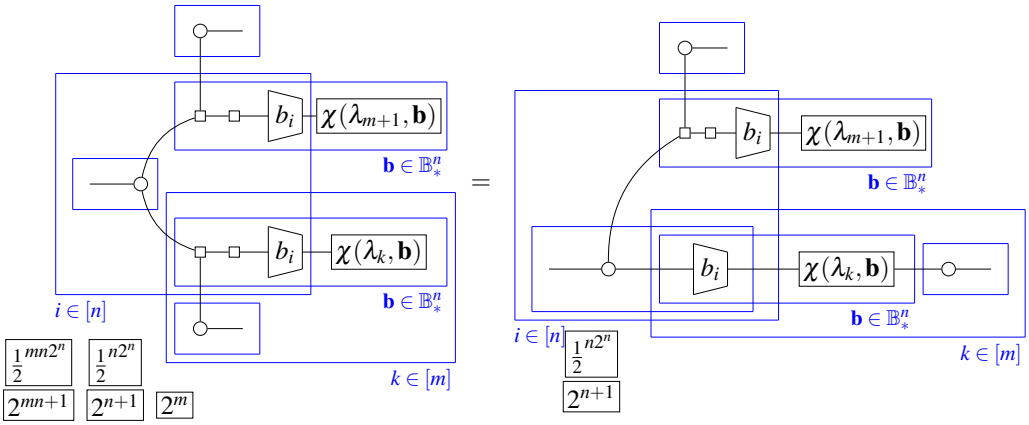
Lemma A.1. Let $\chi(\lambda, \mathbf{b}) := \lambda^{(-2)^{|\mathbf{b}|-1}}$. The following equation holds in the ZH-calculus for all n and m:



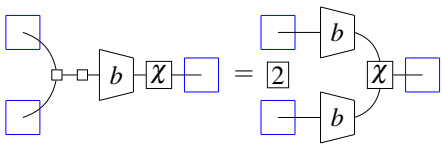
Proof. We prove by induction on m. The base case $m = 0$ is trivial, so suppose the equation holds for a fixed m. We will prove it for $m+1$.
First, expand the !-box of $m+1$ one time:



Then by induction hypothesis:

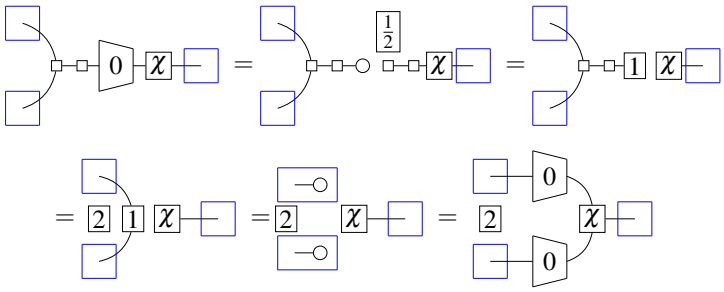


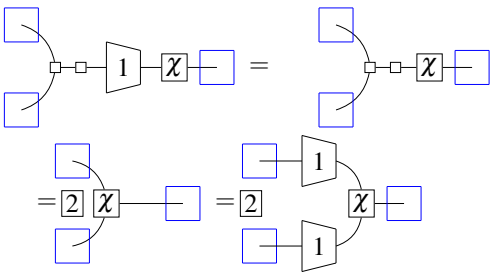
Before stating the next step, we prove that for any Boolean b:



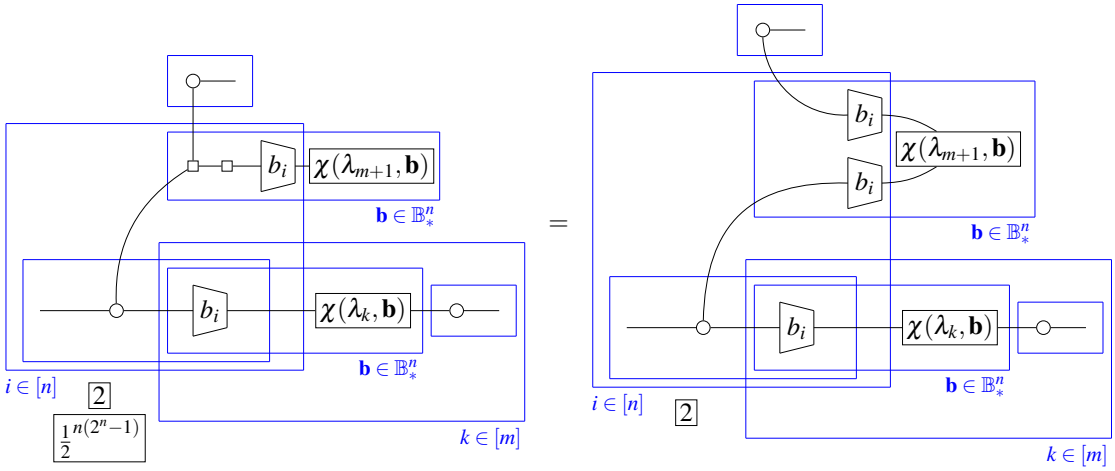
(13)

Indeed for both $b = 0$ and $b = 1$, the equation holds:

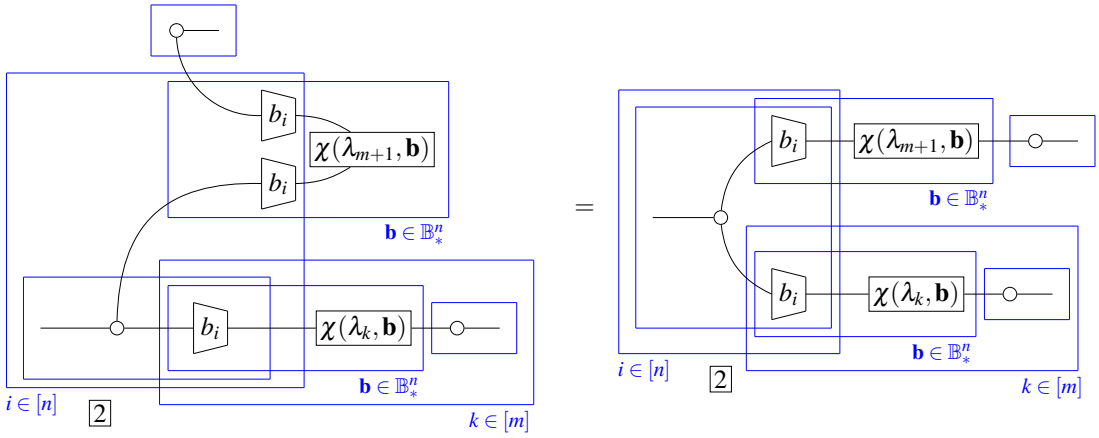




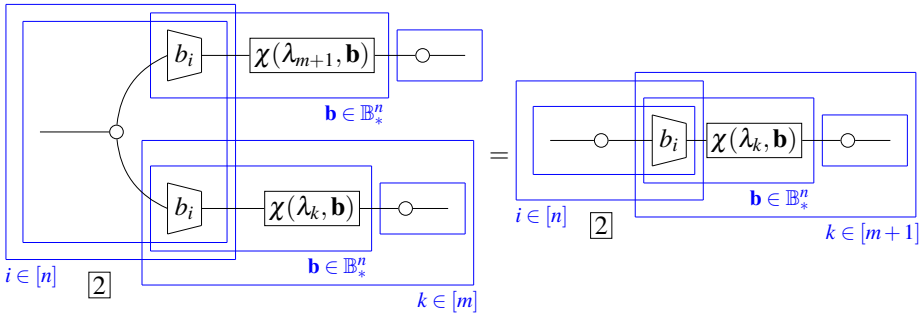
Using Eq. (13):



Now note that each of the white spiders in the top !-box is connected by many wires to the H-box labelled with χ. Hence, using Lemma 2.5 we can ignore the top b_i disconnect box:



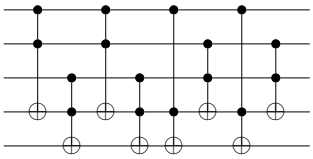
And finally we can put back the $m+1$ term of the !-box:



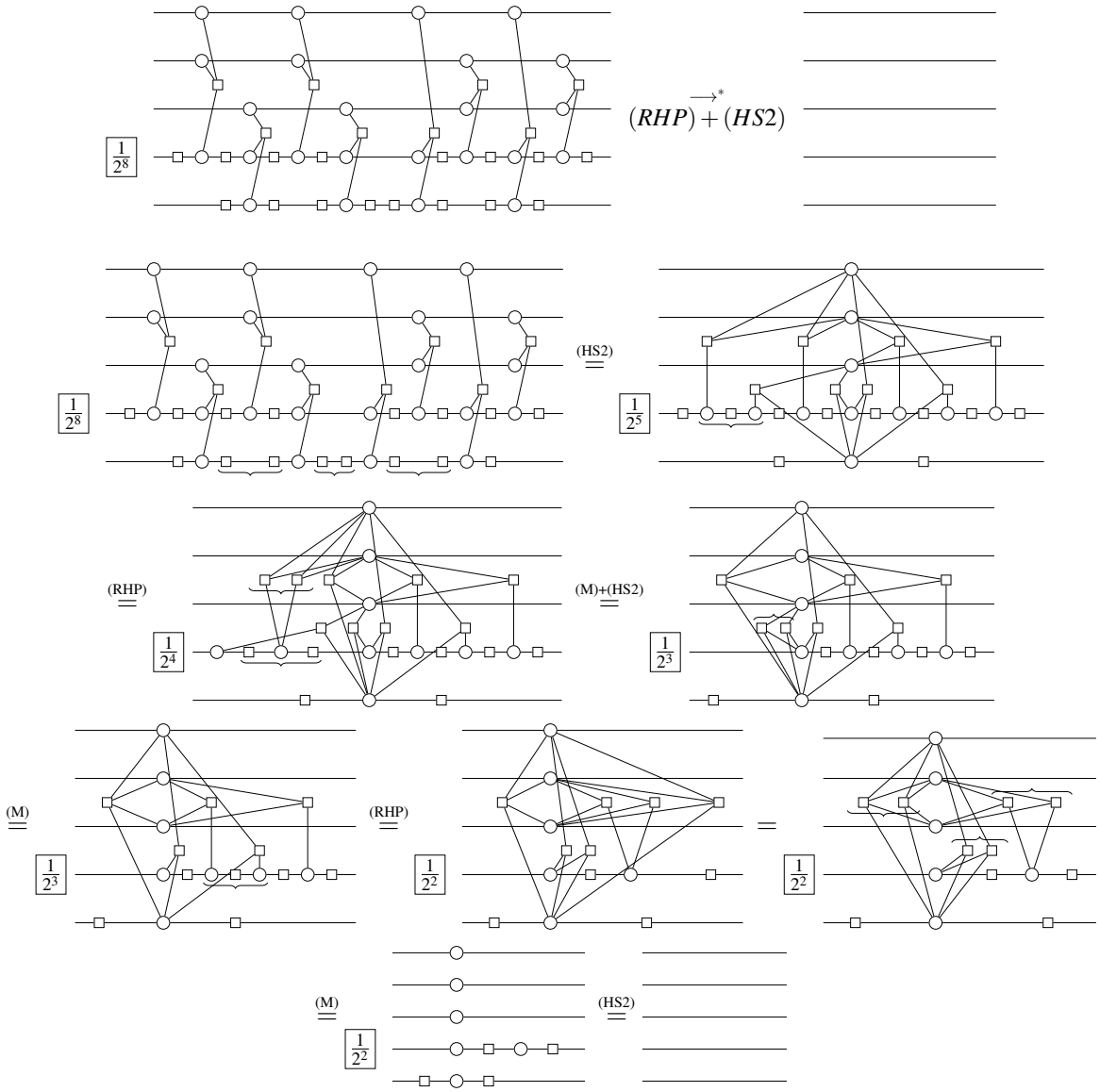
□

B Usage example of regular hyper pivot

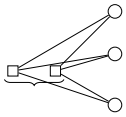
Using the hyperpivot rule we can straightforwardly prove identities that would be hard to show using the ZX-calculus. For instance, consider the following Toffoli circuit.

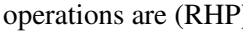


By reducing it to a hypergraph-like ZH-diagram, and repeatedly applying hyperpivoting to any pair of connected internal spiders, we can reduce it to the identity:



The several uses of (HS2) are highlighted by braces as follows:  . In this case, the multiplication law (M) is only used with $-1\times(-1)=1$, then two H-boxes connected to the same white dots eliminates themselves. They are signaled with braces:



Finally, the other operations are (RHP) applications, and highlighted this way:  .

C The [Case] rule in the ZH-calculus

The [Case] rule as stated in Figure 1 can be generalized to the following equation:

$$\sum_{y_0,y_1,\boldsymbol{x}} \varphi^X (-1)^{y_0 Q}(-1)^{y_0 y_1}(-1)^{y_1 Q'} \psi^{1-X} \left|\boldsymbol{x_o}\right\rangle\left\langle \boldsymbol{x_i}\right| \longrightarrow 2\sum_{\boldsymbol{x}} (-1)^{QQ'} \varphi[y_0 \leftarrow \overline{Q'}]^X \psi[y_1 \leftarrow \overline{Q}]^{1-X} \left|\boldsymbol{x_o}\right\rangle\left\langle \boldsymbol{x_i}\right|$$

where φ and ψ are complex functions overy_0 and $\boldsymbol{x}$, respectively y_1 and $\boldsymbol{x}$, and X, Q and Q' are boolean polynomials over $\boldsymbol{x}$.

This rule indeed generalises [Case] in Figure 1, which is easily seen by replacing φ by $e^{2\pi i(\alpha y_0 X+R)}$ and ψ by $e^{2\pi i(\beta y_1(1-X)+R)}$.

Let us first show the correctness of this rule algebraically. For a fixed $\mathbf{x}$, we have:

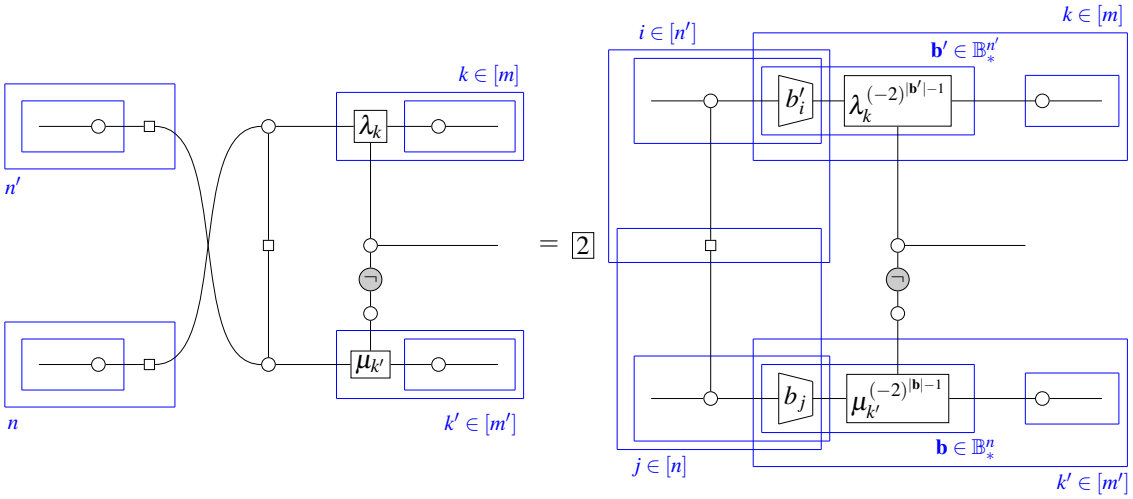
$$\sum_{y_0,y_1} \varphi^X (-1)^{y_0 Q}(-1)^{y_0 y_1}(-1)^{y_1 Q'} \psi^{1-X} = \begin{cases} \sum\limits_{y_0,y_1} (-1)^{y_0 Q}(-1)^{y_0 y_1}(-1)^{y_1 Q'} \psi \text{ if } X=0 \\ \sum\limits_{y_0,y_1} \varphi(-1)^{y_0 Q}(-1)^{y_0 y_1}(-1)^{y_1 Q'} \text{ if } X=1 \end{cases}$$

Now applying the [HH] rule to both cases, this reduces to:

$$\begin{cases} 2(-1)^{QQ'}\psi[y_1 \leftarrow \overline{Q}] \text{ if } X=0 \\ 2\varphi[y_0 \leftarrow \overline{Q'}](-1)^{QQ'} \text{ if } X=1 \end{cases}$$

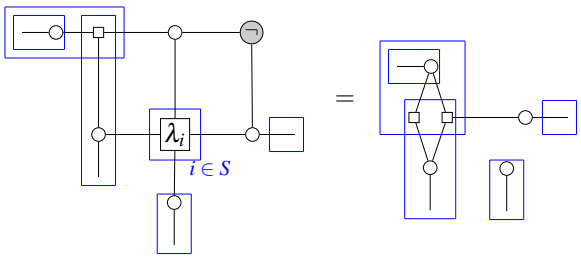
which is indeed equal to $2(-1)^{QQ'}\varphi[y_0 \leftarrow \overline{Q'}]^X \psi[y_1 \leftarrow \overline{Q}]^{1-X}$ as required.

Using similar reasoning as in Sections 4.1 and 4.2 we can show that this path-sum rule is equal to the following diagrammatic rule:



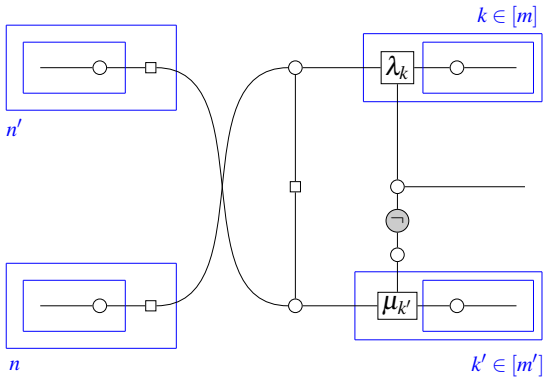
Before we prove this rule in the ZH-calculus, let us state the following lemma that we will need.

Lemma C.1. The ZH-calculus proves the following, for any set of complex numbers $\lambda_i \in S$:



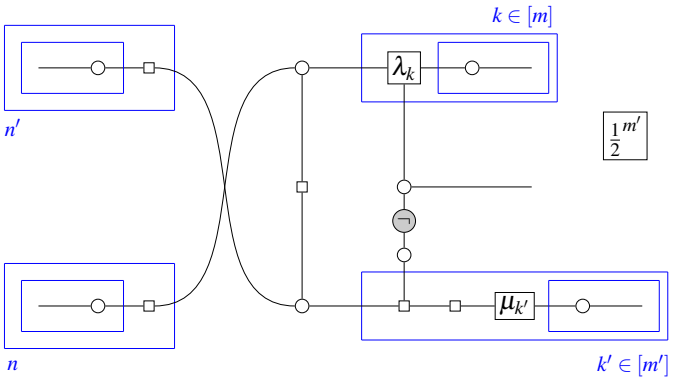
Proof. Expand the NOT using its definition, and apply the hyperpivot rule to the resulting white spider and its neighbour on the left. It is then straightforward to check that all the resulting H-boxes with multiples of λ_i cancel out, resulting in the correct diagram. □

Now let us prove the diagrammatic [Case] rule. Starting with:



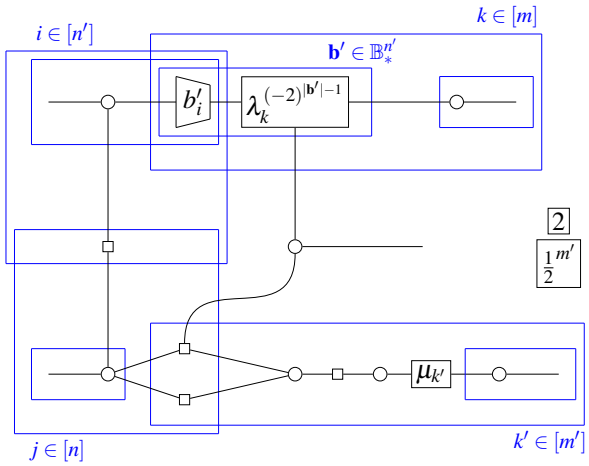
(14)

Unfuse all the μ-labelled H-boxes using (HS1):



(15)

Apply a Fourier hyper-pivot to the middle two spiders. This results in many copies of the λ_k H-boxes, but most of these are canceled by applying Lemma C.1, and we get the following diagram:



Now for every instance of the pair of spiders connected by the arity-2 H-box in the $[m']$-annotated !-box we will apply a Fourier hyper-pivot. This results in diagram (17) below. In order to see why we get this diagram, let us consider one of these hyper-pivots, and calculate the phases that appear on each of the H-boxes. Expanding the $[n]$-annotated !-box we get n groups of white spiders, n '2-legged' H-boxes and n '3-legged' H-boxes. Because each pair of a '2-legged' and '3-legged' H-box coming from the same expansion of the !-box are connected to the same group of white spiders, the Fourier transform of the hyper-pivot will introduce H-boxes that have *multiple* wires to the same group of white spiders. These are then collapsed to a single wire using Lemma 2.5. As a result, multiple H-boxes will be connected to the same set of spiders and hence will fuse using (M).

To understand the resulting phases on the H-boxes, we separate cases into H-boxes that arise just from the '2-legged' H-boxes, and ones that arise from a combination of both. The first case results in the normal pattern for a Fourier hyper-pivot and gives the lower μ-labeled H-boxes in diagram (17).

For the other case, let us denote by $f(p,q)$ the number of H-boxes after the Fourier hyper-pivot that have at least one of its connections arising from a 3-legged H-box, carry a phase of $\mu^{(-2)^{q-1}}$, and are connected to some chosen set of p groups of white spiders (and the group of spiders to the right of the μ H-box, which all H-boxes will be connected to). Then when (M) is applied to fuse all the H-boxes connected to the same set of p groups of spiders, the resulting H-boxes will have phases of

$$\prod_{q=p}^{2n} \mu^{f(p,q)(-2)^{q-1}} = \mu^{\sum_{q=p}^{2n} f(p,q)(-2)^{q-1}}. \tag{16}$$

Hence, to determine this phase, we need to calculate $f(p,q)$. So fix some set of size p of groups of spiders, and suppose there is an H-box with a phase of $(-2)^{q-1}$ connected to these groups. By assumption, some of the connections 'originated' from the 3-legged H-boxes. Let $1 \leq r \leq p$ denote this number. Note that there are $\binom{p}{r}$ ways to choose the 3-legged H-boxes. The remaining original connections of the H-box must have come from the 2-legged H-boxes. Some of these will connect to H-boxes that aren't connected to those of the first r, but others will be 'doubled' up. As the phase on the H-box is $(-2)^{q-1}$ there must have been a total of q wires, and hence $p-q$ wires need to be doubled up. There are r possible choices for doubling up, and hence there are $\binom{r}{p-q}$ possible ways we can double up the correct number of wires. Hence:

$$f(p,q) = \sum_{r=1}^{p} \binom{p}{r} \binom{r}{q-p}$$

Combining this with Eq. (16) we see that the resulting H-box connected to any set of p groups of spiders carries a phase of μ raised to the power of

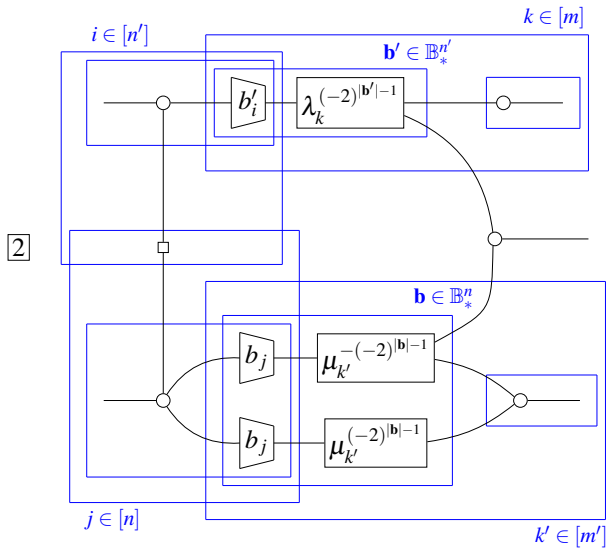
$$\sum_{q=p}^{2n}\sum_{r=1}^{p}\binom{p}{r}\binom{r}{q-p}(-2)^{q-1}.$$

Let us simplify this expression:

$$\begin{aligned}
\sum_{q=p}^{2n}\sum_{r=1}^{p}\binom{p}{r}\binom{r}{q-p}(-2)^{q-1} &\overset{*}{=} \sum_{r=1}^{p}\binom{p}{r}\sum_{q=p}^{p+r}\binom{r}{q-p}(-2)^{q-1}\\
&= \sum_{r=1}^{p}\binom{p}{r}\sum_{q=0}^{r}\binom{r}{q}(-2)^{q+p-1}\\
&= (-2)^{p-1}\sum_{r=1}^{p}\binom{p}{r}\sum_{q=0}^{r}\binom{r}{q}(-2)^{q}\\
&\overset{**}{=} (-2)^{p-1}\sum_{r=1}^{p}\binom{p}{r}(-1)^{r}\\
&= (-2)^{p-1}\left(\sum_{r=0}^{p}\binom{p}{r}(-1)^{r}-1\right)\\
&\overset{**}{=} (-2)^{p-1}(0-1)\\
&= -(-2)^{p-1}
\end{aligned}$$

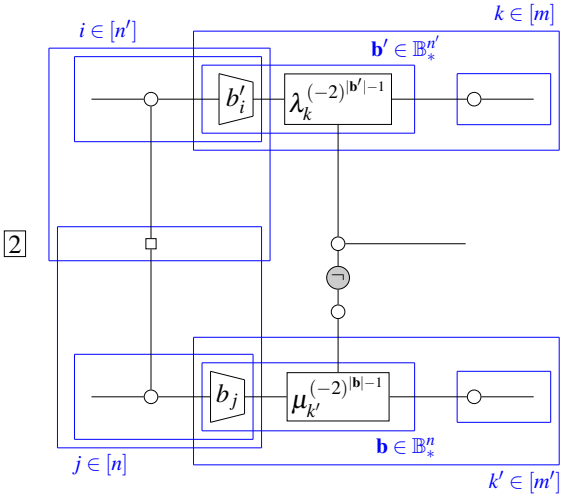
Here in the equality marked $*$ we are warranted in changing the limit of summation because $\binom{r}{q-p}=0$ whenever $q>p+r$, and the equalities marked $**$ are applications of the binomial identity $\sum_{k=0}^{n}\binom{n}{k}x^k y^{n-k}=(x+y)^n$ with $y=1$.

The preceding discussions and calculations show that we indeed get the labeled H-boxes as stated in the following diagram:



(17)

We claim that this diagram is equal to the following, which finishes our proof:



To see that this is true, simply decompose the NOT in this diagram using its definition, and apply a Fourier hyper-pivot on the resulting white spider and the one beneath it.