Relating high-level frameworks for quantum circuits

Louis Lemonnier (louis.lemonnier@ens-paris-saclay.fr),
supervised by Aleks Kissinger
Digital Security
Institute for Computer and Information Science, Radboud University Nijmegen

March 18 to August 2, 2019

Contents

Introduction

Physics in the 20th century experienced a great number of breakthroughs, leading to the formation of one of the most well-known fundamental theories: quantum physics. In quantum physics, a studied system can be considered in a *superposition* of classical states at the same time – the final state being determined only with a *measurement*. This new way of thinking helped discover many new phenomena – especially in the world of extremely small particles. Those behaviors are well-known to be counter-intuitive and to be very different from what people can observe with a macroscopic point of view. Many results in quantum physics already have applications and benefit vastly different fields.

Quantum theory has also started influence computer science. Indeed, a *quantum bit* could be in both states 0 and 1 at the same time before a measurement. The possible operations on *qubits* – the short name for quantum bits – are settled by quantum physics. A quantum algorithm can be described as a sequence of operations on qubits.

Quantum circuits are build on the same idea as classical circuits: it is a sequence of gates applied on qubits. Some gates can only apply on one qubit, others can apply on several qubits. All those operations are described in Hilbert spaces. The two basis states of a qubit are $|0\rangle$ and $|1\rangle$ defined as follows:

$$|0\rangle := \begin{bmatrix}1\\0\end{bmatrix} \qquad\qquad |1\rangle := \begin{bmatrix}0\\1\end{bmatrix}$$

A qubit state is defined as a *superposition* of those two states, with α, β complex numbers:

$$|\psi\rangle := \alpha\,|0\rangle + \beta\,|1\rangle\,, \text{ such that } |\alpha|^2 + |\beta|^2 = 1$$

To operate on several qubits, the state of a pair is calculated as the tensor product of the two qubits. We will note for example $|00\rangle := |0\rangle \otimes |0\rangle$. Importantly, it is not always possible to write a state of a pair of qubits as a tensor product of two states – this is what is called *entanglement*.
The † denotes the *adjoint*. The adjoint of the *ket* notation is the *bra* notation: $\langle\psi| := |\psi\rangle^\dagger$.
An important set of gates are the following:

$$[\![-\boxed{H}-]\!] := \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

$$[\![-\boxed{S}-]\!] := \begin{bmatrix}1 & 0\\ 0 & e^{\frac{i\pi}{2}}\end{bmatrix}$$

$$[\![\text{CNOT}]\!] := \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

With these gates, we can make all the Clifford unitaries, an important subset of all unitaries. Operations inside the Clifford group are known to be efficiently simulated by classical computers. Nevertheless, it is far from realizing every possible quantum operation. To do so, it is necessary to add the T-gate and to form the Clifford+T group:

$$[\![-\boxed{T}-]\!] := \begin{bmatrix}1 & 0\\ 0 & e^{\frac{i\pi}{4}}\end{bmatrix}$$

Clifford+T is not efficiently simulated by classical computation.

The *not* operation can be defined as follows:

$$-\boxed{X}- \;=\; -\boxed{H}\boxed{S}\boxed{S}\boxed{H}-$$

The main focus will be the verification of quantum circuits: given two different quantum circuits, being able to affirm where they are equal or not. This can be done by computing the matrix semantics of both circuits, but the size of the matrices is exponential in the number of qubits. My work focuses then on two frameworks used to verify quantum circuits [9, 1] and which scale better than computing matrices of the circuits.

These two frameworks are path-sum [2] and ZH-calculus [3] – actually, it is ZX-calculus [4] that is used for verification for now, but ZX- and ZH-calculus work in the same way. They are respectively introducted in Sections 1 and 2. At first they seem very different from each other, but they are used for the same goal. Thus we will try here to show to what extend they are similar. The idea was also to look for strategy that can borrow to the other. Section 3 is an example of picturing a path-sum law into ZH-calculus, unveiling an new graphical equation. Section 4 builds an new path-sum formalism – *pure* path-sum – in order to describe a bijection between the latter and ZH-calculus in Section 5. Thanks to this bijection, the path-sum rules can be pictured in ZH-calculus as shown in Sections 6 and 7. This helped reduce the number of axioms of ZH-calculus as proven in Section 8. Finally Section 9 reveals that ZH-calculus can also influence path-sum.

1 Path-sum approach

1.1 Definition

Path-sum has been introduced by Feynman as the path integral formulation of quantum mechanics. In quantum theory, there are several ways – or transformations – to go from one state to another. Feynman's idea is to describe the resulting state as a sum of every possible paths taken. The set of possibilities is continuous, that is why we talk about *path integral*. Since quantum gates are operators over a finite dimensional Hilbert space, the integral over all the paths is a discrete sum in computational frameworks.

To understand what comes, some notations must be introduced. $\mathbb{B}$ will be the Boolean set $\{0, 1\}$. $\mathbb{B}^n$ is the set of vectors composed of n Boolean values. $\mathbb{B}^n_*$ is the same set, but excluding the vector only composed of 0s. $\mathbb{D}_M$ – the positive dyadic rationals – is the set of numbers of the form $\frac{a}{2^b}$ with a, b non-negative numbers.

We can describe a path-sum abstractly as a discrete set of *paths* $S \subseteq \mathbb{B}^m$, together with an amplitude function ϕ and state transformation f representing the operator

$$U : |\mathbf{x}\rangle \mapsto \sum_{\mathbf{y} \in S} \phi(\mathbf{x}, \mathbf{y}) \, |f(\mathbf{x}, \mathbf{y})\rangle \, .$$

This general form does not hold any particular computational property, since f and ϕ might be represented in any way. Therefore, a fixed representation as given by Amy [2] is necessary to make the manipulation of these terms possible.

Definition 1.1 (path-sum)**.** An n-qubit *path-sum* ξ consists of

- an *input signature* $|\mathbf{x} = x_1 x_2 \cdots x_n\rangle$ where each x_i is a (distinct) variable or Boolean constant,
- a *global phase* $\alpha \in \mathbb{C}$.
- a *phase polynomial* $P \in \mathbb{D}_M[\mathbf{x}, \mathbf{y}]$ over input variables $\mathbf{x}$ and *path variables* $\mathbf{y} = y_1 y_2 \ldots y_m$, and
- an *output signature* $|f(\mathbf{x}, \mathbf{y}) = f_1(\mathbf{x}, \mathbf{y}) \cdots f_n(\mathbf{x}, \mathbf{y})\rangle$ where each $f_i \in \mathbb{B}[\mathbf{x}, \mathbf{y}]$ is a Boolean polynomial.

The *associated operator* of a path-sum is the partial linear map U_ξ where

$$U_\xi : |\mathbf{x}\rangle \mapsto \alpha \sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i P(\mathbf{x},\mathbf{y})} |f(\mathbf{x},\mathbf{y})\rangle .$$

We say a path variable is *internal* if it does not appear in the output signature, as in the example of $\mathsf{Toffoli}_3$ below (1.2). Our presentation is inspired by descriptions of quantum operators in mathematical texts [11], and as such we write a path-sum informally by the action of its associated operator. By an abuse of notation, we use $|\mathbf{x}\rangle$ to refer to either an input signature or an arbitrary Boolean vector corresponding to an input signature.

Example 1.2. Path-sum representations of common quantum gates and circuits are listed below:

$$\begin{aligned} T &: |x\rangle \mapsto e^{2\pi i \frac{x}{8}} |x\rangle \\ H &: |x\rangle \mapsto \frac{1}{\sqrt{2}} \sum_{y\in\mathbb{B}} e^{2\pi i \frac{xy}{2}} |y\rangle \\ \mathsf{Toffoli}_3 &: |x_1x_2x_3\rangle \mapsto \sum_{y_1,y_2} e^{2\pi i \frac{1}{2}(x_3y_1 + x_1x_2y_1 + y_1y_2)} |x_1x_2y_2\rangle \end{aligned}$$

Path-sum representations are not unique: $\mathsf{Toffoli}_3$ can also be written $|x_1x_2x_3\rangle \mapsto |x_1x_2(x_1x_2 \oplus x_3)\rangle$. Thus we introduce a calculus to be able to go form complicated path-sums to simpler ones.

1.2 Reduction rules

Figure 1 gives the rules of our calculus, presented as algebraic rewrite rules on exponential sums for convenience and applied to path-sums in the obvious way. We write $\xi \longrightarrow \xi'$ to denote that ξ reduces to ξ', and denote by $\longrightarrow^*$ the transitive closure of $\longrightarrow$. For all rules, y_0 is an internal path variable, the polynomials Q are Boolean-valued and whenever $y_i \leftarrow Q$, y_i does not appear in Q. For the [Case] rule, both y_i and y_j are internal.

Before introducing the reduction rules, we define the *lifting* of a Boolean polynomial P, written $\overline{P}$.

$$\begin{aligned} \overline{\mathbf{x}^\alpha} &= \mathbf{x}^\alpha, \\ \overline{P \oplus Q} &= \overline{P} + \overline{Q} - 2\overline{P}\overline{Q}, \end{aligned}$$

where $\mathbf{x}^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ for $\alpha \in \mathbb{B}^n$ is a multi-index.

$$\alpha \sum_{y_0\in\mathbb{B}} \sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left(\frac{1}{4}y_0 + \frac{1}{2}y_0 Q(\mathbf{x},\mathbf{y}) + R(\mathbf{x},\mathbf{y})\right)} |f(\mathbf{x},\mathbf{y})\rangle \longrightarrow \sqrt{2}\alpha \sum_{y\in\mathbb{B}^m} e^{2\pi i\left(\frac{1}{8} - \frac{1}{4}\overline{Q}(\mathbf{x},\mathbf{y}) + R(\mathbf{x},\mathbf{y})\right)} |f(\mathbf{x},\mathbf{y})\rangle \qquad [\omega]$$

$$\alpha \sum_{y_0,y_1\in\mathbb{B}} \sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left(\frac{1}{2}y_0(y_1 + Q(\mathbf{x},\mathbf{y})) + R(\mathbf{x},y_1,\mathbf{y})\right)} |f(\mathbf{x},y_1,\mathbf{y})\rangle \longrightarrow 2\alpha \sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left(R[y_1\leftarrow\overline{Q}]\right)(\mathbf{x},y_1,\mathbf{y})} |(f[y_1 \leftarrow Q])(\mathbf{x},y_1,\mathbf{y})\rangle \qquad [\mathsf{HH}]$$

$$\frac{P(\mathbf{x},\mathbf{y}) = \frac{1}{4}y_i x + \frac{1}{2}y_i(y_j + Q(\mathbf{x},\mathbf{y})) + R(\mathbf{x},\mathbf{y}) = \frac{1}{4}y_j(1-x) + \frac{1}{2}y_j(y_i + Q'(\mathbf{x},\mathbf{y})) + R'(\mathbf{x},\mathbf{y})}{\alpha \sum_{\mathbf{y}\in\mathbb{B}^{m+2}} e^{2\pi i P(\mathbf{x},\mathbf{y})} |f(\mathbf{x},\mathbf{y})\rangle \longrightarrow 2\alpha \sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left((1-x)R[y_j\leftarrow\overline{Q}] + xR'[y_i\leftarrow\overline{Q'}]\right)(\mathbf{x},\mathbf{y})} |f(\mathbf{x},\mathbf{y})\rangle} \qquad [\mathsf{Case}]$$

Figure 1: Path-sum reduction rules

The rules were developed [2] by translating known circuit identities into path-sums, then minimizing the identities to obtain simple interference patterns which strictly reduce the number of path variables. The [HH] rule derives from the equality $HH = I$. The $[\omega]$ rule arises from the identity $(SH)^3 = e^{\frac{2\pi i}{8}} I$, and the final rule [Case] is a specific case distinction needed to prove the 2-qubit Clifford+T identity $\left(\mathrm{CNOT}(X \otimes T)\mathsf{controlled}\text{-}H(X \otimes T^\dagger)\right)^2$ [6].

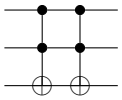
Proposition 1.3 (Correctness)**.** If $\xi \longrightarrow^* \xi''$, then $\xi \equiv \xi'$.

The correctness of our rewrite system follows from direct calculation over symbolic exponential sums. Since the proof is not necessary to understand the other sections, it is left in appendix.

1.3 Examples of path-sum terms and reductions

1.3.1 Two Toffoli gates

It is a known fact that a circuit composed of two Toffoli gates that follow each other is equal to the identity. It is possible to prove it in path-sum.

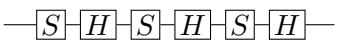


This circuit corresponds to the following path-sum expression. The composition of path-sum is done as presented in the original paper [2].

$$\begin{aligned}
|x_1x_2x_3\rangle &\mapsto \frac{1}{2^2} \sum_{y_1,y_2,y_3,y_4 \in \mathbb{B}^4} e^{2\pi i \frac{1}{2}(x_3y_1 + x_1x_2y_1 + y_1y_2 + y_2y_3 + x_1x_2y_3 + y_3y_4)} |x_1x_2y_4\rangle \\
&\mapsto \frac{1}{2} \sum_{y_1,y_4 \in \mathbb{B}^2} e^{2\pi i \frac{1}{2}(x_3y_1 + y_1y_4)} |x_1x_2y_2\rangle && \textsf{[HH]} \\
&\mapsto |x_1x_2x_3\rangle && \textsf{[HH]}
\end{aligned}$$

1.3.2 The Euler equation

As before, what is called here *Euler equation* is a known circuit semantically equal to the identity up to a phase. This circuit is the following:



Which is written and reduced in path-sum:

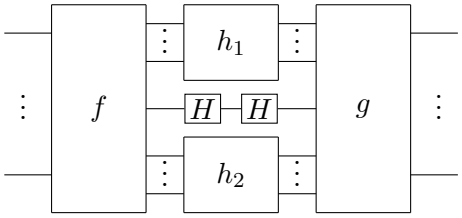
$$\begin{aligned}
|x\rangle &\mapsto \frac{1}{2\sqrt{2}} \sum_{y_1,y_2,y_3 \in \mathbb{B}^3} e^{2\pi i \left(\frac{1}{4}x + \frac{1}{2}xy_1 + \frac{1}{4}y_1 + \frac{1}{2}y_1y_2 + \frac{1}{4}y_2 + \frac{1}{2}y_2y_3\right)} |y_3\rangle \\
&\mapsto \frac{1}{2\sqrt{2}} \sum_{y_1,y_2,y_3 \in \mathbb{B}^3} e^{2\pi i \left(\frac{1}{4}y_1 + \frac{1}{2}y_1(y_2+x) + \frac{1}{4}x + \frac{1}{4}y_2 + \frac{1}{2}y_2y_3\right)} |y_3\rangle \\
&\mapsto \frac{1}{2} e^{\pi i \frac{1}{4}} \sum_{y_2,y_3 \in \mathbb{B}^2} e^{2\pi i \left(-\frac{1}{4}x - \frac{1}{4}y_2 + \frac{2}{4}xy_2 + \frac{1}{4}x + \frac{1}{4}y_2 + \frac{1}{2}y_2y_3\right)} |y_3\rangle && [\omega] \\
&\mapsto \frac{1}{2} e^{\pi i \frac{1}{4}} \sum_{y_2,y_3 \in \mathbb{B}^2} e^{2\pi i \left(\frac{1}{2}xy_2 + \frac{1}{2}y_2y_3\right)} |y_3\rangle \\
&\mapsto e^{\pi i \frac{1}{4}} |x\rangle && \textsf{[HH]}
\end{aligned}$$

1.4 Understanding a path-sum term

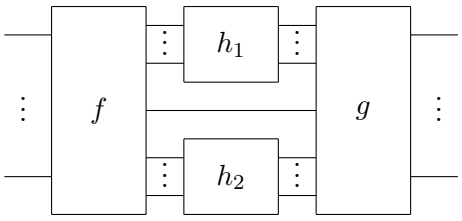
As one can notice above, the path-sum framework gives a straightforward strategy to prove that a circuit is equivalent or not to the identity in some cases. However, these terms reside now far from circuits and it is difficult to visualize what the reduction rules actually do, and particularly the rules involving variable replacement. For example, in [HH], this variable replacement emerges from the deletion of two central variables. As seen in the examples of path-sums above, H gates

introduce path variables – thus eliminating H gates is deleting path variables.

The naïve picture of what [HH] does is the following circuit:



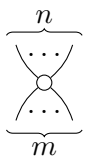
which is reduced to this circuit:



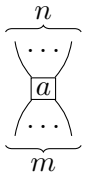
But we will see that [HH] is more than this in a graphical framework – the ZH-calculus. The latter is introduced in the following. We can see that a graphical representation more expressive than circuits is necessary thanks to the examples above: in the middle of the reduction to the identity, it may be difficult – or impossible – to roll back to a circuit represensation of the path-sum.

2 ZH-calculus

The ZH-calculus is a diagrammatic language that represents linear maps between qubits as string diagrams. There are two generators of these diagrams: H-boxes, depicted as white boxes with a complex parameter a, and Z-spiders, depicted as white dots.

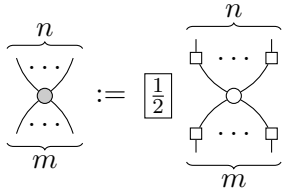


$:= |0\ldots0\rangle\langle 0\ldots0| + |1\ldots1\rangle\langle 1\ldots1|$

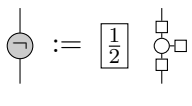


$:= \sum a^{i_1\ldots i_m j_1\ldots j_n} |j_1\ldots j_n\rangle\langle i_1\ldots i_m|$

where in the right-hand equation, the sum runs over all $i_1, \ldots, i_m, j_1, \ldots, j_n \in \{0, 1\}$. An H-box represents a matrix with a as its $|1\ldots1\rangle\langle 1\ldots1|$ entry, and ones elsewhere. By convention, the parameter a is omitted when $a = -1$, so an un-labelled H-box with 1 input and 1 output is a Hadmard gate (up to normalisation).
A pair of diagrams can be composed either by stacking them and joining the outputs of the first with the inputs of the second, which corresponds to composition of linear maps, or placed side-by-side, which represents the tensor product.
For our purposes, we will treat two other spiders as derived generators: the X-spider and the NOT gate.



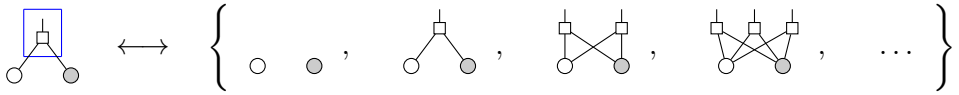
(XS)



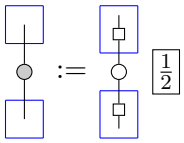
(N)

We consider ZH-diagrams to be equal when they can be topologically deformed into one another [4, 5], preserving the order of in- and outputs, or when they can be transformed into another another by using the rules of the ZH-calculus. The ZH-calculus is *complete* in the sense that any

equality between matrices can be proven just from diagrammatic rules [3], and Figure 2 contains the complete set of rules. As a result, any two diagrams that are equal as linear maps, can be proven to be equal using some combination of these rules.
The calculations in this paper will greatly benefit from the use of !-box – « bang box » – notation [8]. !-boxes represent parts of a diagram that may be replicated an arbitrary number of times, and thus allow one to express a whole family of diagrams at once.

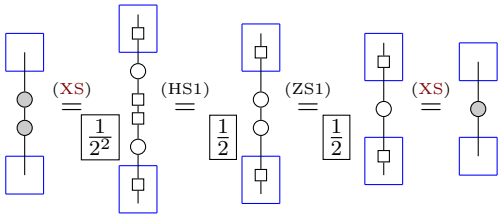


When used in equations, corresponding !-boxes on either side of the equation should be replicated an equal number of times. For example, the grey spider (XS) can be redefined.

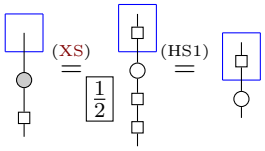


(XS)

and one can straightforwardly prove for example that grey spiders fuse just like white spiders (XS1), and that we have a usual color changing law (CC):



(XS1)



(CC)

To make the equations that follow more readable, it is necessary to picture the !-boxed versions of the rules that are used in this report.

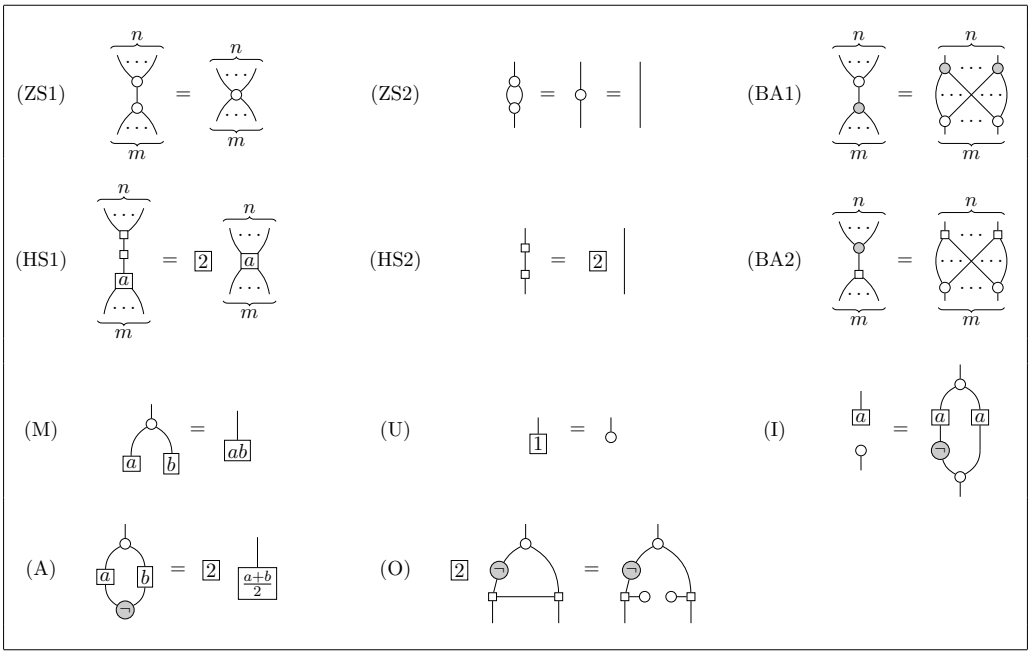
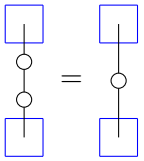
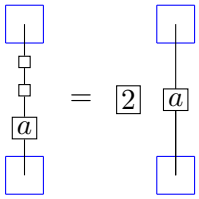
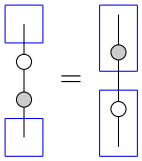
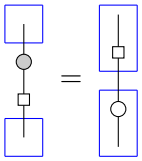
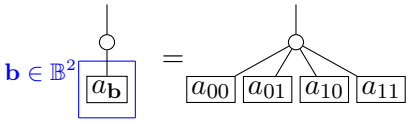


Figure 2: The rules of the ZH-calculus. Throughout, m, n are nonnegative integers and a, b are arbitrary complex numbers. The right-hand sides of both *bialgebra* rules (BA1) and (BA2) are complete bipartite graphs on $(m+n)$ vertices, with an additional input or output for each vertex.

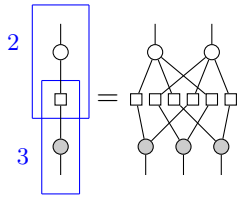
(ZS1)  (HS1)  (BA1)  (BA2) 

Most of the coming !-boxes will be annotated, as in the example that follows:



When a !-box are annotated with a given integer n, there are exactly n copies of what is inside this !-box. A !-box annotated with a set introduces a variable, and pictures the fact that there is a copy of what contains the !-box for each element of the set.

Furthermore, !-boxes can overlap as in this example:



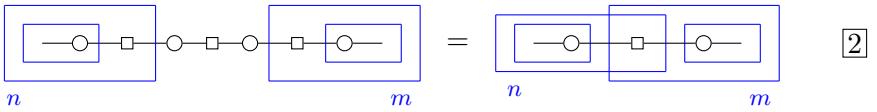
These annotations and overlapping are required to understand the next section.

3 Hyper pivot

In the long process of picturing diagrammatically the [HH] rule, the first result that came out was the *hyper pivot* – which is also called *regular* hyper pivot, since it does not completely picture the [HH] rule. Even if it is a particular case of the action of the path-sum law, it is a first example to illustrate the use of !-boxes in a complex manner.

The name hyper pivot refers to the pivot equation [7] of ZX-calculus, in our case involving hyper graphs.

Theorem 3.1 (Hyper pivot)**.** For n, m two positive numbers:



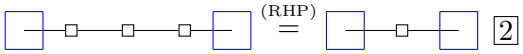
(RHP)

As in the [HH] rule, the two white dots in the middle of the left-side diagram are *internal* variables – or internal white dots. Pivoting on these dots connects any group of white dots from the left-hand side to any group of white dots from the right-hand side with a multi-legged H-box. It can be seen as all the n H-boxes on the left are copied m times; hence the $n \times m$ H-boxes in the resulting diagram.
Diagrammatic equations in these sections seem oriented – either left to right, or top to bottom –, but there is no such thing: in ZH-calculus, any wire can be bended in any direction. The representation of the equation is only a choice to make them more readable.

3.1 Applications

Lemma 3.2. (HS1) is an instance of (RHP).

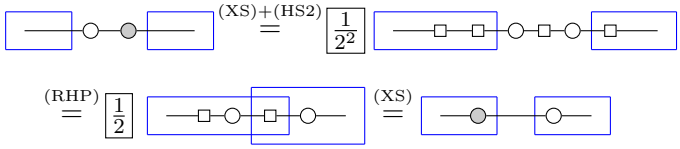
Proof.



□

Lemma 3.3. (BA1) is an instance of (RHP).

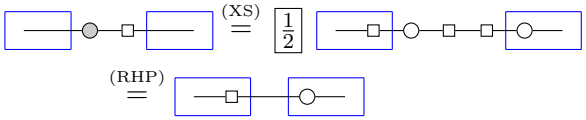
Proof.



□

Lemma 3.4. (BA2) is an instance of (RHP).

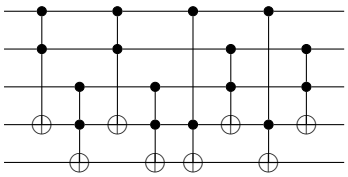
Proof.



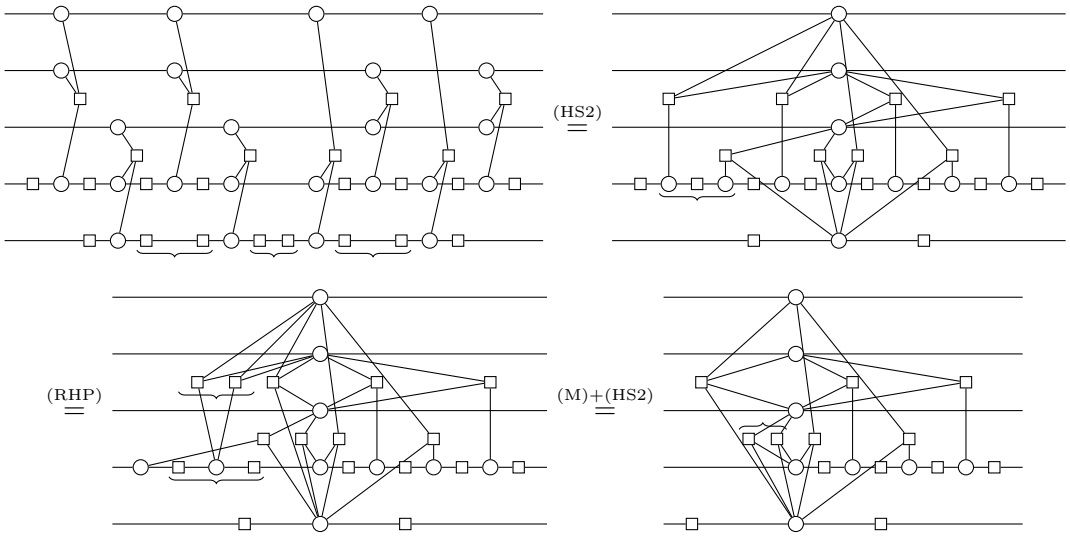
□

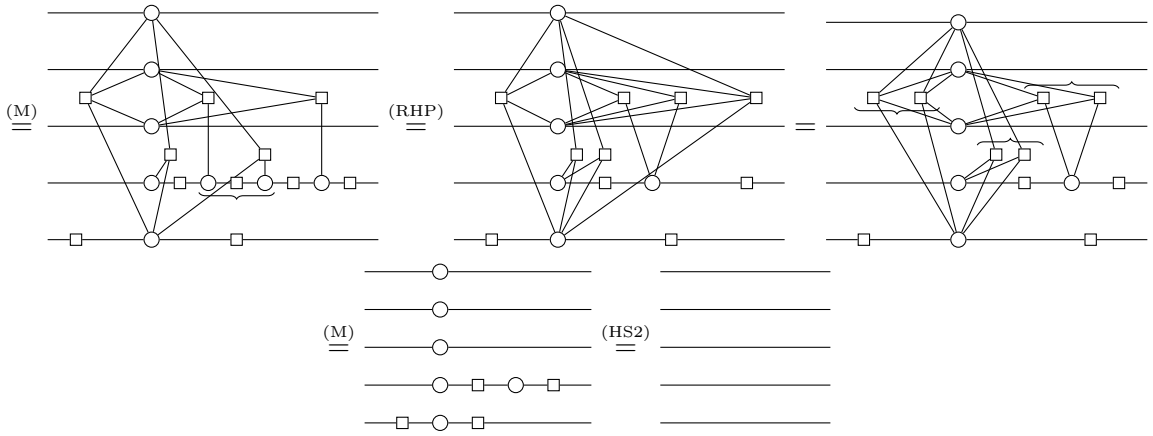
3.2 Bigger example

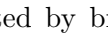
We are going to show the use of this rule on an example: proving that the circuits A and B are equal.

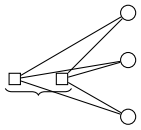


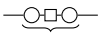
Which becomes in ZH-calculus:





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:

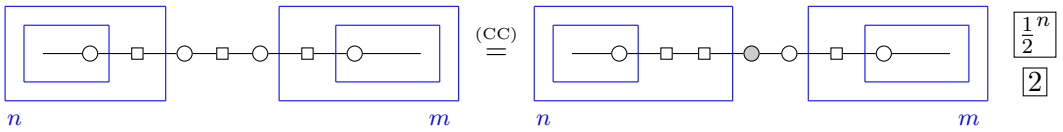


Finally, the other operations are (RHP) applications, and highlighted this way:  .

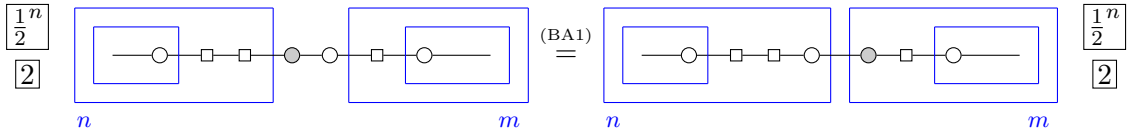
3.3 Proof of hyper pivot

The proof of Theorem 3.1 consists on a sequence of well-chosen bialgebras and color changes. The color changes are responsible for the opverlapping !-boxes, which is natural: there must be a created H-box on every wire.

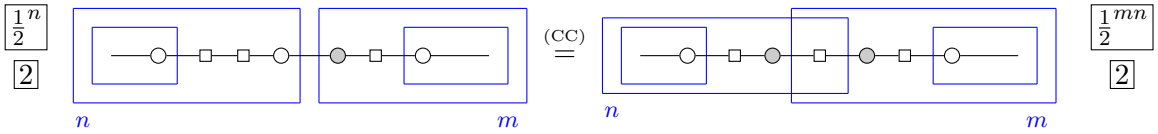
1st step, push the middle H-box to the left.



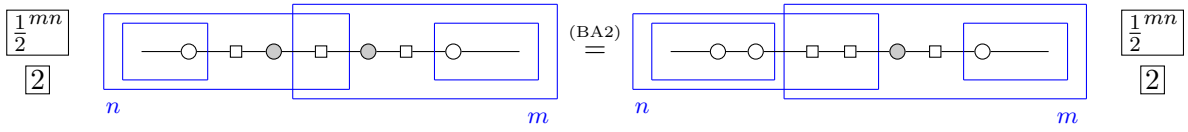
2nd step, BA1 in the middle.



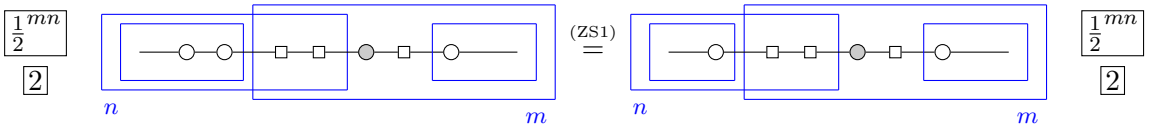
3rd step, push the H-box from the left.



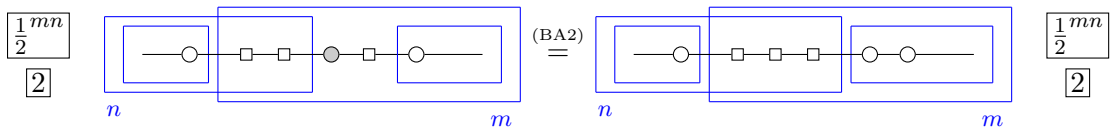
4th step, BA2 on the left.



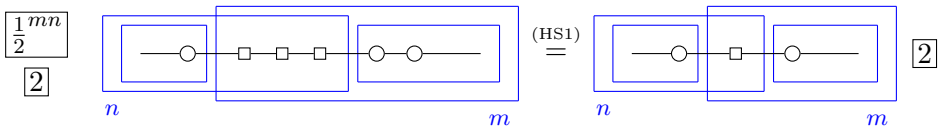
5th step, spider fusion on the left.



6th step, BA2 on the right.



7th step, HS1 in the middle and spider fusion on the right.



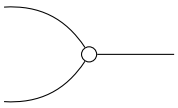
4 Pure path-sum

In order to understand the path-sum rules diagrammatically, and reversely ZH rules in path-sum, it was decisive to have a translation between the two formalisms.

The aim of this section is to adjust path-sum to be able to build a bijection between ZH diagrams and path-sums. This bijection should also respect the correspondance between laws in the two – which is discribed later. In this section, we present the necessity to introduce a *pure* path-sum approach in order to have a bijection.

A pure path-sum expression does not have inputs, only outputs but keeping a memory of whether they are inputs or outputs in another function. Besides, the output signature can no longer be composed of functions, only a list of variables.

This is necessary because otherwise, there is not any proper way to express in a path-sum the ZH diagram composed of a white dot, two inputs and an output.



A path-sum term is now expressed that way :

Definition 4.1 (pure path-sum)**.** An n-qubit *pure path-sum* ξ consists of

- an *input signature* $|\mathbf{x} = x_1 x_2 \cdots x_n\rangle$ where each x_i is a (distinct) variable or Boolean constant,
- a *phase* ϕ over *path variables* $\mathbf{y} = y_1 y_2 \dots y_m$,
- an *output signature* $|\ell(\mathbf{y})\rangle$ which is a list of path variables, and
- a *memory function* $C \in [m]^{\mathbb{B}^n}$ such as $C(\mathbf{x}) \subseteq \ell$, which maps each element of $\mathbf{x}$ to its corresponding y in ℓ.

The *associated vector* of a pure path-sum is $|\psi\rangle$ where:

$$|\psi\rangle : \sum_{\mathbf{y}} \phi(\mathbf{y})|\ell(\mathbf{y})\rangle.$$

To have a better picture of pure path-sum, here are some examples:

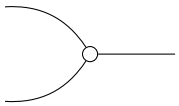
Example 4.2. Path-sum representations of common quantum gates and circuits are listed below:

$$T : |x\rangle \mapsto \begin{cases} x \mapsto y_1 \\ \sum\limits_{y_1} e^{i\frac{\pi}{4}y_1} |y_1 y_1\rangle \end{cases}$$

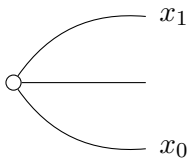
$$H : |x\rangle \mapsto \begin{cases} x \mapsto y_1 \\ \frac{1}{\sqrt{2}} \sum\limits_{y_1,y_2} (-1)^{y_1 y_2} |y_1 y_2\rangle \end{cases}$$

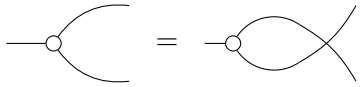
$$\mathsf{CNOT} : |x_1 x_2\rangle \mapsto \begin{cases} x_1, x_2 \mapsto y_1, y_2 \\ \frac{1}{2\sqrt{2}} \sum\limits_{y_1,y_2,y_3,y_4} (-1)^{y_1 y_3} (-1)^{y_2 y_3} (-1)^{y_4 y_3} |y_1 y_1 y_2 y_4\rangle \end{cases}$$

With that formalism, we can express the problematic example from earlier:

 can now be translated as $\begin{cases} C : (x_0, x_1) \mapsto (y, y) \\ \sum\limits_{y} |yyy\rangle \end{cases}$

Which is basically corresponding to the following diagram:



As you can see, a white dot in the graphical calculus corresponds to a *path variable* in path-sum. Besides, two white dots connected by a positive number of wires can be fused – that is to say, they actually correspond to a unique path variable. Since  holds, any output wire can be chosen for the inputs in any order.

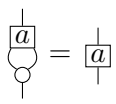
The [HH] law presented at the beginning of this report will be changed to fit this pure approach, and it does not change its reduction power concerning verification.

The original [HH] law authorizes the replacement of a variable by a Boolean polynomial in the output signature. However our new definition, the output signature is a list of variables – it can no longer contain polynomials. Thus, it is not possible to replace a variable that appears in the output signature. The pure [HH] is now applied only when y_0, y_1 are interior variables, expect when the polynomial that is going to replace y_1 is also a variable.

5 Towards a bijection

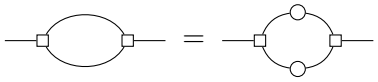
We define $\mathrm{ZH}^\dagger = \mathrm{ZH}/\{(ZS1), (ZS2), (M), (U), (SV)\}$. (ZS1), (ZS2), (M), (U), (SV) are *natural* in path-sum, that is why we consider classes of ZH-diagrams.

Lemma 5.1. A 3-ary white node and H-box that are doubly connected reduces to a single 2-ary H-box:

 (SV)

5.1 ZH to pure path-sum

Let us consider a $\mathrm{ZH}^\dagger$ diagram D. It is composed of white dots and of H-boxes with phases. Since two diagrams equal by spider fusion are in the same class, we fuse as many white dots as we can, such that no white dot is directly connected to another white dot. Similarly, let us put a white dot between every connected H-boxes, for example:

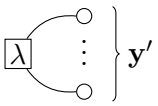


Now every wire is either a connection between a white dot and an H-box, or is an output wire.

Let us name all the white dots y_0, y_1, etc. The order does not matter. Every H-box is then characterized by a complex phase λ and a list of white dots $\mathbf{y'}$, which are the white dots connected to that box. Remember that an empty H-box actually has a phase of -1. The *phase* ϕ of the resulting path-sum is the product of all the terms $\lambda^{\mathbf{y}'}$. The output signature is naturally produced observing the output wires. The resulting path-sum is written $[\![D]\!]_{\mathrm{ZH}^\dagger \to \mathrm{PPS}}$.

5.2 Pure path-sum to ZH

Let us consider a pure path-sum ξ. To draw the corresponding diagram, let us create as many white dots as path variables in ξ. Then for every term $\lambda^{\mathbf{y}'}$, we draw a λ-labelled H-box connected to all the white dots of $\mathbf{y}'$.



Then we can put some more wires to the output white dots, and bend those which were meant to be inputs (stored in C). The resulting diagram is noted $[\![E]\!]_{\mathrm{PPS} \to \mathrm{ZH}^\dagger}$.

Theorem 5.2. $[\![[\![.]\!]_{\mathrm{PPS} \to \mathrm{ZH}^\dagger}]\!]_{\mathrm{ZH}^\dagger \to \mathrm{PPS}} = \mathbb{1}$ and $[\![[\![.]\!]_{\mathrm{ZH}^\dagger \to \mathrm{PPS}}]\!]_{\mathrm{PPS} \to \mathrm{ZH}^\dagger} = \mathbb{1}$

6 Fourier hyper pivot

6.1 Notations and the Fourier transform

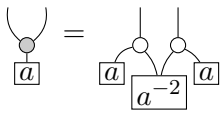
The translation from path-sum to ZH-calculus helps to see that the connected H-boxes to the internal variables on which one can pivot are not necessarily phase-free: one side can have phases. However in this case, we must picture diammagratically this term:

$$\lambda^{\overline{Q}}$$

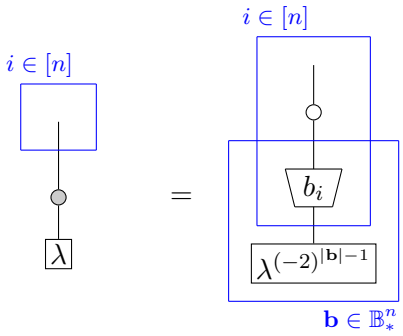
For example, if $Q = x_1 \oplus x_2$, $\overline{Q} = x_1 + x_2 - 2x_1x_2$, which makes:

$$\lambda^{x_1+x_2-2x_1x_2} = \lambda^{x_1}\lambda^{x_2}\left(\lambda^{-2}\right)^{x_1x_2}$$

The corresponding diagram is similar to a Fourier transform [10]:



which has a general version with !-boxes:

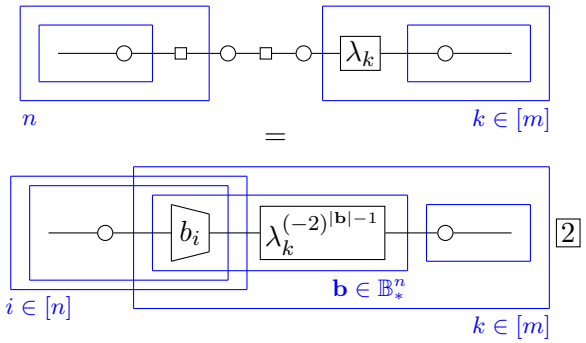


with a new notation:

Each of these tools above are necessary to completely picture the [HH] law in ZH-calculus. The resulting diagrammatic law will be named *Fourier hyper pivot*.

6.2 General theorem

Theorem 6.1 (Fourier hyper pivot)**.** λ can be any list of complex numbers.



(FHP)

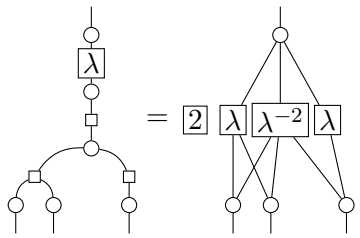
The left-hand side of the first diagram pictures the polynomial Q, and the right-hand side is what is left but only the parts connected to y_1, since the other parts of the path-sum are not modified by [HH], it is not necessary to picture them in a diagrammatic equation. As explained above, the Fourier transform comes from the *lifting* of Q.

The proof of Theorem 6.1 is left in appendix.

6.3 Simple examples

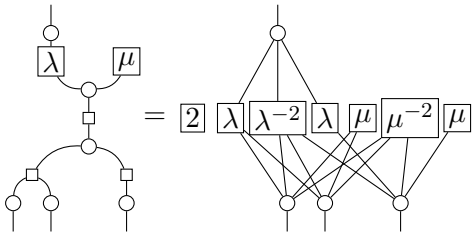
The (FHP) equation involves an undecent number of !-boxes, this is why simple examples might be easier to handle.

6.3.1 First example



In this example, $n = 2$, $m = 1$ and one of the *white dots groups* inside the n-labelled !-box is composed of two white dots, the other only one. Since $n = 2$, the labels in H-boxes are λ and λ^{-2}. The λ-boxes are only connected to one group of white dots, the λ^{-2}-box is connected to both groups.

6.3.2 Second example



This example is similar to the previous one, only a μ-box is added, to demonstrate that the H-boxes can also be connected to nothing at all.

7 Case hyper pivot

7.1 Path-sum rule and proof

The original [Case] rule in figure 1 can be generalized this way:

$$\sum_{y_0,y_1,\mathbf{y}} \varphi^X (-1)^{y_0 Q} (-1)^{y_0 y_1} (-1)^{y_1 Q'} \psi^{1-X} |\ell\rangle \to 2 \sum_{\mathbf{y}} (-1)^{QQ'} \varphi[y_0 \leftarrow \overline{Q'}]^X \psi[y_1 \leftarrow \overline{Q}]^{1-X} |\ell\rangle$$

where φ is a complex function over y_0 and $\mathbf{y}$; ψ over y_0 and $\mathbf{y}$; X is a boolean polynomial over $\mathbf{y}$; Q, Q' are boolean polynomials over $\mathbf{y}$ and y_0, y_1 are not present in ℓ.

When $\mathbf{y}$ is fixed:

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

when applied the same idea as the [HH] rule on both cases, it is proven equal to:

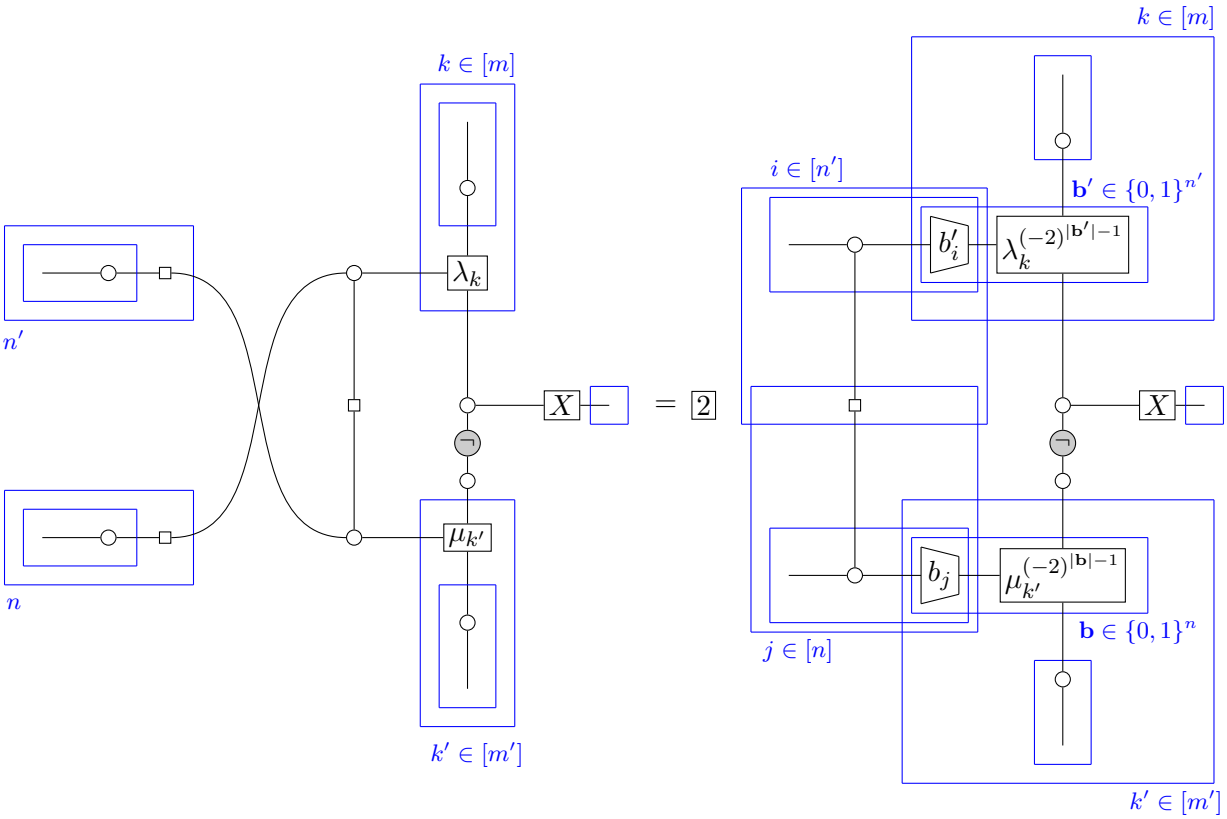
$$\begin{cases} 2(-1)^{QQ'} \psi[y_1 \leftarrow \overline{Q}] \text{ if } X = 0 \\ 2\varphi[y_0 \leftarrow \overline{Q'}](-1)^{QQ'} \text{ if } X = 1 \end{cases}$$

which is equal to

$$2(-1)^{QQ'} \varphi[y_0 \leftarrow \overline{Q'}]^X \psi[y_1 \leftarrow \overline{Q}]^{1-X}$$

7.2 Diagramatic version

Here is the ZH translation of the path-sum rule:



The boxed X is a Boolean polynomial, meaning a sequence of sum –  – and product –  – operations.

7.3 What we can learn from it

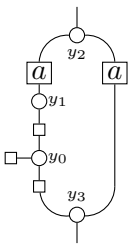
The [Case] rule was created out of a two-qubit equation [6] and translated into the path-sum framework [2]. This rule appears at first as a *special case* – a rule that would not be used much. But we can show that both sides of the case rule are [HH] reductions of a common diagram. This provides a general receipe to create new path-sum rules: from a diagram, compute all its [HH] reductions until they are not reductible anymore; then every couple of resulting diagrams can make a new rule, given that they have a different number of variables.

8 New ZH set of axioms

In the section presenting the *regular hyper pivot*, we have shown that the axioms (HS1), (BA1) and (BA2) are all included into (RHP). We will now prove that with (FHP), the (I) law is also induced.

Lemma 8.1. (FHP) includes (I).

Proof. This can be proven with the path-sum [HH] law – then also with the Fourier hyper pivot.



This diagram corresponds to the following path-sum expression.

$$\sum_{y_0,y_1,y_2,y_3} \sum_{\mathbf{y}} (-1)^{y_0+y_0y_1+y_0y_3} a^{y_1y_2} a^{y_2y_3} \phi(y_2,y_3,\mathbf{y}) |\ell(y_2,y_3,\mathbf{y})\rangle$$

Which is reduced when applying $[\mathsf{HH}]_{y_0,y_1}$ to:

$$2 \sum_{y_2,y_3} \sum_{\mathbf{y}} a^{y_2} (a^{-1})^{y_2y_3} a^{y_2y_3} \phi(y_2,y_3,\mathbf{y}) |\ell(y_2,y_3,\mathbf{y})\rangle$$

$$2 \sum_{y_2,y_3} \sum_{\mathbf{y}} a^{y_2} \phi(y_2,y_3,\mathbf{y}) |\ell(y_2,y_3,\mathbf{y})\rangle$$

That is exactly the result. □

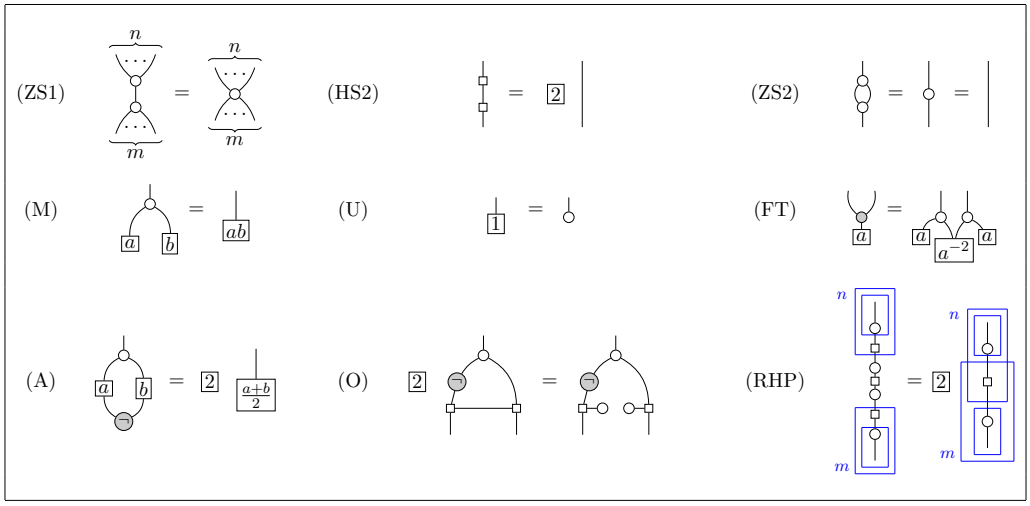


Figure 3: The rules of the ZH-calculus. Throughout, m, n are nonnegative integers and a, b are arbitrary complex numbers.

Theorem 8.2. The set of rules in Figure 3 is complete for ZH-calculus.

Proof. As seen in lemmas 3.2, 3.3 and 3.4, (HS1), (BA1) and (BA2) can all be replaced by (RHP). Moreover, (RHP) and (FT) are enough to prove (FHP) – Fourier hyper pivot – since it is only proven through color changes, bialgebras and Fourier transform. Then (RHP) and (FT) can prove (I), thanks to lemma 8.1. The set of axioms in Figure 2 being complete, the one in Figure 3 is also complete. □

9 Future work

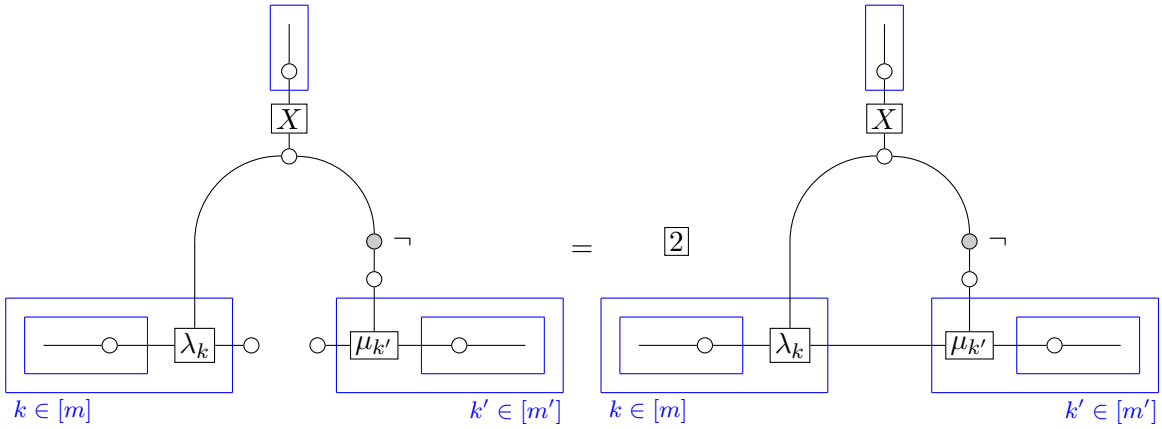
ZH-calculus and path-sum may still have a lot to learn from one another, there must be some reductions that are more natural in path-sum, which could be then translated into ZH-calculus. Or the other way around, an equation of ZH-calculus that could be a reduction into path-sum. This happened with the (O) law in ZH-calculus, which appears to reduce the number of variables.

9.1 Ortho

Ortho is an equation of the ZH-calculus, that we can now translate into path-sum. A generalized version of this law is the following:

$$\sum_{y_0,y_1,\mathbf{y}} \varphi(y_0)^X \psi(y_1)^{1-X} \phi' |\ell\rangle \to 2 \sum_{y_0,\mathbf{y}} \varphi(y_0)^X \psi(y_0)^{1-X} \phi' |\ell\rangle$$

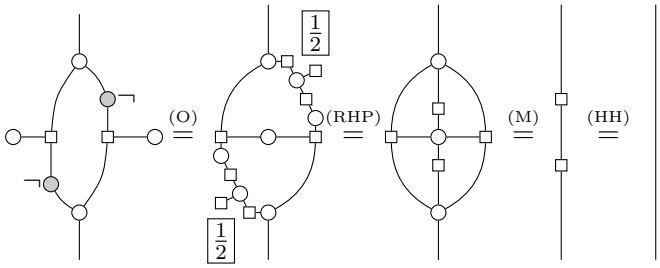
Going back to the world of diagrams now, we obtain this:



The above ZH equation is the complete !-boxed version of (O).

9.1.1 Example of use

An example of application of (O) is



Then adding this new law to path-sum would increase its power in reducing term to the identity. The (O) law has not yet been proven independant from the other axioms of ZH-calculus nor is proven using those axioms; it is likely that it is an independant axiom.

References

[1] Matthew Amy: *Feynman*. https://github.com/meamy/feynman.

[2] Matthew Amy (2018): *Towards Large-scale Functional Verification of Universal Quantum Circuits*. *CoRR* abs/1805.06908. Available at http://arxiv.org/abs/1805.06908.

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

[4] Bob Coecke & Ross Duncan (2009): *Interacting Quantum Observables: Categorical Algebra and Diagrammatics*. *CoRR* abs/0906.4725. Available at http://arxiv.org/abs/0906.4725.

[5] 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. Available at https://doi.org/10.1007/978-3-319-91376-6_6.

[6] Bob Coecke & Quanlong Wang (2018): *ZX-Rules for 2-Qubit Clifford+T Quantum Circuits*. In: *Reversible Computation - 10th International Conference, RC 2018, Leicester, UK, September 12-14, 2018, Proceedings*, pp. 144–161, doi:10.1007/978-3-319-99498-7_10. Available at https://doi.org/10.1007/978-3-319-99498-7_10.

[7] Ross Duncan & Simon Perdrix (2013): *Pivoting makes the ZX-calculus complete for real stabilizers*. doi:10.4204/EPTCS.171.5.

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

[9] Aleks Kissinger & John van de Wetering: *PyZX*. https://github.com/Quantomatic/pyzx.

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

[11] Michael A. Nielsen & Isaac L. Chuang (2011): *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th edition. Cambridge University Press, New York, NY, USA.

Acknowledgements

I would like to thank my supervisor Aleks who has been guiding me during my whole internship and listening to my ideas. I would have never done that much without him. Some very special thanks to John, with whom I have shared my work and thoughts and from whom I have learnt a lot. I also thank Kang Feng for our interesting conversations, and in general every person in Nijmegen that made my stay there enjoyable.

Appendix

Correctness of the path-sum rules

$$
\begin{aligned}
[\omega]: \quad & \sum_{y_0\in\mathbb{B}}\sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left(\frac{1}{4}y_0+\frac{1}{2}y_0Q(\mathbf{x},\mathbf{y})+R(\mathbf{x},\mathbf{y})\right)}\,|f(\mathbf{x},\mathbf{y})\rangle \\
&= \sum_{\mathbf{y}\in\mathbb{B}^m}\left(1+e^{2\pi i\left(\frac{1}{4}+\frac{1}{2}Q(\mathbf{x},\mathbf{y})\right)}\right)e^{2\pi iR(\mathbf{x},\mathbf{y})}\,|f(\mathbf{x},\mathbf{y})\rangle \\
&= \begin{cases} \sum_{\mathbf{y}\in\mathbb{B}^m}(1+i)\,e^{2\pi iR(\mathbf{x},\mathbf{y})}\,|f(\mathbf{x},\mathbf{y})\rangle & \text{if } Q(\mathbf{x},\mathbf{y})=0 \mod 2 \\ \sum_{\mathbf{y}\in\mathbb{B}^m}(1-i)\,e^{2\pi iR(\mathbf{x},\mathbf{y})}\,|f(\mathbf{x},\mathbf{y})\rangle & \text{if } Q(\mathbf{x},\mathbf{y})=1 \mod 2 \end{cases} \\
&= \begin{cases} \sqrt{2}\sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left(\frac{1}{8}+R(\mathbf{x},\mathbf{y})\right)}\,|f(\mathbf{x},\mathbf{y})\rangle & \text{if } Q(\mathbf{x},\mathbf{y})=0 \mod 2 \\ \sqrt{2}\sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left(\frac{1}{8}+\frac{3}{4}+R(\mathbf{x},\mathbf{y})\right)}\,|f(\mathbf{x},\mathbf{y})\rangle & \text{if } Q(\mathbf{x},\mathbf{y})=1 \mod 2 \end{cases} \\
&= \sqrt{2}\sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left(\frac{1}{8}+\frac{3}{4}\overline{Q}(\mathbf{x},\mathbf{y})+R(\mathbf{x},\mathbf{y})\right)}\,|f(\mathbf{x},\mathbf{y})\rangle
\end{aligned}
$$

$$
\begin{aligned}
[\mathsf{HH}]: \quad & \sum_{y_0\in\mathbb{B}}\sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left(\frac{1}{2}y_0(y_i+Q(\mathbf{x},\mathbf{y}))+R(\mathbf{x},\mathbf{y})\right)}\,|f(\mathbf{x},\mathbf{y})\rangle \\
&= \sum_{\mathbf{y}\in\mathbb{B}^m}\left(1+e^{2\pi i(y_i+Q(\mathbf{x},\mathbf{y}))}\right)e^{2\pi iR(\mathbf{x},\mathbf{y})}\,|f(\mathbf{x},\mathbf{y})\rangle \\
&= \sum_{\mathbf{y}\in\mathbb{B}^m,\, y_i=Q(\mathbf{x},\mathbf{y}) \mod 2}\left(1+e^{2\pi i(2k)}\right)e^{2\pi iR(\mathbf{x},\mathbf{y})}\,|f(\mathbf{x},\mathbf{y})\rangle \\
&\quad + \sum_{\mathbf{y}\in\mathbb{B}^m,\, y_i=1+Q(\mathbf{x},\mathbf{y}) \mod 2}\left(1+e^{2\pi i(2k+1)}\right)e^{2\pi iR(\mathbf{x},\mathbf{y})}\,|f(\mathbf{x},\mathbf{y})\rangle \\
&= 2\sum_{\mathbf{y}\in\mathbb{B}^m,\, y_i=Q(\mathbf{x},\mathbf{y}) \mod 2} e^{2\pi iR(\mathbf{x},\mathbf{y})}\,|f(\mathbf{x},\mathbf{y})\rangle \\
&= 2\sum_{\mathbf{y}\in\mathbb{B}^{m-1}} e^{2\pi i\left(R[y_i\leftarrow\overline{Q}]\right)(\mathbf{x},\mathbf{y})}\,|(f[y_i\leftarrow Q])\,(\mathbf{x},\mathbf{y})\rangle
\end{aligned}
$$

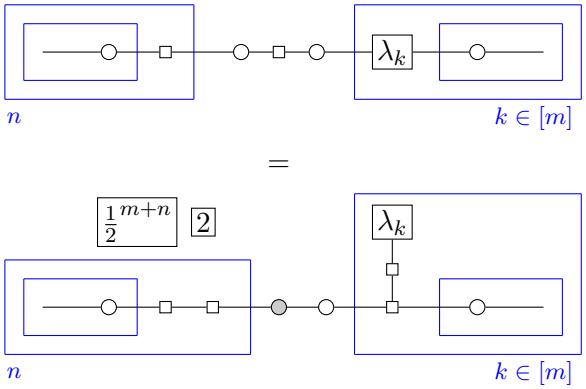
[Case]: Recall the precondition

$$P(\mathbf{x},\mathbf{y}) = \frac{1}{4}y_ix + \frac{1}{2}y_i(y_j+Q(\mathbf{x},\mathbf{y})) + R(\mathbf{x},\mathbf{y}) = \frac{1}{4}y_j(1-x) + \frac{1}{2}y_j(y_i+Q'(\mathbf{x},\mathbf{y})) + R'(\mathbf{x},\mathbf{y}).$$

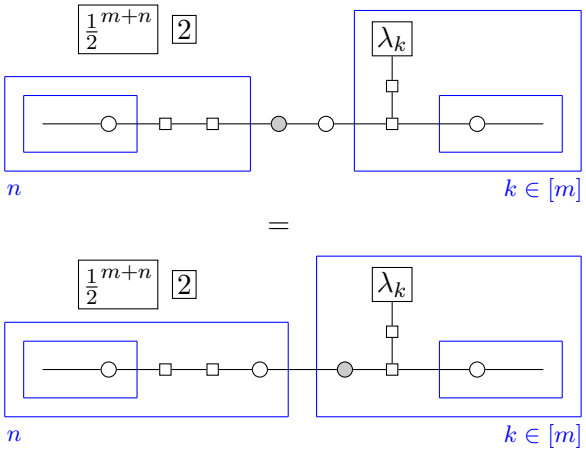
$$
\begin{aligned}
& \sum_{\mathbf{y}\in\mathbb{B}^{m+2}} e^{2\pi iP(\mathbf{x},\mathbf{y})}\,|f(\mathbf{x},\mathbf{y})\rangle \\
&= \begin{cases} \sum_{\mathbf{y}\in\mathbb{B}^{m+2}} e^{2\pi i\left(\frac{1}{2}y_i(y_j+Q(\mathbf{x},\mathbf{y}))+R(\mathbf{x},\mathbf{y})\right)}\,|f(\mathbf{x},\mathbf{y})\rangle & \text{if } x=0 \\ \sum_{\mathbf{y}\in\mathbb{B}^{m+2}} e^{2\pi i\left(\frac{1}{2}y_j(y_i+Q'(\mathbf{x},\mathbf{y}))+R'(\mathbf{x},\mathbf{y})\right)}\,|f(\mathbf{x},\mathbf{y})\rangle & \text{if } x=1 \end{cases} \\
&= \begin{cases} 2\sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left(R[y_j\leftarrow\overline{Q}]\right)(\mathbf{x},\mathbf{y})}\,|(f[y_j\leftarrow Q])\,(\mathbf{x},\mathbf{y})\rangle & \text{if } x=0 \\ 2\sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left(R'[y_i\leftarrow\overline{Q'}]\right)(\mathbf{x},\mathbf{y})}\,|(f[y_i\leftarrow Q'])\,(\mathbf{x},\mathbf{y})\rangle & \text{if } x=1 \end{cases} \qquad \text{by } [\mathsf{HH}] \\
&= 2\sum_{\mathbf{y}\in\mathbb{B}^m} e^{2\pi i\left((1-x)R[y_j\leftarrow\overline{Q}]+xR'[y_i\leftarrow\overline{Q'}]\right)(\mathbf{x},\mathbf{y})}\,|f(\mathbf{x},\mathbf{y})\rangle \qquad \text{since } y_i, y_j\notin f
\end{aligned}
$$

Proof of Fourier hyper pivot

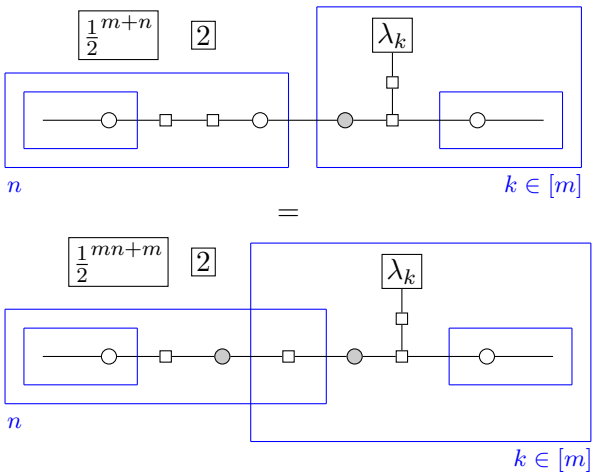
1st step, push the the H box from the middle and unfold the λ H boxes.



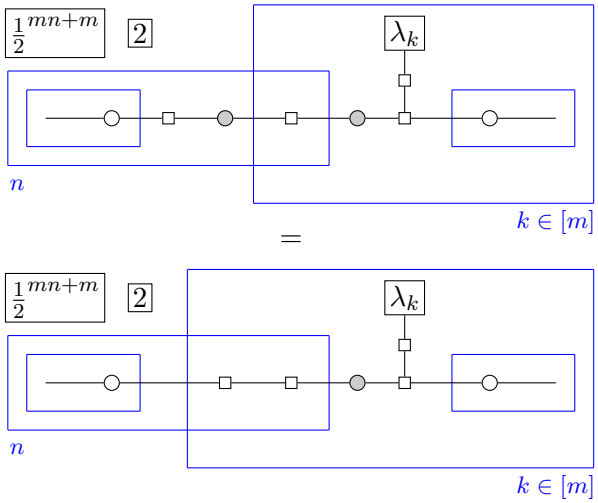
2nd step, BA1 between the two dots in the middle.



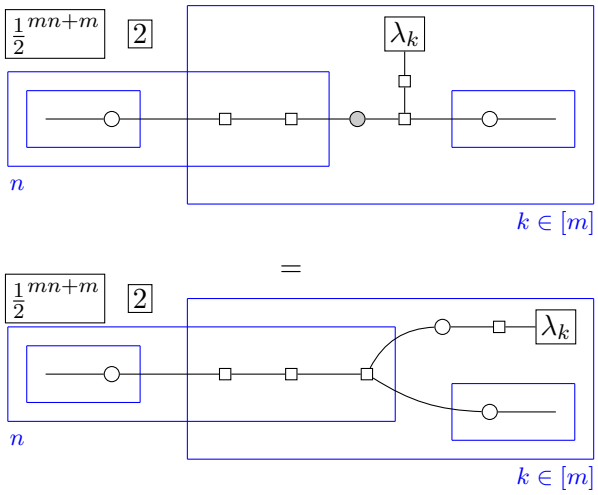
3rd step, push the H box from the left.



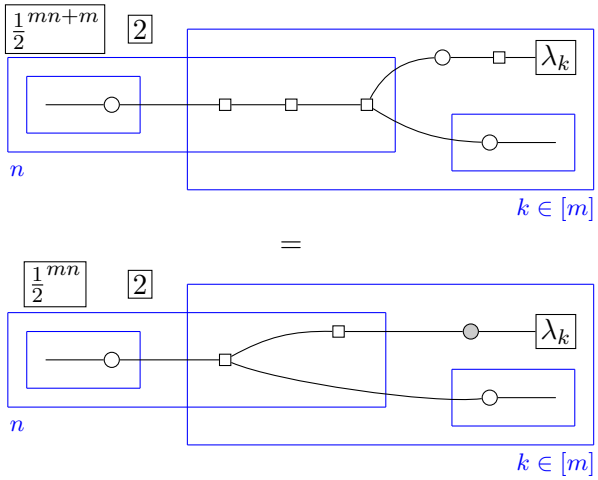
4th step, BA2 on the left-hand side.



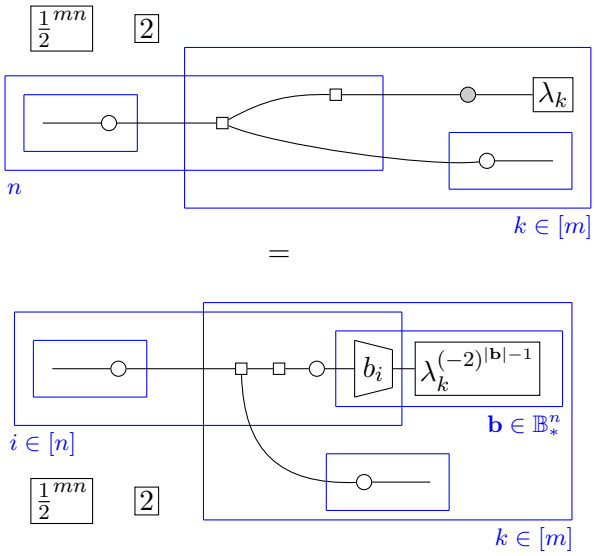
5th step, BA2 on the right-hand side.



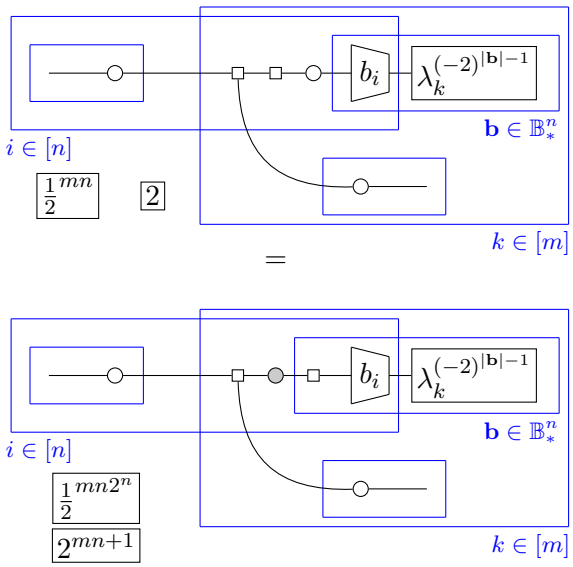
6th step, push the H box from the right.



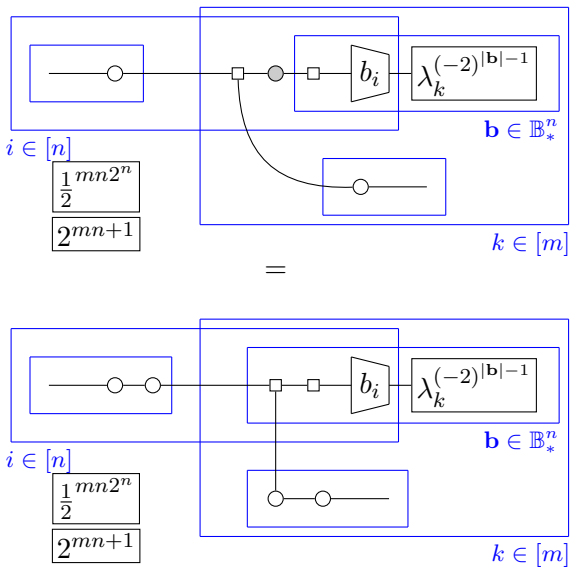
7th step, Fourier transform.



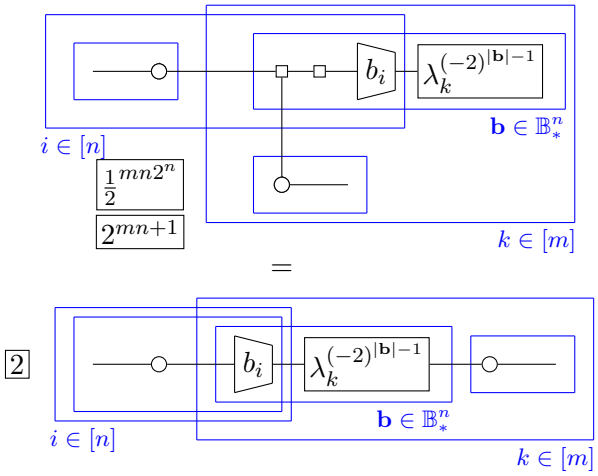
8th step, push H box from the left.



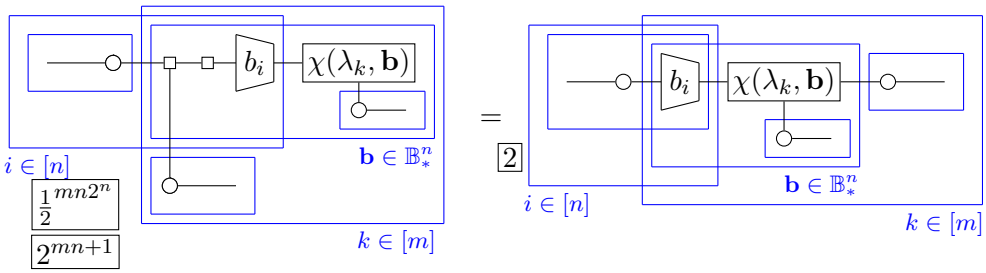
9th step, BA2 on the left-hand side.



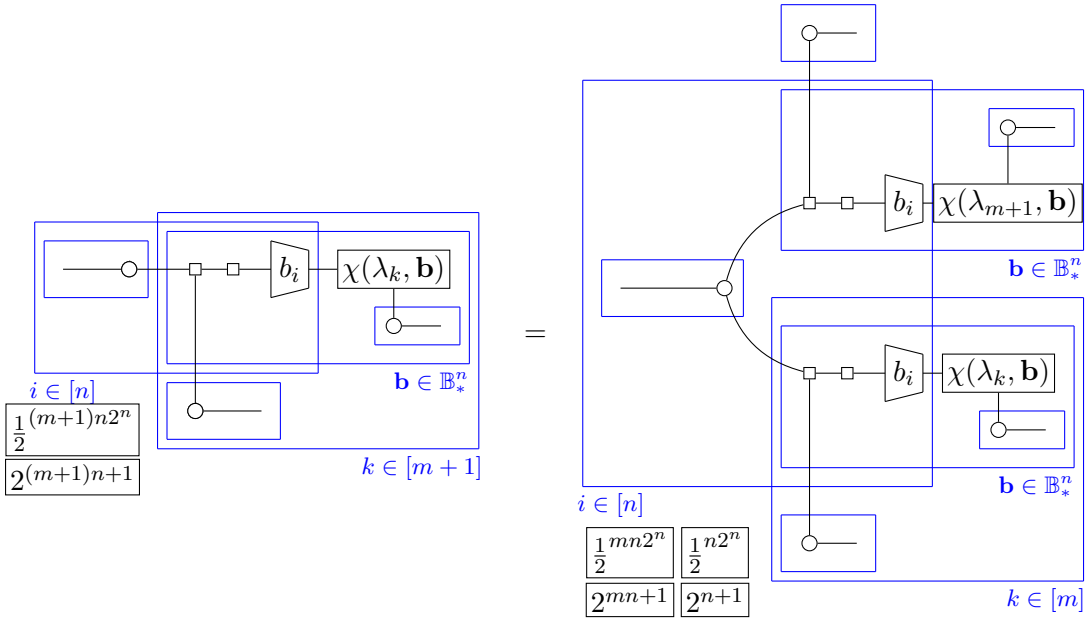
Final step, proven in the following lemma.



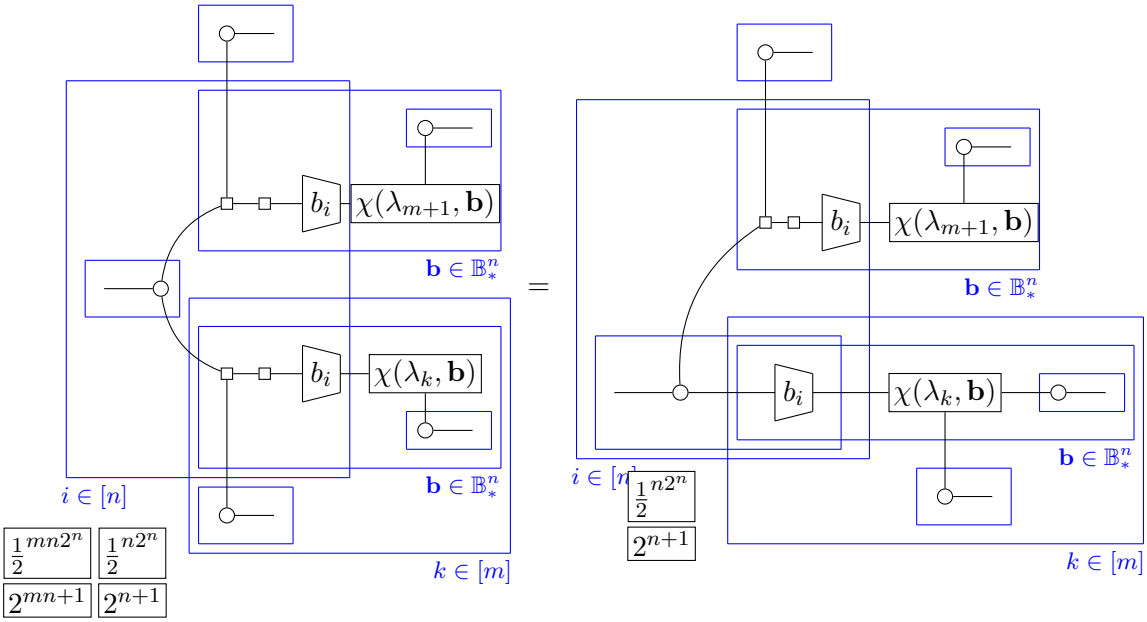
Lemma 9.1. We are going to prove the following by induction on n and m:



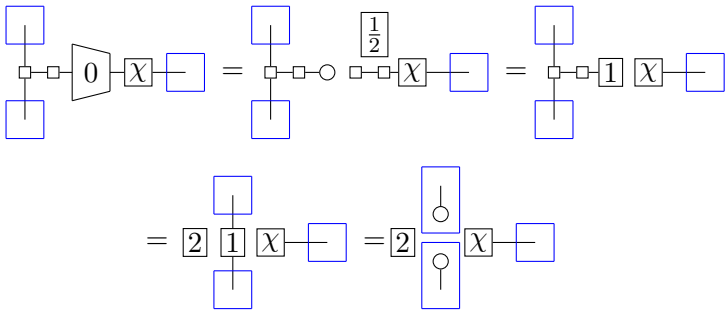
Proof. We start with an induction on m, by at first expending the $m+1$ term of the !-box:



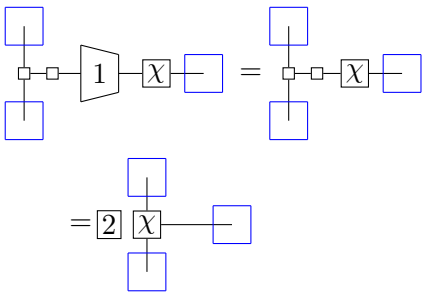
Then by induction hypothesis:



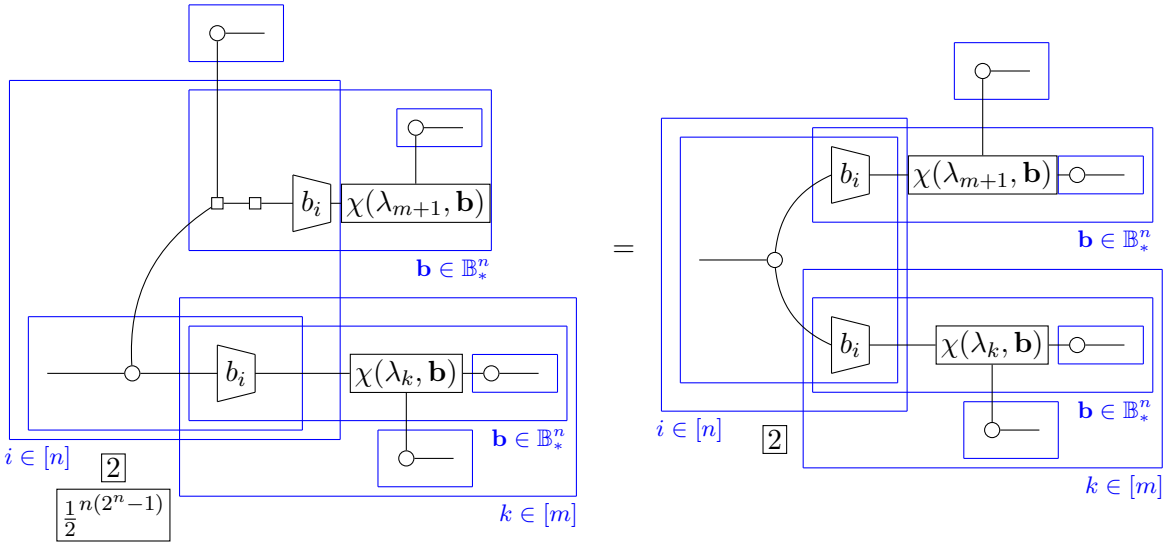
Before stating the next step, we detail the reasons why it is true:



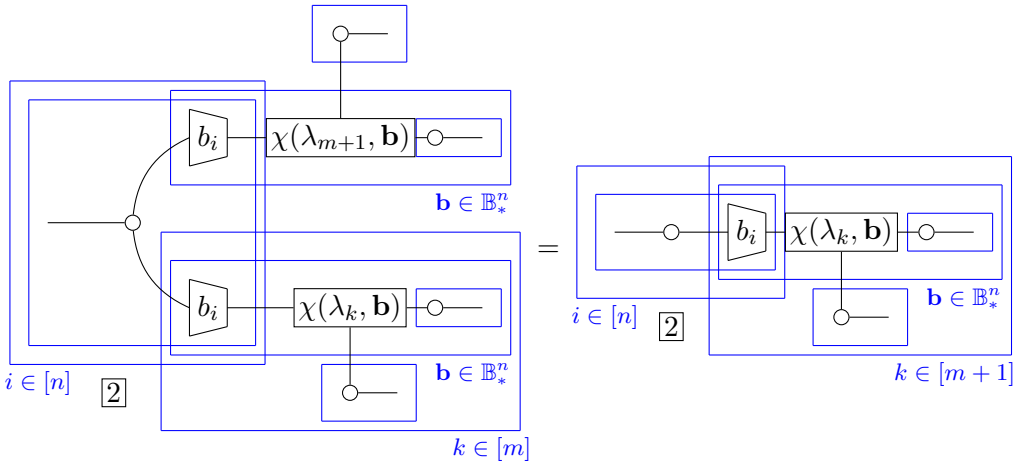
and also:



Thanks to the two previous equations:



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



Then, the proof carries on with the induction on n, the expansion of the Booleans $\mathbf{b}$ doubles what resides in the m-labelled !-box, which meanss that we need the induction hypothesis for the coupe $(n, 2m)$ – which is already proven since we have proven the induction over m. Thus the lemma holds. □