Bachelor thesis
Computing Science

Radboud University

# Investigating the minimality of the ZH-calculus

*Author:*
Thomas van Ouwerkerk
s4708105

*First supervisor/assessor:*
Aleks Kissinger
aleks.kissinger@cs.ox.ac.uk

*Second assessor:*
Freek Wiedijk
freek@cs.ru.nl

June 2019

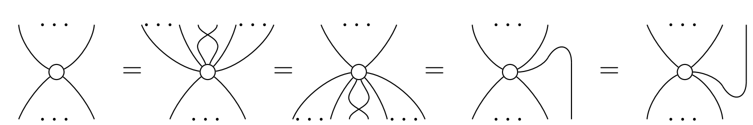
# 1 Introduction

## 1.1 History

In 1982 well-known physicist Richard Feynman suggested we might need to build a new kind of computer to accurately simulate all of physics. Specifically quantum-mechanics is a problem, as it is currently impossible to simulate the behaviour of quantum-mechanics efficiently on a classical computer. So we need a new device, something that exploits the properties of quantum mechanics: a quantum computer [7].
While physicists and engineers were thinking about how to build such a device, computing scientists and mathematicians put work into thinking about how to use a quantum computer once it would be built. In the following two decades a number of quantum algorithms were designed, e.g. [4][11][8]. These are algorithms that can run on a quantum computer to grant some improvement in solving problems which are hard to solve on classical computers. This improvement can come in the form of significant speedup in calculation, compared to classical computers, or giving new solutions with fewer drawbacks to existing problems.
In 2004 Abramsky and Coecke took a more abstract look at quantum mechanics and introduced a semantics using concepts from category theory [1]. This semantics was worked on in the following years and in 2008 Coecke and Duncan introduced the ZX-calculus [5]. Many papers have been written about this calculus and some alternatives have even been created, like the ZW-calculus[9] and the ZH-calculus[3]. The latter of these is the main topic of this thesis. Further, more detailed, reading on the history of the ZX-calculus can be found in chapter 1.3 of [6]. This introduction was only a very brief summary of that section.

## 1.2 Graphical calculi and ZX

The ZX-calculus is a graphical calculus. It is a language of string diagrams, used for reasoning about quantum mechanics and quantum information theory.[6].
A string diagram is a diagram consisting of a number of boxes, or other 2-dimensional shapes like circles or triangles, connected by lines (wires). The way to read such a diagram is from bottom to top. The wires dangling at the bottom are inputs and the free wires at the top are outputs. The shapes in between represent some operation on information on the wires. The most important rule here is: Only connectivity matters. This means that you can freely move and turn and twist boxes and wires.



For the ZX-calculus, the most important 'boxes' are the *Z-spider* and the *X-spider*. These are displayed as a white dot and a grey dot respectively.


The rules of a calculus like ZX can be used to rewrite diagrams, which is useful in circuit optimisation for example. If there exists some circuit that is designed to solve a certain problem one might ask whether it can be designed more efficiently. Computing scientists (or others who have experience in digital circuit design) will likely be familiar with De Morgan's laws, which can be used to rewrite and ultimately simplify classical electrical circuits. The ZX-calculus allows for the same thing in the case of quantum circuits.

## 1.3 Completeness and minimality

A graphical calculus, like ZX, will consist of some definitions and a number of rewriting rules. These rules can then be used to rewrite other diagrams and so create expressions which equate different diagrams to each other.
The thing that is desired of such a set of rules is *completeness*. Completeness is the following property for a set of axioms $X$ and diagrams $D_1$ and $D_2$ :

$$\forall D_1, D_2; [\![D_1]\!] = [\![D_2]\!] \rightarrow X \vdash D_1 = D_2$$

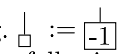
This means that if two diagrams have equal interpretations, they can be proven to be equal using the rules in $X$.

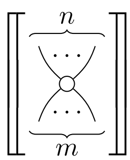
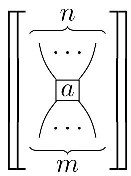
When a complete set of rules is found, the next challenge is to figure out whether that set is minimal. Finding a complete set of rules can be made trivial if you just assume *everything*, but the resulting calculus is not very interesting. It is a lot less trivial to find a complete rule set that is as small as possible.
Such a minimal set of rules is more powerful, as it needs to assume less to prove the same things. Additionally it makes automated proving easier, as provers need to consider fewer rules, which can potentially speed up computation. Finally, a minimal set of axioms can be seen as the most fundamental set of rules, which most closely relates to reality[10].

## 1.4 Outline

This thesis will investigate the minimality of the rule set of the ZH-calculus. First, the ZH-calculus itself will be introduced and explained. Secondly, the concept of independence will be introduced. Along with this the main technique used to find results is explained and a proof for its functioning is shown. Besides being able to show independence, this technique also allows for making statements about the necessity of rules. After the explanation, this technique is applied a number of times to find independence between subsets of the ZH-calculus' axioms. Finally all these results are collected and combined to give an overview of the (in)dependence relations between the axioms and a suggestion for future work will be made.

## 2 The ZH-calculus

In 2018 Backens and Kissinger first introduced the ZH-calculus [9]. In essence, this calculus is very similar to ZX. It is another graphical calculus, using string diagrams for diagrammatic reasoning about quantum mechanics. Like the ZX-calculus, ZH uses a white Z-spider as one of its generators. Unlike ZX, ZH has an ‘H-box’ as its second generator. This is an $n$-ary generalisation of the Hadamard gate, an operation used in quantum computing that is normally only defined to be used on a single qubit.
Diagrams in the ZH-calculus are constructed using the white Z-spiders, and H-boxes. The boxes are displayed as white squares, labelled by a complex number $a$. When an H-box has a label of $-1$, the label is left out and the box symbol is drawn smaller, e.g. 
These diagrams represent linear maps, which are interpreted in the following way, where $[\![\cdot]\!]$ is the interpretation mapping, mapping diagrams to linear maps.


$$:= |0\rangle^{\otimes n}\langle 0|^{\otimes m} + |1\rangle^{\otimes n}\langle 1|^{\otimes m}$$

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

The notation using vertical lines and angle brackets is called bra-ket notation, also known as Dirac notation. $\langle a|$ is called a *bra*, $|b\rangle$ is called a *ket* and $\langle a|b\rangle$ is a *braket*. It is used here to represent the following vectors:

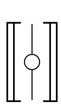
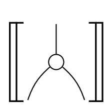
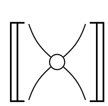
| | |
|---|---|
| $\langle 0\|$ | $\begin{pmatrix}1 & 0\end{pmatrix}$ |
| $\|0\rangle$ | $\begin{pmatrix}1\\0\end{pmatrix}$ |
| $\langle 1\|$ | $\begin{pmatrix}0 & 1\end{pmatrix}$ |
| $\|1\rangle$ | $\begin{pmatrix}0\\1\end{pmatrix}$ |

The $\otimes$ symbol is used to denote the Tensor-product, which is used here as a kind of matrix multiplication, where you take the elements of the matrix on the left one by one and multiply them by the entire matrix on the right every time. A new matrix is then constructed using these scalar multiplied matrices by placing them in the same way the original elements of the matrix on the left were ordered, e.g.

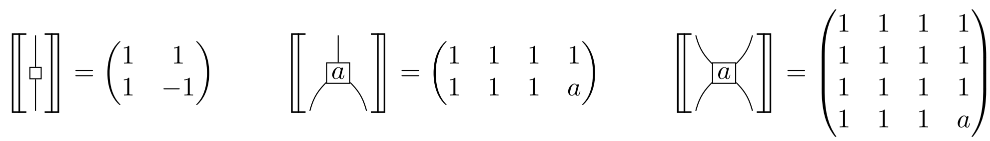
$$\mathrm{V}\otimes\mathbf{W} = \begin{pmatrix}v_1 & v_2\\ v_3 & v_4\end{pmatrix}\otimes\begin{pmatrix}w_1 & w_2\\ w_3 & w_4\end{pmatrix} = \begin{pmatrix} v_1 * \begin{pmatrix}w_1 & w_2\\ w_3 & w_4\end{pmatrix} & v_2 * \begin{pmatrix}w_1 & w_2\\ w_3 & w_4\end{pmatrix} \\ v_3 * \begin{pmatrix}w_1 & w_2\\ w_3 & w_4\end{pmatrix} & v_4 * \begin{pmatrix}w_1 & w_2\\ w_3 & w_4\end{pmatrix}\end{pmatrix} = \begin{pmatrix} v_1w_1 & v_1w_2 & v_2w_1 & v_2w_2\\ v_1w_3 & v_1w_4 & v_2w_3 & v_2w_4\\ v_3w_1 & v_3w_2 & v_4w_1 & v_4w_2\\ v_3w_3 & v_3w_4 & v_4w_3 & v_4w_4\end{pmatrix}$$

The superscript tensor product notation used in the definition of Z-spiders is similar to the general power-notation most people are familiar with, but using the tensor-product as its repeating operation, instead of a regular product. e.g. $|0\rangle^{\otimes 3} = |0\rangle\otimes|0\rangle\otimes|0\rangle$
Having a bra or ket with multiple numbers in it, e.g. $\langle a_1a_2|$, is also a shorthand for the tensor product of individual bras or kets, so $\langle a_1a_2| = \langle a_1|\otimes\langle a_2|$

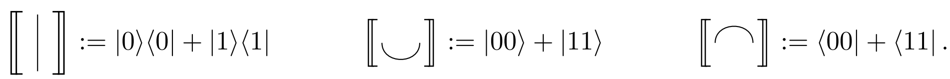
With these definitions, the general structure of generators' matrices can be described as follows: *Z-spiders* have a matrix interpretation which is a matrix with a one in the top-left and bottom-right corners and zeroes as the rest of its entries, a height of $2^n$, and a width of $2^m$ , e.g.


$$= \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$$

$$= \begin{pmatrix}1 & 0 & 0 & 0\\0 & 0 & 0 & 1\end{pmatrix}$$

$$= \begin{pmatrix}1 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 1\end{pmatrix}$$

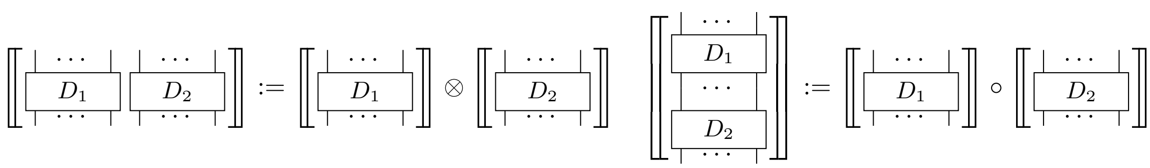
*H-boxes* have a matrix interpretation which is a matrix with its label in the bottom right and ones as all its other entries. Just like the *Z-spiders* they have a height of $2^n$ and a width of $2^m$, e.g.



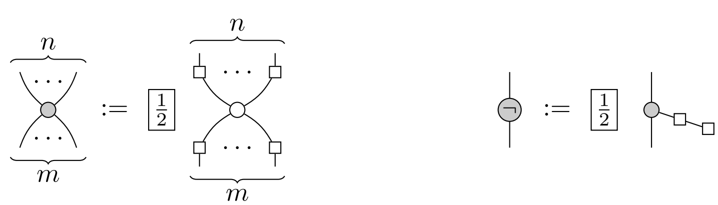
Straight and curved wires are defined as follows:



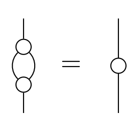
When two diagrams are placed next to each other, the associated linear map is the tensor product of the two separate linear maps. Two diagrams can also be sequentially composed, connecting their wires, which is associated with the matrix product of the associated matrices.



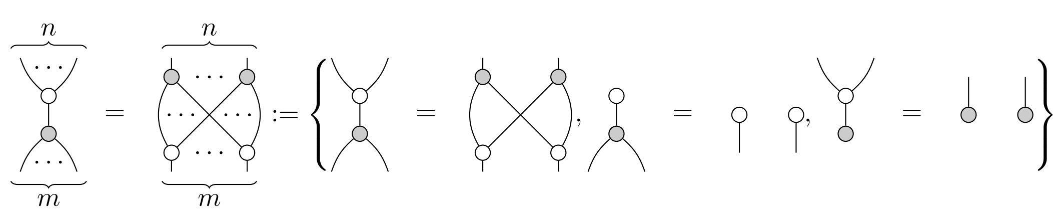
In order to make some diagrams easier to read, derived generators are introduced. These are grey spiders, also known as *X-spiders*, and *NOT* respectively.



The rules of the ZH-calculus, as they were originally presented, are shown in figure 1. Since the publishing the following equality has been proved to be derivable from the other rules, as can be found in [12]:



As can be seen, some of the presented rules include a scalar multiplication of two. This is necessary to make the presented equality true. In the context of this language, it is necessary to present proper rules. But in the context of quantum computation, these scalar multiplications are not that important. Since all measurements that can be done in the context of quantum computing give us results up to a non-zero scalar, it is fine to show equality up to a non-zero scalar.
The bialgebra rules BA1 and BA2, while presented as such, are not actually two distinct rules, but rather two families of rules. BA1 and BA2, as described here, show what all members of these families look like. However, they both follow from three different finite rules respectively, when taking spider laws into account. The proof for this and further explanation on the concept can be found in [6] (section 9.3) covering *strong complementarity.*



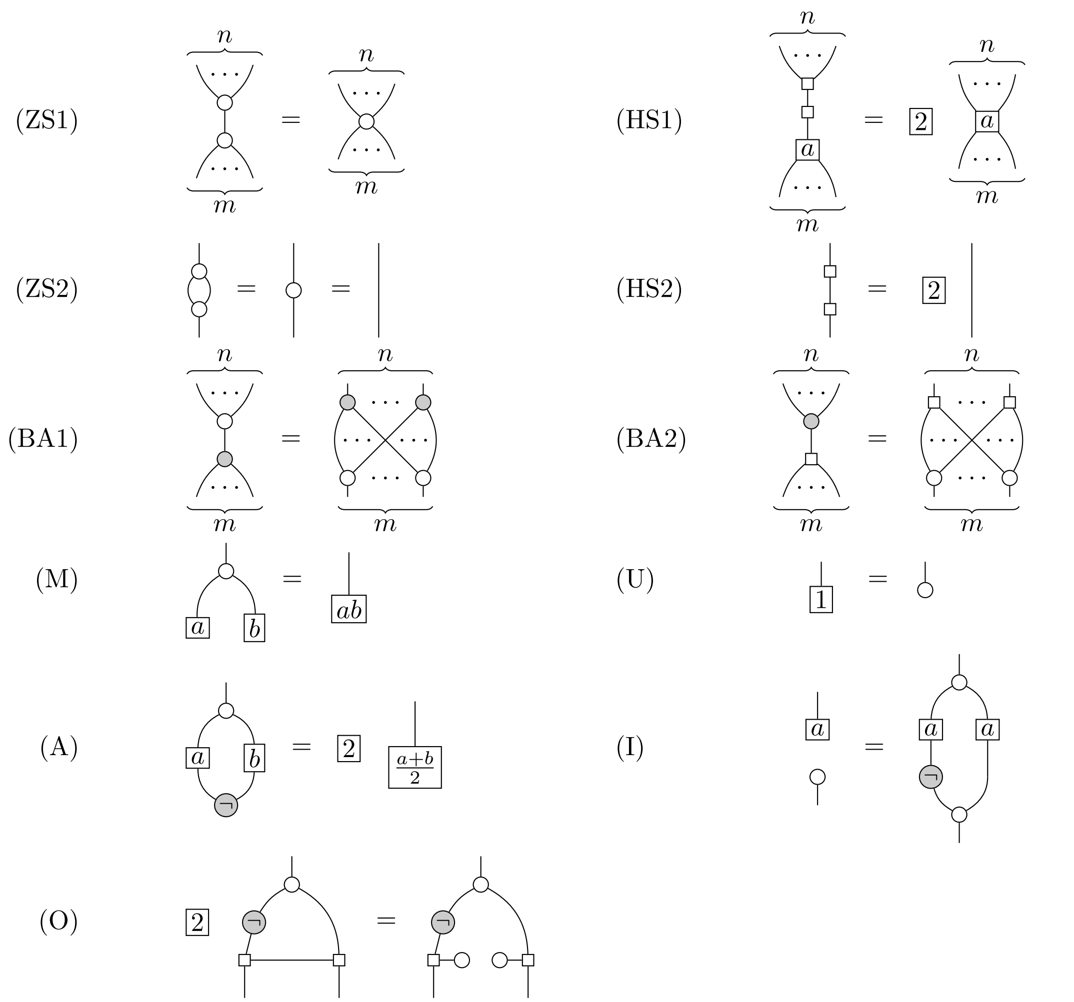
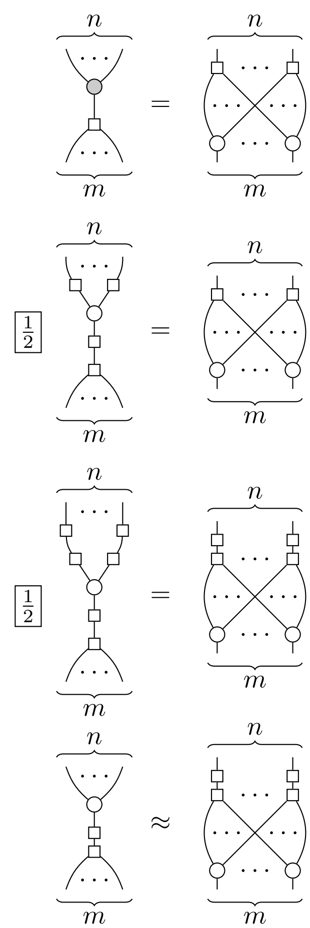
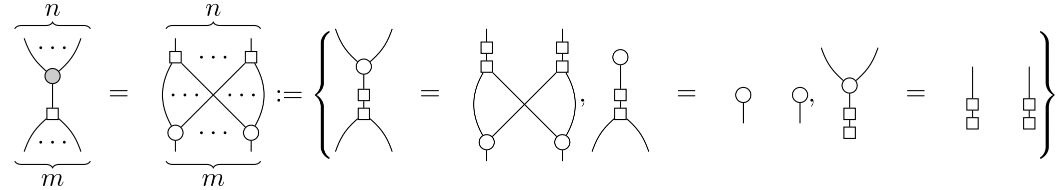



Figure 1: The original rules of the ZH-calculus as defined in [3]. $m, n$ are nonnegative integers and $a, b$ are arbitrary complex numbers. The right-hand sides of both bialgebra rules (BA1) and (BA2) are complete bipartite graphs on $(m+n)$ vertices, with an additional input or output for each vertex. The horizontal edges in equation (O) are well-defined because only the topology matters and we do not need to distinguish between inputs and outputs of generators. n.b. the rules (M), (A), (U), (I), and (O) are pronounced *multiply*, *average*, *unit*, *intro*, and *ortho*, respectively.

The rules composing BA1, as can be seen, are the special cases $(n = 2, m = 2)$, $(n = 0, m = 2)$, $(n = 2, m = 0)$. BA2 cannot be expanded in the same way directly, but it can be rewritten to follow the same pattern as BA1, after which its corresponding three finite rules can easily be found:




The steps followed are, in order, applying the definition of the X-spider, adding a row of $n$ boxes to each side and then applying HS2 to get rid of the boxes on the left-hand side. In this last step a scalar of $2(n-1)$ appears, which is left out. Using this alternative shape of the rule, the following can be defined:



# 3 Independence

Independence is the notion that an axiom $A$ in a rule set $P$ does not follow from other axioms in that rule set, i.e. $A$ cannot be proven using the rules in $P - A$.
This definition can also be expanded to subsets of $P$. If one was to split $P$ into the non-empty subsets $Q \subset P$ and $P - Q$, $Q$ can be said to be independent from $P$ if no axiom in $Q$ can be proven from the rules in $P - Q$.
Independence can be used in the context of the search for a minimal set of axioms. Say there is some complete set of rules $P$, but it is not certain whether some $A \in P$ can be removed from $P$ while still having a set of rules that is complete. If it is known $A$ is independent from the rest of the rules, removing it would cause $A$ to no longer be provable, which would mean $P$ is no longer complete. So if some $A$ is independent from the other rules, it is apparently a necessary part of the minimal rule set.
If a non-empty subset $Q \subset P$ is independent from $P - Q$, at least one, but maybe more than one, rule must be added to $P - Q$ to be able to prove the rules in $Q$. It can be concluded that at least one of the rules in $Q$ is necessary to create a complete minimal set of rules. This may not be clear immediately, but should be once shown in the proof in the next section.

## 3.1 Showing independence

Simply put, we can show independence of some subset of axioms by changing the interpretation mapping $[\![.]\!]$ and looking for the axioms that no longer hold. These are independent from the rest. The proof for this is covered in [10], but because of its importance to this thesis it is covered below as well, in more detail. The proof uses some concepts from a branch of mathematics called category theory, but only a few basic ones, which are explained here first.
*Categories* are used as abstractions to other mathematical concepts and can be seen as labelled directed graphs. We will call the nodes of these graphs 'objects' and the directed edges 'arrows'. An example of a category is the category of sets, where sets are the objects and functions between sets are the arrows. The ZH-calculus is a category as well, where diagrams are arrows going from inputs to outputs.
A *functor* is a mapping between categories. It maps all objects to objects and arrows to arrows. So if we take diagrams in the ZH-calculus and count the number of H-boxes, this is a functor that maps diagrams to numbers and the objects to some other arbitrary object. If we interpret the diagrams in the ZH-calculus and get their corresponding matrices, this is a functor that maps the diagrams to matrices of complex numbers [2].
With this bit of theory, it should be possible to understand the following definitions and accompanying proof.

**Definition 3.1.** *An axiom $A$ is independent from a set of axioms $ZH$ if $ZH - A \nvdash A$. A non-empty subset of axioms $S \subseteq ZH$ is independent from $ZH$ if $\forall A \in S, A$ is independent from $ZH$.*

**Definition 3.2.** *Given a functor $[\![.]\!]' : ZH \to \mathcal{C}$, an axiom $A := A_l = A_r$ is satisfied if $\mathcal{C} \vdash [\![A_l]\!]' = [\![A_r]\!]'$. A subset of axioms $S \subseteq ZH$ is satisfied if $\forall A \in S, A$ is satisfied.*

**Definition 3.3.** *An axiom $A \in ZH$ is necessary if $ZH - A$ is incomplete.*

**Lemma 3.1.** *Given a functor $[\![.]\!]' : ZH \to \mathcal{C}$, if a non-empty subset $S \subseteq ZH$ is the set of axioms not satisfied by $[\![.]\!]'$, $S$ is independent from $ZH$.*

*Proof.* Let $S$ be the axioms not satisfied by $[\![.]\!]'$, i.e.

$$\forall A \in S, \mathcal{C} \nvdash [\![A]\!]' \tag{1}$$

Suppose $S$ is not independent from $ZH$, i.e.

$$\exists A \in S, ZH - S \vdash A \tag{2}$$

From (2) follows there exists some finite proof $P$, that proves $A$ using the rules in $ZH - S$ and from (1) follows all of the steps in this proof are satisfied by $[\![.]\!]'$
Using $P$, an alternate proof $P'$ can be constructed in $\mathcal{C}$, where every step $D_1 = D_2$ is substituted by $[\![D_1]\!]' = [\![D_2]\!]'$. Such a substitution is guaranteed to be possible, since all steps are satisfied. This proof shows the following:

$$\mathcal{C} \vdash [\![A]\!]' \tag{3}$$

(1) and (3) contradict each other, thus the original assumption, (2), must be false and $S$ is independent from $ZH$. □

**Lemma 3.2.** *If a non-empty subset $S \subseteq ZH$ is independent from $ZH$, at least one axiom in $S$ is necessary.*

*Proof.* Suppose the axioms in $S$ are not necessary, i.e.

$$ZH - S \text{ is complete.} \tag{4}$$

From 4 follows:

$$\forall A \in S, ZH - S \vdash A \tag{5}$$

(5) is in contradiction with our definition, since $S$ is independent from $ZH$. This means the original assumption was false and it must be the case at least one axiom in $S$ is necessary.

□

Using this, the goal is no longer to directly find independence, but to choose categories and functors. Similarly to [10], the following categories will be considered:

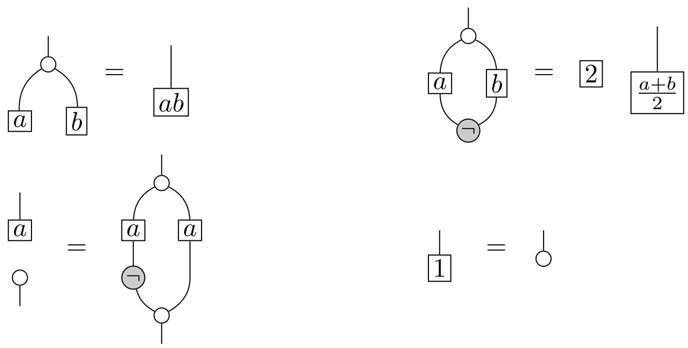
1. Booleans. This boils down to choosing some property about diagrams and determining whether this property is retained when applying the axiom. One example of such a property could be: 'The diagram contains boxes'. If this is true for both sides or false for both sides of an axiom equality, it satisfies the functor. The actual category will be a single object, with two arrows to itself, one labeled 'true' and one labeled 'false'.

2. Infinite sets, specifically using the set of complex numbers $\mathbb{C}$. This set is used to apply operators to the labels of H-boxes that some of the diagrams contain. For example adding them together, multiplying, or some other operation, that yields a complex number. Similarly to the Booleans, this category will also have a single object, but with more arrows pointing to itself, one labeled for each complex number.

3. ZH. Alternate interpretations for diagrams can be given. As defined earlier in this thesis, the ZH-calculus has two generators, with their own matrix interpretations. These interpretations can be chosen to be different, causing some axioms to hold under interpretation, while others break. The standard interpretation functor maps the arrows of diagrams to arrows that are matrices of complex numbers. This mapping is maintained, but the matrices at the arrows are changed.

Some of the ZH axioms are visually rather similar. The bialgebra rules for example:

or the patterns found among M, A, I and U.



These similarities could suggest that one of them might follow from others. A particularly interesting result, would be finding independence between rules which are very similar, indicating they are more unique than they seem at first glance.

## 3.2 Diagrammatic properties

A similar question to the one being asked in this thesis, has been covered before in a different paper [12]. It defines an alternate version of the ZH-calculus, where phases are removed from consideration and a similar but different set of axioms is defined.
In a part of this paper, the authors have investigated the minimality of their new set of axioms. The authors construct a number of graph-theoretic arguments, which can also be applied to the regular ZH-calulus. This lead to the functors $[[.]]_{ew}$, $[[\cdot]]_{ed}$, $[[\cdot]]_{Z4}$,and $[[.]]_{H4}$. These functors map the category of diagrams to a category $\mathcal{B}$ containing a single object, with two arrows 'true' and 'false'. All objects in the ZH category are mapped to that single object, while every diagram is mapped to either the 'true' or 'false' arrow. In the new category sequential composition and parallel composition both need to be defined, but these operations will depend on the chosen properties, so they are shown along with the functor.

### 3.2.1 Empty wires

When looking at the rules of the ZH-calculus, there are only two rules which can change an empty wire into a non-empty one. So a functor can be created

$$[[.]]_{ew} : ZH \to \mathcal{B}$$

which maps diagrams to the corresponding Boolean value using the property 'ignoring generators without inputs or outputs, the diagram consists only of empty wires'. Here parallel and sequential composition,respectively $\otimes$ and $\circ$, are defined as the logic AND: $\wedge$. Only if both diagrams consist only of empty wires, the compositions of them do as well.
The only rules that do not satisfy this functor are ZS2 and HS2. All of the other rules do not involve empty wires on either side of the equality, meaning both sides of each rule get mapped to 'false' and the rule will satisfy the functor. ZS2 and HS2 both have an empty wire on one side of their equations, but not the other, meaning one side of each equation is mapped to 'true' by the new functor, while the other side is $False$.

### 3.2.2 Empty diagrams

Similarly to looking at empty wires, the empty diagram can be considered as well. there are two rules which can have an empty diagram on one side. The functor

$$[[\cdot]]_{ed} : ZH \to \mathcal{B}$$

| **Satisfy** $[[.]]_{ew}$ | **Do not satisfy** $[[.]]_{ew}$ |
|---|---|
| ZS1 | ZS2 |
| HS1 | HS2 |
| BA1 | |
| BA2 | |
| M | |
| A | |
| U | |
| I | |
| O | |

Table 1: Results investigation satisfying the functor $[[.]]_{ew}$

maps diagrams to the corresponding Boolean value using the property 'the diagram is the empty diagram'. For this functor both types of composition need to be defined as the logic AND, as a composed diagram will only be the empty diagram if both the diagrams composing it are also empty. So $\circ = \otimes = \wedge$. The functor will show another independence. It is satisfied by all rules besides BA1 and BA2.

BA1 and BA2, taking $m = 0$ and $n = 0$, will both have an empty diagram on the right-hand side of their equations. The other rules do not involve empty diagrams, as they all have at least a single wire or generator on each side of their equations.

| **Satisfy** $[[.]]_{ed}$ | **Do not satisfy** $[[.]]_{ed}$ |
|---|---|
| ZS1 | BA1 |
| ZS2 | BA2 |
| HS1 | |
| HS2 | |
| M | |
| A | |
| U | |
| I | |
| O | |

Table 2: Results investigation satisfying the functor $[[.]]_{ed}$

#### 3.2.3 Z-spiders with a degree $\geq 4$

Another one of the properties presented in [12], is the following: ZS1 is the only rule that shows equality between a diagram with a Z-spider of a degree less than 4 and a diagram with a Z-spider of a degree greater than or equal to 4. This property can be rephrased to define another functor:

$$[[.]]_{Z4} : ZH \to \mathcal{B}$$

which evaluates diagrams using the statement 'This diagrams contains a Z-spider with a degree of 4 or greater'. Both types of composition are defined in the new category as the logic OR operation, since neither will influence the degree of generators and either composing diagram needs to contain one to be true. This means $\circ = \otimes = \vee$

For most of the rules, it is easy to see this functor will evaluate to 'false' on both sides of the equality, as these do not contain any Z-spiders of a degree of 4 or greater. Since BA1 and BA2 are defined by three distinct cases, there only needs to be consideration of rules up to $n = 2$ and $m = 2$, as all other possibilities are covered by these under induction. For all of these cases the maximum degree of Z-spiders on either side of their equality is 3, since the maximum degree of

Z-spiders is $1+n$ or $1+m$. This means both sides of each equality will evaluate to 'false' and the rules satisfy the functor.
Finally, the case of ZS1 where $(n=2, m=2)$, will evaluate to 'false' on the left-hand side, as both Z-spiders have a degree of 3, but the right-hand side will evaluate to 'true', as the degree of the Z-spider there is 4. This means ZS1 is the only rule that does not satisfy this functor.

| **Satisfy** $[[.]]_{Z4}$ | **Do not satisfy** $[[.]]_{Z4}$ |
|---|---|
| ZS2 | ZS1 |
| HS1 | |
| HS2 | |
| BA1 | |
| BA2 | |
| M | |
| A | |
| U | |
| I | |
| O | |

Table 3: Results investigation satisfying the functor $[[.]]_{Z4}$

### 3.2.4 H-boxes with a degree $\geq 4$

Following the same reasoning as $[[.]]_{Z4}$, [12] shows that HS1 is the only rule that relates H-boxes with a degree of 4 or greater to ones below that.
Defining this is a functor gives:

$$[[.]]_{H4} : ZH \to \mathcal{B}$$

Which evaluates to 'true' iff a diagram contains an H-box with a degree of 4 or greater. Just like the $[[.]]_{H4}$ functor, sequential and parallel composition are defined as OR: $\circ = \otimes = \vee$
Again, it is easy to see most rules do not contain any H-boxes of this degree, so it follows quickly that these rules satisfy the functor. The only ones that really need to be considered in more detail are HS1 and BA2. At first glance, BA1 might also still be a candidate for not satisfying this functor, since it can fan out and might create some boxes with high degree. But at closer inspection, applying the definition of the X-spider will only create H-boxes with a degree of 2, so both sides of this rule evaluate to 'false' as well.
When considering BA2, the rules up to $n = 2$ and $m = 2$ need to be considered, similarly to the investigation of $[[.]]_{Z4}$. All of these rules do not contain any H-boxes with a degree of 4 or greater. So it can be concluded they all satisfy the functor and by that BA2 does too. HS1, just like ZS1 with $[[.]]_{Z4}$, with the case $(n=2, m=2)$, does not satisfy $[[.]]_{H4}$, as the left side will evaluate to 'false', while the right-hand side will evaluate to 'true'. This means HS1 is the only rule that does not satisfy this functor.

### 3.2.5 H-box parity

For this functor to work, the diagrams first need to change slightly. Under the standard interpretation, a single white dot with no connecting wires equals 2:

$$[\bigcirc] = |0\rangle^{\otimes 0} \langle 0|^{\otimes 0} + |1\rangle^{\otimes 0} \langle 1|^{\otimes 0} = 1 * 1 + 1 * 1 = 2$$

This means we can freely swap out the h-box scalar $\boxed{2}$ for a white dot.

With these alternate diagrams, where the scalars are now white dots instead of boxes, the following functor becomes interesting:

$$[[.]]_{ParityH} : ZH \to \mathcal{B}$$

| **Satisfy** $[[.]]_{H4}$ | **Do not satisfy** $[[.]]_{H4}$ |
|---|---|
| ZS1 | HS1 |
| ZS2 | |
| HS2 | |
| BA1 | |
| BA2 | |
| M | |
| A | |
| U | |
| I | |
| O | |

Table 4: Results investigation satisfying the functor $[[.]]_{H4}$

This functor will evaluate to 'true' if, and only if, a given diagram contains an even amount of boxes. Effectively, this functor now takes the standard diagrams, swaps out the scalar box for a white dot and then checks the parity of the amount of H-boxes. This means that if two diagrams which evaluate to 'true' are composed or two diagrams which evaluate to 'false' are composed, the resulting diagram should also evaluate to 'true'. This corresponds to the logic XNOR operation. So both types of composition are defined in the new category as XNOR: $\circ = \otimes = \odot$.
When investigating the rules for this property, the following things can be taken into consideration:

- NOT contains five H-boxes in its definition. Therefore it does need to be counted as it will change the parity.
- The X-spider contains, by its definition, an H-box for every wire connected to it, so its degree can be used in counting.

Looking at the rules, ZS1 and ZS2 satisfy the functor, as they both have an even amount of boxes on both sides of the rule: none. Not taking the scalar in consideration, HS1 has odd amounts on both sides and HS2 even amounts, so these satisfy the functor as well. M, U, A, I and O are also quite easily counted by hand. O satisfies the functor, which can be seen at a glance because the only difference is the splitting of a wire and addition of two Z-spiders. M and U can also be counted at a glance, two and one, and one and zero, from left to right respectively. So M and U do not satisfy the functor. Counting the NOT as five boxes, A and I both have seven boxes on one side and a single one on the other. Both odd numbers, so these both satisfy the functor.
BA1 and BA2 need to be considered a bit more carefully. First BA1, which we will find does not satisfy the functor. All of its composing rules do not in fact. The first has an X-spider of degree three on the left-hand side, while having two of those on the right-hand side. This means both sides of the equation do not have the same parity of H-boxes, odd and even respectively, and the equation does not satisfy. The other two rules also do not satisfy the functor, as they contain 3 and 1 boxes respectively on the left, while having 0 and 2 respectively on the right.
Secondly, the decomposing rules of BA2. These rules follow a similar pattern to BA1. But the position of the X-spider is replaced by two H-boxes. This means that the parity of the amount of H-boxes actually does not change, as the amount only changes in steps of two, which does not affect the parity.
This result tells us the bialgebra rules are not necessarily directly dependent on each other.

## 3.3 Infinite sets

### 3.3.1 Multiplying labels

$$[[.]]_{MultH} : ZH \to \mathcal{M}$$

Just like the functor considering the parity of the amount of H-boxes, this functor also takes two steps. First the floating H-boxes with a label of two are again replaced by Z-spiders without

| **Satisfy** $[[.]]_{ParityH}$ | **Do not satisfy** $[[.]]_{ParityH}$ |
|---|---|
| ZS1 | BA1 |
| ZS2 | M |
| HS1 | U |
| HS2 | |
| BA2 | |
| A | |
| I | |
| O | |

Table 5: Results investigation satisfying the functor $[[.]]_{ParityH}$

inputs or outputs. Then the labels of all H-boxes in the diagram are multiplied. The category $\mathcal{M}$ is defined by a single object with an arrow for each possible complex number. All diagrams are mapped by $[[.]]_{MultH}$ to the arrow with a label equal to their multiplied labels. The composition operators are both defined by multiplication, i.e. $\circ = \otimes = \times$.

ZS1 and ZS2 do not contain any H-boxes, so they automatically satisfy the functor. HS1 gets a left-hand side of $-1 * -1 * a$ and a right-hand side of $a$. These are equal to each other so HS1 satisfies the functor. HS2 gets $-1 * -1$ on the left-hand side and 1 on the right-hand side, so this rule also satisfies the functor.

BA1 and BA2 only contain boxes with labels of $-1$. This means if the amount of boxes on either side of either equation is even, the result is 1, while otherwise the result is $-1$. From this can be concluded the rules satisfy the functor if and only if they satisfy the functor $[[.]]_{ParityH}$. This means BA1 does not satisfy $[[.]]_{MultH}$ while BA2 does.

M, U, and O also satisfy the functor. This can be quickly checked, as it can be seen quickly that $a * b = ab$ and $1 = 1$, and O is symmetric in the amount and labels of H-boxes. A and I do not satisfy $[[.]]_{MultH}$, as for A $-1 * a * b \neq \frac{a+b}{2}$, and for I $a \neq -1 * a * a$. In summary: BA1, A, and I do not satisfy $[[.]]_{MultH}$, as displayed in table 6.

| **Satisfy** $[[.]]_{MultH}$ | **Do not satisfy** $[[.]]_{MultH}$ |
|---|---|
| ZS1 | BA1 |
| ZS2 | A |
| HS1 | I |
| HS2 | |
| BA2 | |
| M | |
| U | |
| O | |

Table 6: Results investigation satisfying the functor $[[.]]_{MultH}$

## 3.4 Alternate $ZH$ interpretation models

### 3.4.1 H-box becomes a W-gate

A possible alternate generator interpretation is:

$$[[.]]_W : ZH \to ZH$$

where the H-box's interpretation becomes that of the W-state as defined in [9]. The standard interpretation of diagrams takes the category and maps the diagrams to a category $\mathcal{C}$, where arrows are labelled by matrices of complex numbers. This altered interpretation is the same, only the matrices that label the arrows are changed to fit the W-spider definition. The compositional rules remain unchanged.

To represent this matrix, label all rows and columns with the binary representation of their number, starting with zero, e.g. the third row and column are both labelled `010`. Then place a one in column 0 for every row where the binary label contains exactly a single one and place a one in row 0 for every column with a binary label that contains exactly a single one. Examples can be seen in figure 2. In this model placing a complex number $a$ in the H-box has no effect on its corresponding matrix.

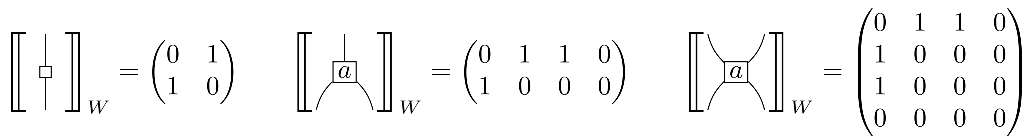
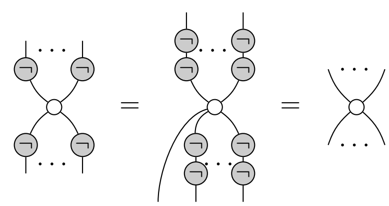


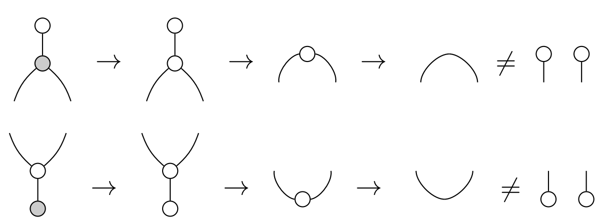
Figure 2: Example W-gate matrices

A W-state with one input and one output gives a matrix which is equal to the NOT operation. From this follows that grey spiders become white spiders surrounded by NOTs. NOT copies through the white spiders, as proven in [3](Lemma 3.2). For every grey spider, one of the connected NOTs can copy through, making the wire empty, while adding a NOT on every other wire connected to the spider. Since NOT is its own inverse, the NOTs on all wires will cancel out and leave just a white spider.

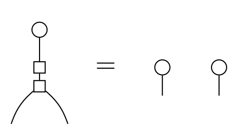


We now consider each of the ZH equational rules and determine which still hold up to a scalar and which no longer do.

Since only the H-box is altered, spider fusion for white spiders is unchanged and ZS1 and ZS2 still hold as they did before, so they satisfy our functor.
The H-boxes are now W-states. The properties described by HS1 and HS2 follow from the definition of this W-state, as can be found in [9].
Looking at BA1, we can turn the grey spiders into white ones, as stated before. This creates a problem. When looking at the second and third rule of its definition, we see that the left sides will fuse into a cap and a cup respectively, while having disconnected nodes on both right sides.



In both cases, calculating the matrix of the separated nodes will yield a different matrix than the one of the cap and cup.
For BA2, it suffices to investigate the rule



Calculating the matrices of the left-hand side and right-hand side yields two different vectors which are not scalar multiples of each other, thus BA2 does not satisfy the functor.
The remaining rules are finite and their matrix equalities can be checked by calculating them. This shows that multiply and average still hold, while unit, intro and ortho break.

This leaves us with the first division of independent axioms:

| **Satisfy** $[[.]]_W$ | **Do not satisfy** $[[.]]_W$ |
|---|---|
| ZS1 | BA1 |
| ZS2 | BA2 |
| HS1 | U |
| HS2 | I |
| M | O |
| A | |

Table 7: Results of changing H-box to W-gate

# 4 Results

This thesis has given an explanation on graphical calculi and the ZH-calculus. The concept of independence was explained. It was shown that the independence of a subset of axioms can be shown and that from this follows one of those axioms is necessary for the minimal set of axioms. After this a number of functors were presented and it was investigated which axioms would and would not satisfy these functors. These functors can be viewed in table 8.

The functors defined in this thesis give a number of restrictions on the possible minimal subset. Since there are 11 axioms, there are $2^{11} = 2048$ possible ways of using or not using each of the separate axioms.
Looking at the 'Does not satisfy' column in table 8, 7 disjunctions can be made of rules that should be included. Running an exhaustive search, it was determined there are 246 possible combinations of axioms which satisfy these restrictions, i.e. 246 subsets of the axioms of the ZH-calculus contain at least one of each rule of each disjunction. It should be noted that not all of the functors are necessary to come to this conclusion. Since the set of possibly necessary rules associated with $[[.]]_{ed}$ is a proper subset of the one associated with $[[.]]_W$, the latter of these two does not actually add to restricting the amount of possible subsets of rules. Quantitatively this means $[[.]]_W$ is not an interesting result, but it still adds to this thesis by showing an example of a possible alternative interpretation and can act as inspiration for future work.
Besides these quantitative results, the specific subsets that were found also show some more interesting qualities. BA1 and BA2, while visually similar, are in fact independent. And the rules ZS1 and HS1 are necessary. These last two are the most restrictive type of results, which is good as it greatly decreases the amount of possible minimal subsets. It also indicates that any functors that are added to these results, should at least satisfy these rules.

| **Functor** | **Satisfies** | **Does not satisfy** |
|---|---|---|
| $[[.]]_{ew}$ | ZS1, HS1, BA1, BA2, M, A, U, I, O | ZS2, HS2 |
| $[[.]]_{ed}$ | ZS1, ZS2, HS1, HS2, M, A, U, I, O | BA1, BA2 |
| $[[.]]_{Z4}$ | ZS2, HS1, HS2, BA1, BA2, M, A, U, I, O | ZS1 |
| $[[.]]_{H4}$ | ZS1, ZS2, HS2, BA1, BA2, M, A, U, I, O | HS1 |
| $[[.]]_{parityH}$ | ZS1, ZS2, HS1, HS2, BA2, A, I, O | BA1, M, U |
| $[[.]]_{MultH}$ | ZS1, ZS2, HS1, HS2, BA2, M, U, O | BA1, A, I |
| $[[.]]_W$ | ZS1, ZS2, HS1, HS2, M, A | BA1, BA2, U, I, O |

Table 8: The functors presented throughout this thesis and the gathered results.

# 5 Future work

The following additions and improvements can be made on the results provided by this thesis:

- A limited amount of functors was investigated, trying other functors might yield additional interesting results to narrow down the amount of possible minimal subsets. Specifically there could be more investigation into functors mapping diagrams to other diagrams or to the infinite sets of numbers, e.g. by applying arithmetic on phases or counting the amount of occurrences of some generator.
- This thesis has investigated the relations between different axioms, but has not shown any distinctive proofs of rules *not* being necessary. A great result would be finding a proof that a certain rule is not necessary, for example by showing how this rule follows from others.
- Like this thesis made use of minimality results found regarding the phase-free ZH-calculus, the knowledge gained in investigating the minimality of this specific graphical calculus, could be applied to other calculi.

## References


[1] Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, pages 415–425. IEEE, 2004.

[2] Steve Awodey. *Category theory*. Oxford University Press, 2010.

[3] Miriam Backens and Aleks Kissinger. Zh: A complete graphical calculus for quantum computations involving classical non-linearity. *arXiv preprint arXiv:1805.02175*, 2018.

[4] Charles H Bennett and Gilles Brassard. Quantum cryptography: public key distribution and coin tossing. *Theor. Comput. Sci.*, 560(12):7–11, 2014.

[5] Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, 2011.

[6] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017.

[7] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, Jun 1982.

[8] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325–328, Jul 1997.

[9] Amar Hadzihasanovic. A diagrammatic axiomatisation for qubit entanglement. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 573–584. IEEE, 2015.

[10] Borun Shi. Towards minimality of clifford + t zx-calculus. 2018.

[11] Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[12] John van de Wetering and Sal Wolffs. Completeness of the phase-free zh-calculus. *arXiv preprint arXiv:1904.07545*, 2019.