

## Completeness of the ZH-calculus

Backens, Miriam; Kissinger, Aleks; Miller-Bakewell, Hector; van de Wetering, John; Wolffs, Sal

DOI:

[10.32408/compositionality-5-5](https://doi.org/10.32408/compositionality-5-5)

License:

Creative Commons: Attribution (CC BY)

*Document Version*

Publisher's PDF, also known as Version of record

*Citation for published version (Harvard):*

Backens, M, Kissinger, A, Miller-Bakewell, H, van de Wetering, J & Wolffs, S 2023, 'Completeness of the ZH-calculus', *Compositionality*, vol. 5, no. 5. <https://doi.org/10.32408/compositionality-5-5>

[Link to publication on Research at Birmingham portal](#)

### General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

### Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact [UBIRA@lists.bham.ac.uk](mailto:UBIRA@lists.bham.ac.uk) providing details and we will remove access to the work immediately and investigate.

# Completeness of the ZH-calculus

Miriam Backens<sup>1</sup>, Aleks Kissinger<sup>2</sup>, Hector Miller-Bakewell<sup>2</sup>, John van de Wetering<sup>3,2</sup>, and Sal Wolfs<sup>3</sup>

<sup>1</sup>School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK

<sup>2</sup>Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK

<sup>3</sup>Institute for Computing and Information Sciences, Radboud Universiteit, Toernooiveld 212, 6525 EC Nijmegen, NL

There are various gate sets used for describing quantum computation. A particularly popular one consists of Clifford gates and arbitrary single-qubit phase gates. Computations in this gate set can be elegantly described by the *ZX-calculus*, a graphical language for a class of string diagrams describing linear maps between qubits. The ZX-calculus has proven useful in a variety of areas of quantum information, but is less suitable for reasoning about operations outside its natural gate set such as multi-linear Boolean operations like the Toffoli gate. In this paper we study the *ZH-calculus*, an alternative graphical language of string diagrams that does allow straightforward encoding of Toffoli gates and other more complicated Boolean logic circuits. We find a set of simple rewrite rules for this calculus and show it is complete with respect to matrices over  $\mathbb{Z}[\frac{1}{2}]$ , which correspond to the approximately universal Toffoli+Hadamard gateset. Furthermore, we construct an extended version of the ZH-calculus that is complete with respect to matrices over any ring  $R$  where  $1 + 1$  is not a zero-divisor.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Related work . . . . .	4
1.2	Overview of the paper and structure of the main proof . . . . .	4
<b>2</b>	<b>The ZH-calculus</b>	<b>5</b>
2.1	The generators . . . . .	5
2.2	The rules . . . . .	7
2.3	Motivation for the rewrite rules . . . . .	8
2.4	Basic derived rules . . . . .	11
2.5	Labelled H-boxes . . . . .	15
2.6	!-box notation . . . . .	18
2.7	Annotated !-boxes . . . . .	18
<b>3</b>	<b>Special cases of multiply, intro and average</b>	<b>21</b>
3.1	The multiply rule . . . . .	21
3.2	The intro rule . . . . .	22
3.3	The average rule . . . . .	26
<b>4</b>	<b>Normal forms</b>	<b>27</b>
4.1	Normal form diagrams . . . . .	28

Miriam Backens: [m.backens@bham.ac.uk](mailto:m.backens@bham.ac.uk)

Aleks Kissinger: [aleks0@gmail.com](mailto:aleks0@gmail.com)

Hector Miller-Bakewell: [hmillerbakewell@gmail.com](mailto:hmillerbakewell@gmail.com), <https://hjmb.co.uk/>

John van de Wetering: [john@vdwetering.name](mailto:john@vdwetering.name), <http://vdwetering.name>

Sal Wolfs: [sal.wolfs@gmail.com](mailto:sal.wolfs@gmail.com)

4.2 Conditional reduction to normal form . . . . . 30

**5 Arithmetic 38**

5.1 Proving addition for natural numbers . . . . . 38

5.2 Proving addition for all integers . . . . . 41

5.3 Proving multiplication . . . . . 43

**6 Completeness 46**

**7 The ZH-calculus over arbitrary rings 52**

7.1 ZH-calculus over rings with a half . . . . . 52

7.2 The ZH-calculus over rings without a half . . . . . 56

**8 Modifications to the ZH-calculus 58**

8.1 Replacing the (o) rule . . . . . 59

8.2 The (hh) rule is not necessary . . . . . 61

8.3 Merging the intro and average rule . . . . . 61

8.4 ZH-calculus as a complete language for Toffoli+Hadamard circuits . . . . . 63

**9 Conclusion 64**

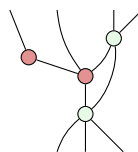
# 1 Introduction

Graphical calculi give us a compact way to express and reason about complex, interacting processes using *string diagrams*. String diagrams represent processes that can compose in sequence or parallel using a notation consisting of boxes or nodes connected by wires. Notably, wires can be left ‘open’ at one or both ends to indicate inputs and outputs of the composed process. For example, depicting sequential composition as  $\circ$  and parallel composition as  $\otimes$ , we can translate expressions such as the following one into a string diagram:



Such notation has proven convenient for expressing compositions and tensor products of linear maps (as used extensively in quantum theory), and more generally for expressing morphisms in a generic symmetric monoidal category.

A *graphical calculus* is both a collection of graphical building-blocks and a collection of equations between string diagrams that we can use to transform one diagram into another, typically equivalent, one. One of the best-studied graphical calculi is the *ZX-calculus* [11, 12, 14, 44], which has been applied extensively in the study of quantum circuits and related structures in quantum computation (e.g. measurement-based quantum computing [6, 19, 31], fault-tolerant quantum computations [18, 24, 28], error-correcting codes [10], circuit optimisation [20, 22, 32], and classical simulation [33, 35]). ZX-diagrams consist of *spiders*, a type of well-behaved linear map that is depicted by coloured dots:



Quantum circuits lie at the heart of the *circuit model* of quantum computation, which represents the quantum part of the computation as a large unitary operator arising from compositions and tensor products of basic gates [41]. There are many choices of basic gates which are *universal*, in

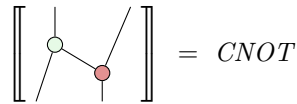
the sense that they can construct or approximate arbitrary  $2^n \times 2^n$  unitary matrices. A popular choice is the *Clifford+phase* set of gates:

$$\text{CNOT} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad Z_\alpha := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$$

This set of gates is universal, here in the sense that it can exactly express any unitary matrix. A common finite restriction of this infinite set is the *Clifford+T* gate set, which replaces  $Z_\alpha$  with  $T := Z_{\pi/4}$ . This set of gates is able to approximate any unitary up to arbitrary finite precision. A further restriction to the *Clifford* gate set (CNOT,  $H$ ,  $S := T^2$ ) is no longer universal, and in fact yields only circuits that can be efficiently simulated on a classical computer [1]. Nevertheless, Clifford circuits play an important role in quantum computation [41], quantum error correction [7, 23, 25], and many quantum communication protocols.

Notably, these three important families of circuits map straightforwardly to three fragments of the ZX-calculus: where parameters are restricted to integer multiples of  $\frac{\pi}{2}$  (Clifford), restricted to integer multiples of  $\frac{\pi}{4}$  (Clifford+T), or are unrestricted (Clifford+phase). Furthermore, each of these fragments of the ZX-calculus has been proven *complete* in the sense that two ZX-diagrams which evaluate to the same linear operator are provably equal using just diagram rewrite rules [4, 29, 40].

However, the ZX-calculus remains closely tied to the structure of the Clifford+phase family of circuits. The further one gets from this family, the more awkward it becomes to work with computations using the ZX-calculus. For instance, the CNOT gate (‘controlled-NOT’) can be expressed quite simply in the ZX-calculus:



In contrast, the ‘controlled-controlled-NOT’, commonly referred to as the Toffoli gate, is considerably more complicated, with typical presentations requiring many more generators which cannot be easily manipulated using the ZX-calculus rules [14, Ex. 12.10].

The problem with the Toffoli gate is that it acts on basis states in a manner that is not  $\mathbb{Z}_2$ -affine. For example, the Boolean CNOT is the map  $\text{CNOT} : \{0,1\}^2 \rightarrow \{0,1\}^2$  given by  $\text{CNOT}(x,y) = (x, x \oplus y)$ , where  $\oplus$  denotes XOR. Such a Boolean function is called *affine*, in analogy to the theory of polynomials, since it only contains terms of multiplicative degree at most 1 that are then added together (XOR acts as addition on the field  $\mathbb{Z}_2$ ). Similarly, we say the NOT gate  $\text{NOT}(x) = x \oplus 1$  is affine, since again it is a sum of terms of degree at most 1. In the ZX-calculus there are two main types of generators: Z-spiders and X-spiders. Both of these are closely related to affine Boolean functions: the Z-spider is based on the COPY Boolean map  $\text{COPY}(x) = (x,x)$ , while the X-spider is based on XOR. As being affine is baked into the generators, some tricks are required to represent non-affine Boolean functions in the ZX-calculus, such as the Toffoli gate  $\text{TOF}(x,y,z) = (x,y,(x \cdot y) \oplus z)$  that contains the degree-2 term  $x \cdot y$ .

The way we solve this issue in this paper is by introducing a third diagrammatic type of generator beyond the Z and X spiders: the H-box. An  $n$ -input 0-output H-box is defined by  $|x_1 \cdots x_n\rangle \mapsto (-1)^{x_1 \cdots x_n}$ . This can then be used in combination with Z-spiders (i.e. copy maps) to represent the ‘controlled-controlled-Z’ gate, and thus the Toffoli gate.

Another way to view the H-box generator is as a generalisation of the Hadamard gate to a linear map with an arbitrary number of inputs and outputs.

With this new generator comes a new set of graphical rewrite rules that allow us to reason more efficiently about quantum computation involving non-affine Boolean functions. The resulting graphical language is called the *ZH-calculus*, since its main generators are Z-spiders and H-boxes. The main topic of this paper is to show that various fragments of the ZH-calculus are complete. Specifically, we find a set of particularly simple and compelling rewrite rules (see Figure 1) that are complete for the approximately universal Toffoli-Hadamard gate set.<sup>1</sup> Arguably, this is the

<sup>1</sup>The Toffoli-Hadamard gate set is only approximately universal for ‘real’ quantum computing, where all unitaries

smallest complete rule set for an approximately universal fragment of quantum computing (we discuss this claim in more detail in Section 2.3). Building on this work, we show that for any ring where  $2 := 1 + 1$  is not a zero divisor, we can generalise the ZH-calculus in order to express all matrices over this ring, and we exhibit a complete set of rules for this generalisation as well.

This article is based on a conference paper by Backens and Kissinger [5] which first introduced the ZH-calculus and proved completeness for the ZH-calculus with complex parameters, an unpublished preprint of van de Wetering and Wolffs [45] which proved completeness for the phase-free fragment of the calculus, and the DPhil thesis of Miller-Bakewell [39] which showed how the ZH-calculus and its completeness could be extended to arbitrary rings. The formal developments in those papers have been expanded and unified, and the technical content has been extended significantly in three ways: we find a simpler complete rule set for the phase-free fragment, we give a completely self-contained proof of phase-free completeness based on an encoding of arithmetic into ZH, and we extend the ZH-calculus to any ring where  $2 := 1 + 1$  is not a zero divisor.

## 1.1 Related work

In the earliest complete version of the ZX-calculus, a new generator, the ‘triangle’, was introduced in order to represent non-affine Boolean functions [40]. This has later also been used to prove completeness of the fragment corresponding to the Toffoli+Hadamard gate set that we also study [47], as well as to find an axiomatisation of the ZX-calculus over arbitrary rings [52]. In [16], a complete ruleset for classical circuits, i.e. natural number matrices, was found which uses almost the same ruleset as we do in the phase-free fragment. The difference is that [16] treats the AND gate as an atomic non-symmetric generator, while we decompose it into H-boxes.<sup>2</sup> The seminal complete graphical language for the related fragment of integer matrices is the *ZW-calculus* [26] which has generators based on the two different types of entanglement that are possible in tripartite qubit systems [13]. The related language  $\text{RING}_R$  [39] was designed to unify the study of qubit graphical calculi parameterised by phase rings, and indeed both parameterised ZH (Section 7) and the interpretation of ZH as operations on Boolean functions (Section 2.3) fit into this scheme. Therefore there are natural maps from  $\text{RING}_{\mathbb{B}}$  to ZH and from  $\text{RING}_R$  to  $\text{ZH}_R$ , the second of which is exhibited in [39]. Note that [9] shows that, under certain assumptions, there are only three basic graphical calculi for qubits: ZX, ZW, and ZH. The results presented in the current paper prove that the youngest of these calculi, ZH, is complete, concluding the program of finding complete graphical calculi for qubits.

Using the ZH-calculus, an efficient description of *hypergraph states* can be given [37], which allows for a description of ZH-diagrams in terms of *path-sums* [2, 49, 51]. There is also a close connection between ZH-diagrams and quantum multiple-valued decision diagrams [50]. The ZH-calculus has been used to give a graphical description of *AKLT states*, a particularly canonical type of condensed matter system [17], and of spin-networks [21].

## 1.2 Overview of the paper and structure of the main proof

In Section 2 we introduce the ZH-calculus. We present and motivate the rewrite rules and we introduce the notation we will use throughout the paper (the rules can be found in Figure 1, their Boolean counterparts that motivate them can be found in Figure 2). Note that in particular, in subsection 2.5, we define ‘labelled H-boxes’, which are derived generators that form an essential ingredient to our proof of completeness of the ZH-calculus.

The majority of the paper, Sections 3–6, is devoted to this proof of completeness.

First, in Section 3, we prove several special cases of three families of equations that lie at the core of our completeness proof. These families are called ‘multiply’, ‘intro’, and ‘average’ (see, respectively,  $(M_{a,b})$ ,  $(I_a)$  and  $(A_{a,b})$ ) and apply to labelled H-boxes.

---

contain only real numbers. However, such real unitaries can simulate with constant overhead unitaries with complex entries [43]. The notion of approximate universality is here hence different than that of Clifford+T, and might better be called ‘computationally universal’. We will however not make this technical distinction in the remainder of the paper.

<sup>2</sup>In fact, his rules were inspired by the rules of the phase-free ZH-calculus presented in an earlier preprint [45] (personal communication, Comfort).

Then, in Section 4, we introduce normal form diagrams. A normal form diagram corresponds in a precise way to the matrix the diagram represents, and hence is unique for a given underlying linear map. Given two diagrams representing the same linear map, the ability to reduce both of them to normal form then shows that they can be transformed into the same diagram, and hence that the calculus is complete. In this section, we give a proof that the reduction to normal form can be done *conditional* on having proved all instances of multiply, intro and average. This conditional proof works by showing that each generator can be transformed into normal form (Lemmas 4.5 and 4.20), that a tensor product of normal forms can be reduced to normal form (Corollary 4.12), and that connecting any of the wires in a normal form results in a diagram that can again be brought to normal form (Corollary 4.19). With these steps and the condition, any diagram can be brought into normal form (Proposition 4.21). It remains to prove all instances of multiply, intro and average.

In Section 5, we prove that we can do basic arithmetic on the labels of H-boxes: addition of H-box labels in Proposition 5.10, and multiplication of H-box labels (which is exactly the general multiply rule) in Proposition 5.16. Then, in Section 6, we give proofs of the general average rule (Proposition 6.2) and general intro rule (Proposition 6.4). Together with the conditional reduction to normal form, this finishes our proof of completeness (Theorem 6.5). The reason we structured the proof like this, starting with a conditional proof of the reduction to normal form, is because some of the proofs in Sections 5 and 6 use this reduction to normal form (with a careful analysis that shows the required instances of average, intro and multiply were all proved in Section 3), since we could not find a more direct way to prove them.

In Section 7, we shift to studying the ZH-calculus over arbitrary rings. We prove its completeness by assuming the multiply, intro, and average rules as axioms. Then in Section 8, we study a couple of modifications to the calculus and its rules. We end with some concluding remarks in Section 9.

## 2 The ZH-calculus

The ZH-calculus is a graphical language for expressing maps with zero or more inputs and outputs as certain string diagrams called ZH-diagrams. As a convention, we will draw inputs as wires entering the bottom of the diagram and outputs as wires exiting the top.

Throughout this paper, we will distinguish formal ZH-diagrams  $D : m \rightarrow n$  with  $m$  input wires and  $n$  output wires, from their standard interpretation as linear maps  $\llbracket D \rrbracket : (\mathbb{C}^2)^{\otimes m} \rightarrow (\mathbb{C}^2)^{\otimes n}$ , where  $(\mathbb{C}^2)^{\otimes m} \cong \mathbb{C}^{2^m}$  is the  $m$ -fold tensor product of the 2D vector space  $\mathbb{C}^2$ .

**Remark 2.1.** In categorical terms, ZH-diagrams are morphisms in the free PROP  $\mathcal{ZH}$  presented by the ZH-calculus, considered as a symmetric monoidal category, and  $\llbracket - \rrbracket$  is a strong monoidal functor from  $\mathcal{ZH}$  to  $(\mathbf{Vect}_{\mathbb{C}}, \otimes)$ , the category of finite-dimensional complex vector spaces, with the monoidal product given by the tensor product. Though familiarity with symmetric monoidal categories and PROPs is not required to read this paper, the interested reader can find an accessible introduction in Chapter 2 of [53].

### 2.1 The generators

ZH-diagrams have three types of generators: *Z-spiders* represented by white dots, *H-boxes* represented by white boxes, and a *star* generator represented by  $\star$ . These generators are interpreted as linear maps on copies of  $\mathbb{C}^2$ . We denote the standard (also called *computational*) basis of  $\mathbb{C}^2$  in Dirac notation as:

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

We will be using this Dirac ‘ket’ notation  $|\psi\rangle$  to denote a state in a vector space. We will also denote the ‘bra’  $\langle\psi|$  for the linear map that calculates the inner product with  $|\psi\rangle$ . That is, inputting a state  $|\phi\rangle$  gives  $\langle\psi|(|\phi\rangle) = \langle\psi|\phi\rangle$ .

The computational basis states  $|0\rangle$  and  $|1\rangle$  extend naturally to a basis for  $(\mathbb{C}^2)^{\otimes n}$  by interpreting bitstrings as tensor products:

$$|x_1 \dots x_n\rangle := |x_1\rangle \otimes \dots \otimes |x_n\rangle$$

We can define the interpretation of Z-spiders and H-boxes in terms of these basis states. Z-spiders and H-boxes can have any number of inputs and outputs, and are then interpreted as the following linear maps:

$$\left[ \begin{array}{c} n \\ \vdots \\ \text{X} \\ \vdots \\ m \end{array} \right] := |0\rangle^{\otimes n} \langle 0|^{\otimes m} + |1\rangle^{\otimes n} \langle 1|^{\otimes m} \quad \left[ \begin{array}{c} n \\ \vdots \\ \text{H} \\ \vdots \\ m \end{array} \right] := \sum (-1)^{i_1 \dots i_m j_1 \dots j_n} |j_1 \dots j_n\rangle \langle i_1 \dots i_m|$$

where  $\llbracket \cdot \rrbracket$  denotes the map from diagrams to matrices. The sum in the second equation is over all  $i_1, \dots, i_m, j_1, \dots, j_n \in \{0, 1\}$  so that an H-box represents a matrix with all entries equal to 1, except the bottom right element, which is  $-1$ .

Finally we have the generator *star*, which has zero inputs and outputs. Its interpretation is:

$$\llbracket \star \rrbracket := \frac{1}{2}$$

Straight and curved wires have the following interpretations:

$$\llbracket | \rrbracket := |0\rangle\langle 0| + |1\rangle\langle 1| \quad \llbracket \cup \rrbracket := |00\rangle + |11\rangle \quad \llbracket \cap \rrbracket := \langle 00| + \langle 11|.$$

When two diagrams are juxtaposed, the corresponding linear map is the tensor product (a.k.a. Kronecker product) of the matrices corresponding to the individual diagrams. A sequential composition of two diagrams is interpreted as the matrix product of the matrices corresponding to the individual diagrams:

$$\left[ \begin{array}{c} \dots \\ \boxed{D_1} \\ \dots \end{array} \right] \left[ \begin{array}{c} \dots \\ \boxed{D_2} \\ \dots \end{array} \right] := \left[ \begin{array}{c} \dots \\ \boxed{D_1} \\ \dots \end{array} \right] \otimes \left[ \begin{array}{c} \dots \\ \boxed{D_2} \\ \dots \end{array} \right] \quad \left[ \begin{array}{c} \dots \\ \boxed{D_2} \\ \dots \\ \boxed{D_1} \\ \dots \end{array} \right] := \left[ \begin{array}{c} \dots \\ \boxed{D_2} \\ \dots \end{array} \right] \circ \left[ \begin{array}{c} \dots \\ \boxed{D_1} \\ \dots \end{array} \right]$$

Some diagram motifs appear so often that we create shorthands for them, called *derived* generators. We define the derived grey spider and the grey spider with a NOT as:

$$(X) \quad \begin{array}{c} \dots \\ \text{X} \\ \dots \end{array} := \star \quad \begin{array}{c} \dots \\ \text{NOT} \\ \dots \end{array} := \star \quad (1)$$

The generator  $\begin{array}{c} \dots \\ \text{X} \\ \dots \end{array}$  acts as XOR on the computational basis while  $\begin{array}{c} \dots \\ \text{NOT} \\ \dots \end{array}$  acts as NOT:

$$\llbracket \begin{array}{c} \dots \\ \text{X} \\ \dots \end{array} \rrbracket = |0\rangle\langle 00| + |0\rangle\langle 11| + |1\rangle\langle 01| + |1\rangle\langle 10| \quad \llbracket \begin{array}{c} \dots \\ \text{NOT} \\ \dots \end{array} \rrbracket = |0\rangle\langle 1| + |1\rangle\langle 0|.$$

In some of the later sections there will be diagrams with many NOTs. To avoid overcrowding, we will draw a red dashed edge for an edge between Z-spiders with a NOT on it:

$$\begin{array}{c} \dots \\ \vdots \\ \text{---} \\ \vdots \\ \dots \end{array} := \begin{array}{c} \dots \\ \text{NOT} \\ \dots \end{array} \quad (2)$$

We also introduce the derived *negate* spider (a white NOT):

$$(Z) \quad \begin{array}{c} \dots \\ \text{NOT} \\ \dots \end{array} := \begin{array}{c} \dots \\ \text{NOT} \\ \dots \end{array} \quad (3)$$

The negate white spider with one input and output acts like the Z gate:

$$\llbracket \begin{array}{c} \dots \\ \text{NOT} \\ \dots \end{array} \rrbracket = |0\rangle\langle 0| - |1\rangle\langle 1|$$

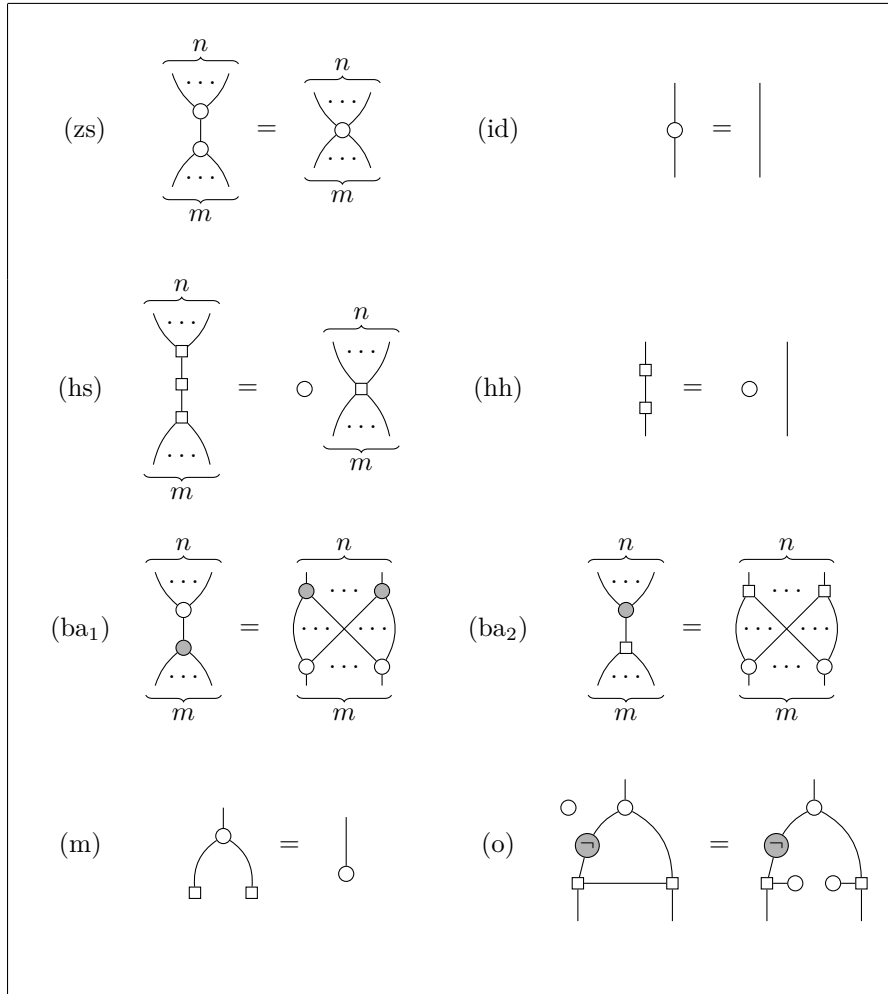


Figure 1: The rules of the ZH-calculus. Throughout,  $m, n$  are nonnegative integers. The right-hand sides of both *bialgebra* rules  $(ba_1)$  and  $(ba_2)$  are complete bipartite graphs on  $(m+n)$  vertices, with an additional input or output for each vertex. The horizontal edges in equation  $(o)$  are well-defined because only the topology matters and we thus do not need to distinguish between inputs and outputs of generators.  $(zs)$  and  $(hs)$  stand respectively for *Z-spider* and *H-spider*;  $(id)$  for *identity*;  $(ba_1)$  and  $(ba_2)$  for *bialgebra*;  $(m)$  for *multiply*; and  $(o)$  for *ortho*.

## 2.2 The rules

The ZH-calculus comes with a set of rewrite rules shown in Figure 1. Of the 8 rules in Figure 1, 7 are ‘obvious’ in the sense that they express simple properties of the structure of the underlying Boolean functions (see Section 2.3). The only outlier is  $(o)$  which seems more arbitrary. In Section 8.1 we will see that we can replace  $(o)$  with two smaller rules.

Additionally, the calculus has the meta-rule that *only topology matters*. This means that two diagrams are considered equal when one can be topologically deformed into the other, while respecting the order of the inputs and outputs. Finally, the two generators are considered to be symmetric and undirected, so that the following equations also hold:



These symmetry properties also hold for the derived grey spider and NOT gate.

**Proposition 2.2.** The ZH-calculus is sound.



*Proof.* It is straightforward to check that the symmetry properties for each generator and the finite rules of Figure 3 are sound by concrete calculation. (ba<sub>1</sub>) and (ba<sub>2</sub>) can both be reduced to a finite set of equations for which soundness is easily checked and from which the general infinite family of rules can be derived using induction (see for instance [14, Theorem 9.71]). The rules (zs) and (hs) express equations related to Frobenius algebras for which soundness can also be checked in a standard manner, see for instance [15]. Soundness of the meta rule ‘only topology matters’ follows by considering the string diagrams as morphisms in a compact closed category [42]. ■

The converse to soundness is completeness. Determining that the ZH-calculus is complete will be the main objective of this paper. A third important property of diagrammatic languages is *universality*. It can be seen from the definitions of the generators that their interpretations as linear maps are all matrices over  $\mathbb{Z}[\frac{1}{2}]$ , and therefore all  $\otimes$ - and  $\circ$ -products of the generators also have interpretations as matrices over  $\mathbb{Z}[\frac{1}{2}]$ . Universality means that we can represent *any* matrix over  $\mathbb{Z}[\frac{1}{2}]$  using the ZH-calculus. We cannot prove this property yet, but it will follow immediately from Theorem 4.4. These matrices, and hence the ZH-calculus, are closely related to the approximately universal gate set Toffoli+Hadamard [3]. We discuss the precise connection in more detail in Section 8.4.

### 2.3 Motivation for the rewrite rules

Let us give some motivation for the rewrite rules of Figure 1. Each of the three main (derived) generators of the ZH-calculus, the Z-spiders, X-spiders and H-boxes, corresponds to a family of Boolean functions. Using this correspondence, each of the rules of Figure 1 can be seen to be equivalent to an equation that holds between Boolean functions.

In order to make this correspondence, we first recall that we can lift a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  to a linear map  $\hat{f} : \mathbb{C}^{2^n} \rightarrow \mathbb{C}^{2^m}$  by its action on the computational basis states:

$$\hat{f} |x_1 \dots x_n\rangle = |f(x_1 \dots x_n)\rangle. \tag{4}$$

Define the COPY family of Boolean functions  $\text{COPY}^n : \{0, 1\} \rightarrow \{0, 1\}^n$  by  $\text{COPY}(x) = (x, \dots, x)$ . Then  $\widehat{\text{COPY}}^n$  is equal as a linear map to the Z-spider with a single input and  $n$  outputs. Similarly, the X-spider with  $n$  inputs and a single output is equal to  $\widehat{\text{XOR}}^n$ , while the single input, single output NOT generator is, unsurprisingly, equal to  $\widehat{\text{NOT}}$ . H-boxes are not induced by Boolean functions, as the linear map of an H-box contains negative numbers. Yet composing an H-box with a further Hadamard gives the family of Boolean AND functions. In summary:

Diagrammatic equations (5) showing the correspondence between gates and Boolean functions:

- NOT gate (square with a dot) is equivalent to a NOT symbol (circle with a dot).
- COPY gate (square with multiple outputs) is equivalent to a Z-spider (circle with multiple outputs).
- XOR gate (square with multiple outputs) is equivalent to an X-spider (circle with multiple outputs).
- AND gate (square with multiple outputs) is equivalent to an H-box with a square on top.

Seeing this, one might wonder why we simply did not choose some other generator that can represent the AND directly. The reason for this is that it is in fact useful to represent the inputs and outputs of AND asymmetrically.

Indeed, the COPY and XOR maps are symmetric under interchange of the inputs and outputs, a property also known as *flexsymmetry* [8]:

Diagrammatic equations (6) showing flexsymmetry:

- A Z-spider with multiple outputs is equal to a Z-spider with multiple inputs.
- An X-spider with multiple outputs is equal to an X-spider with multiple inputs.

This property is not true for AND:

Diagrammatic equation (7) showing that AND is not flexsymmetric:

- An H-box with a square on top is not equal to a square on top of an H-box.

We can now re-express the rules of Figure 1 with all the generators presented as Boolean functions, as shown in Figure 2. The first five rules in this figure are exactly equal to the rules

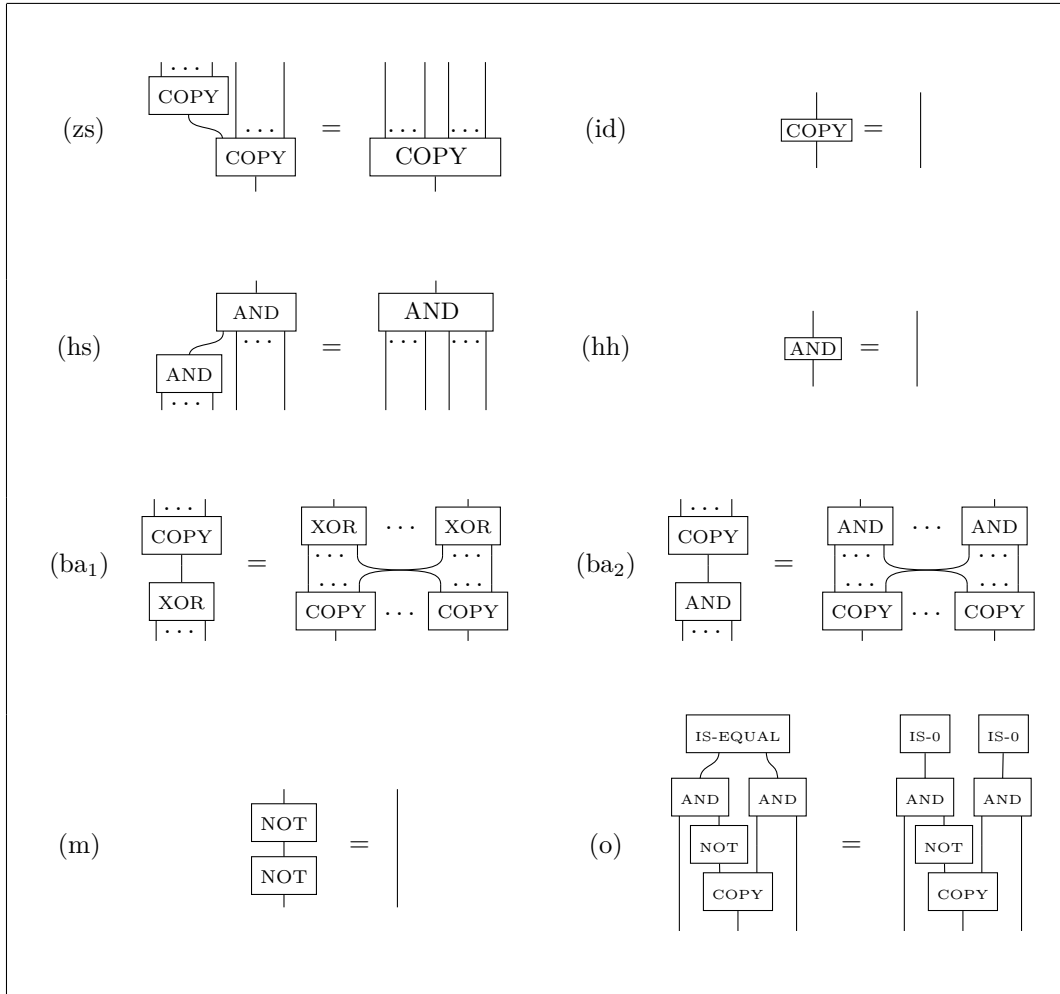


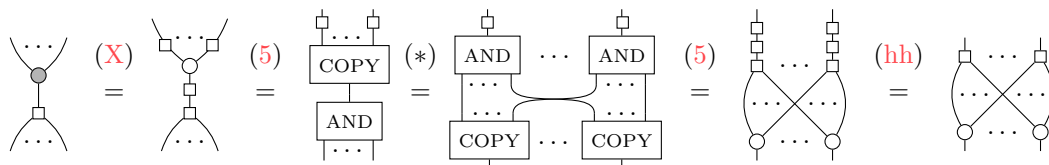
Figure 2: The rules of the ZH-calculus presented as equations between Boolean functions.

of Figure 1 when we translate the Boolean functions back into the generators according to (5). The other three are not equal, but can be proven to be equivalent using the first five equations (which we will do below), and hence the two rulesets are equivalent. For the representation of (o) we introduce two new Boolean ‘predicates’:

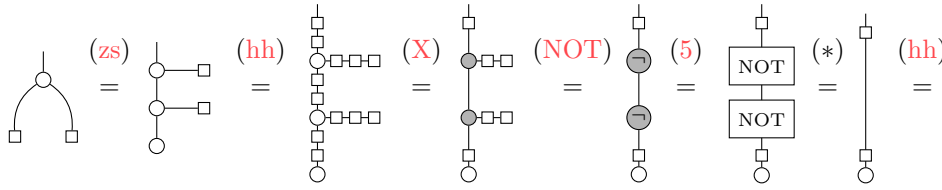
$$\begin{array}{c} \boxed{\text{IS-EQUAL}} \\ \hline \end{array} := \cap \qquad \begin{array}{c} \boxed{\text{IS-0}} \\ \hline \end{array} := \bullet \quad (8)$$

Recall that the linear map corresponding to the cap is  $\langle 00| + \langle 11|$ , and hence we can interpret IS-EQUAL as selecting for the case where both inputs are equal to one another. Similarly IS-0 is just  $\langle 0|$  and selects for its input to be 0.

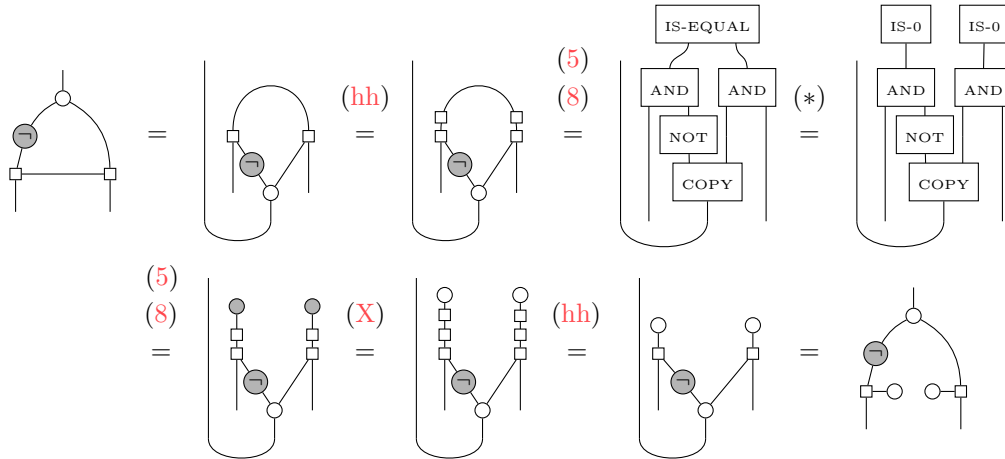
Let us demonstrate how the rules of Figure 2 suffice to prove the rules of Figure 1. We will ignore scalar factors for these derivations (correct scalars can be introduced by repeated use of Lemma 2.3). First, let us derive (ba<sub>1</sub>) from its Boolean counterpart. We denote the usage of the ‘Boolean version’ of (ba<sub>1</sub>) by (\*).



Next, we prove (m) from the cancellation of two NOT gates. Here (\*) denotes the application of the ‘Boolean version’ of (m):



Finally, we derive (o) from the corresponding relation between Boolean functions, where here (\*) denotes the Boolean version of (o):



The converse proofs that the rules of Figure 1 imply those of Figure 2 are similar (and will also implicitly follow from completeness of the ZH-calculus).

Looking at Figure 2 we see that (zs) and (hs) express the associativity of COPY, respectively AND, and that (id) and (hh) express the triviality of COPY and AND when they have only a single input. We see that the only difference between (ba<sub>1</sub>) and (ba<sub>2</sub>) is which Boolean function comes before the COPY. These rules are in fact special cases of the rule that any Boolean function copies through COPY [14, Prop. 8.19]. The rule (m) expresses that NOT is self-inverse, and finally there is (o), which requires some more explanation. On both sides of the equation, the middle input is copied and sent to both AND gates, but one of the copies is negated. Hence the output of at least one of the AND gates will be zero, so that the only way for the outputs of these AND gates to be equal is if they are both zero. This is the property that (o) expresses. As this is still not a particularly concise motivation, we present in Section 8.1 an alternative pair of rules that can replace (o) and which do have a more concise motivation.

It might be surprising that this rule set expressing identities about Boolean functions suffices to reason about quantum computation. However, what is not apparent from this rule set is that the AND gate is ‘broken up’ into two parts, given by the H-boxes and the Hadamard. The Hadamard gives the ‘Fourier transform’ between the Z and X spiders and relates the COPY and XOR maps in a way that does not follow easily from their definition as Boolean functions. In fact, in [16], a calculus for integer matrices is presented using XOR, COPY and AND as generators, and which uses a rule set that is essentially the same as that of Figure 2, but with (o) replaced by the rules we present in Section 8.1. Hence, the difference between this ‘classical’ calculus and our ‘quantum’ calculus is the usage of the Hadamard and the necessity of negative numbers this requires.

Beyond having a nice interpretation in terms of Boolean identities, our rule set is furthermore arguably preferable to that of other (approximately) universal complete graphical calculi for qubits in terms of simplicity. This is because:

1. It only requires a small number of rules (8, of which one is actually superfluous, cf. Section 8.2).
2. Each of these rules only involve a few (derived) generators.
3. A priori, the set of generators does not involve any further information, such as a label taken from a ring or group.

4. As a result, no rule involves a ‘side condition’ on the elements of the ring or group.

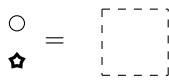
While there are other calculi that satisfy items 1) and 2) — notably ZX $_{\pi/4}$  [29], universal ZX [46], RING [39], and ZQ [38] also only require a small number of rules — each of these requires either a complicated side condition on at least one the rules, or directly encodes properties of the underlying ring in its generators. Other calculi like ZW [26] and  $\Delta$ ZX [47] also do not have a lot of rules, but many of them lack a clear interpretation.

This is of course not to say that those calculi are ‘worse’, but just that the ZH-calculus has a particularly elegant set of rewrite rules, which somehow neatly capture the complexity of quantum computing.

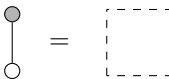
### 2.4 Basic derived rules

In this section we will prove some basic but useful rewrite rules that we will use throughout the paper. Since the statements and proofs are often very short we will state all the lemmas first and then all the proofs. Lemmas 2.3–2.8 are different ways to cancel or simplify scalar diagrams. Lemmas 2.9–2.14 are various ways to simplify the derived generators. Lemmas 2.15–2.20 and 2.29 give ways to commute Z’s, NOTs and Hadamards through the other generators. Lemmas 2.21–2.28 govern the interaction of states with generators (mostly showing that many states can copy through a generator), and finally Lemma 2.30 is the Hopf rule that holds between the Z and X spider. While some of these lemmas are direct special cases of the rules, in particular of (ba<sub>1</sub>) and (ba<sub>2</sub>), we include them here for convenience of reference, and to give a more complete set of ‘state commutation’ rules. Note that for Lemma 2.15, if  $m = 0$ , then instead of the  $\star$ ’s, there is a single  $\circ$ ; similarly for Lemma 2.16, if  $m = 0$ , then instead of the  $\circ$ ’s, there is a single  $\star$ .

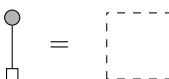
**Lemma 2.3.**



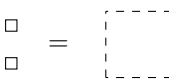
**Lemma 2.4.**



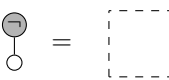
**Lemma 2.5.**



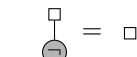
**Lemma 2.6.**



**Lemma 2.7.**



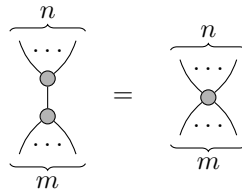
**Lemma 2.8.**



**Lemma 2.9.**



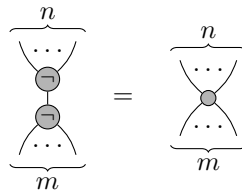
**Lemma 2.10.**



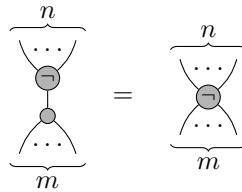
**Lemma 2.11.**



**Lemma 2.12.**



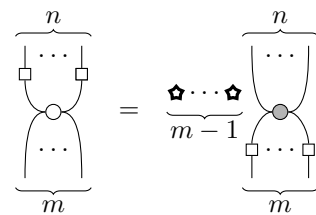
**Lemma 2.13.**



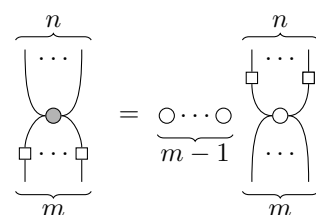
**Lemma 2.14.**



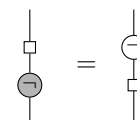
**Lemma 2.15.**



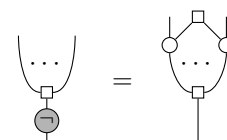
**Lemma 2.16.**



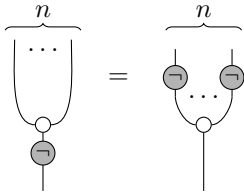
**Lemma 2.17.**



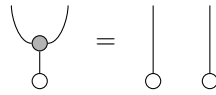
**Lemma 2.18.**



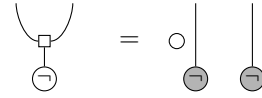
**Lemma 2.19.**



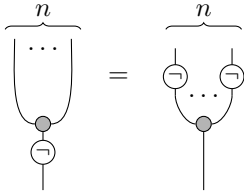
**Lemma 2.23.**



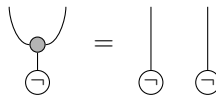
**Lemma 2.28.**



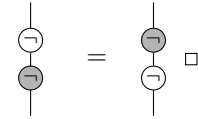
**Lemma 2.20.**



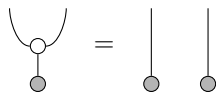
**Lemma 2.24.**



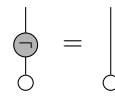
**Lemma 2.29.**



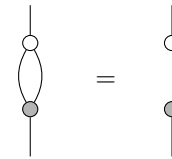
**Lemma 2.21.**



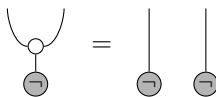
**Lemma 2.25.**



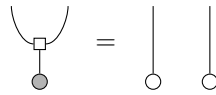
**Lemma 2.30.**



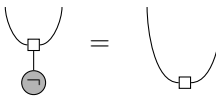
**Lemma 2.22.**



**Lemma 2.26.**

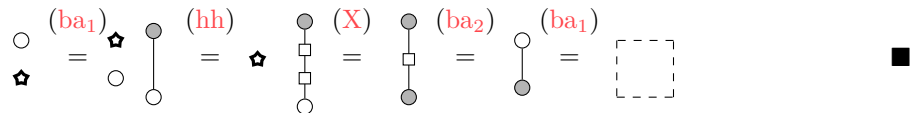


**Lemma 2.27.**

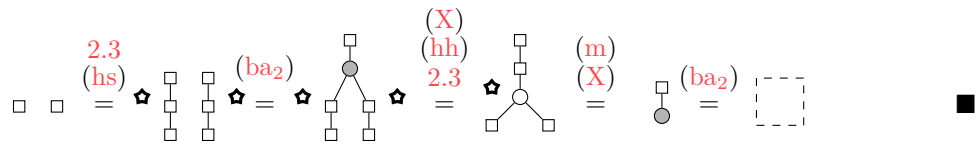


We now prove all the above lemmas. The proofs are all quite basic and straightforward, except that of Lemma 2.28 which is the only one to require (o).

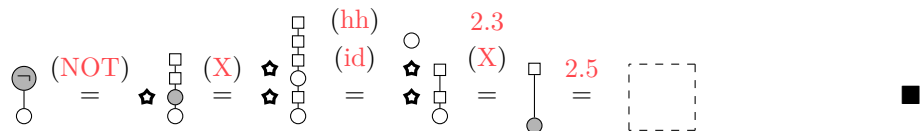
*Proof of Lemmas 2.3 and 2.4.*



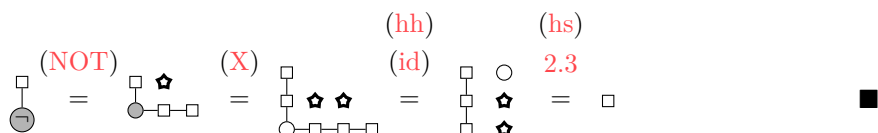
*Proof of Lemmas 2.5 and 2.6.*



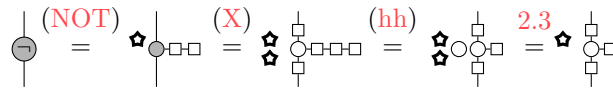
*Proof of Lemma 2.7.*



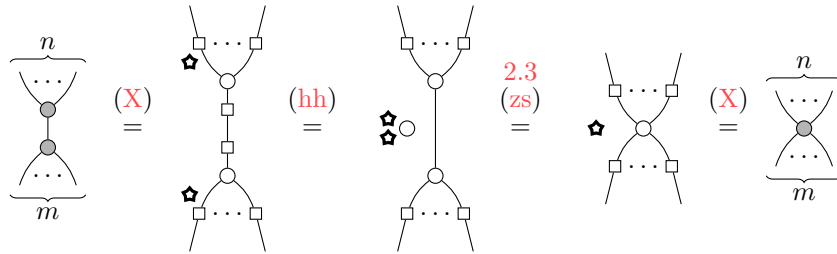
*Proof of Lemma 2.8.*



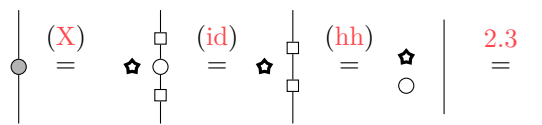
Proof of Lemma 2.9.



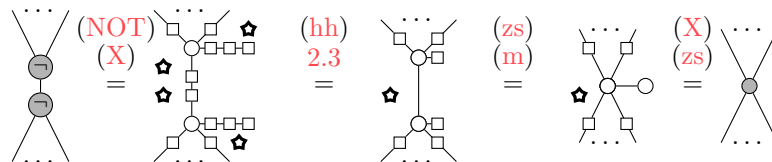
Proof of Lemma 2.10.



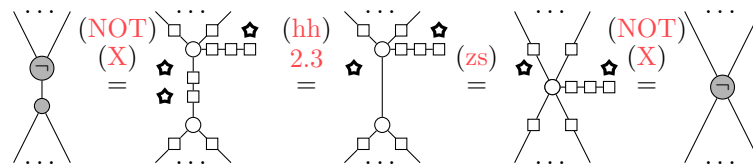
Proof of Lemma 2.11.



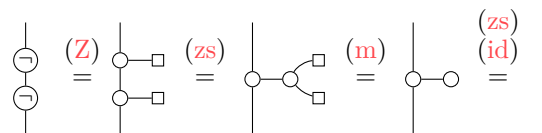
Proof of Lemma 2.12.



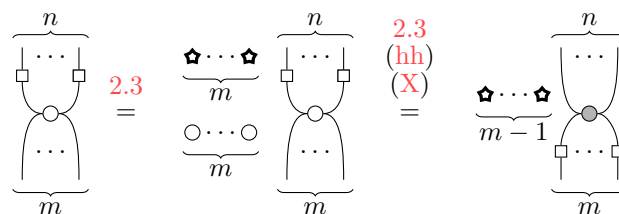
Proof of Lemma 2.13.



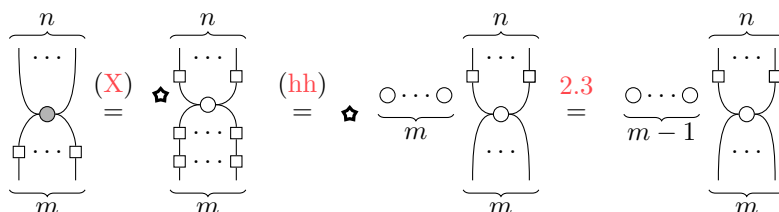
Proof of Lemma 2.14.



Proof of Lemma 2.15.



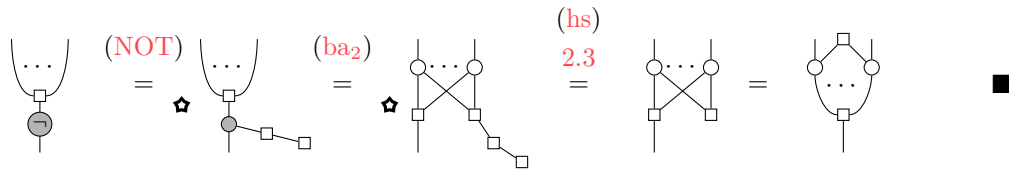
Proof of Lemma 2.16.



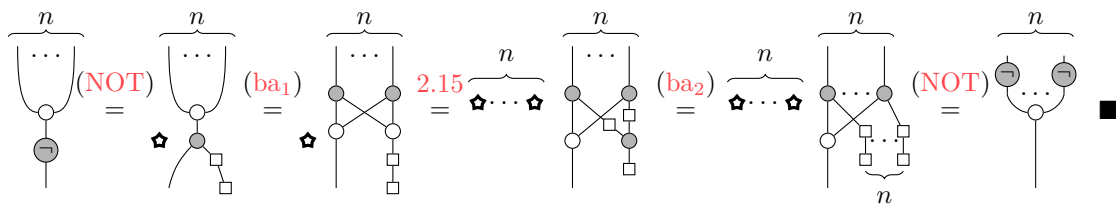
Proof of Lemma 2.17.



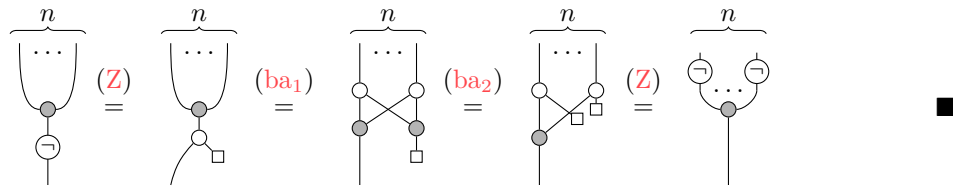
Proof of Lemma 2.18.



Proof of Lemma 2.19.

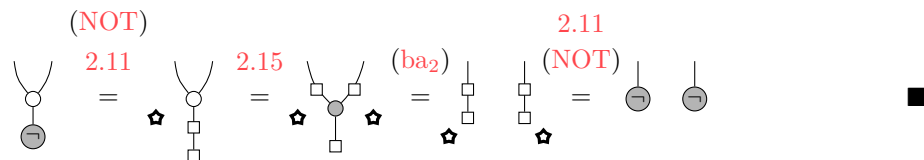


Proof of Lemma 2.20.

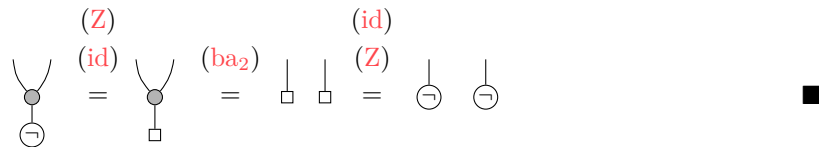


Proof of Lemmas 2.21, 2.23, and 2.26. These are just applications of (ba<sub>1</sub>) and (ba<sub>2</sub>).

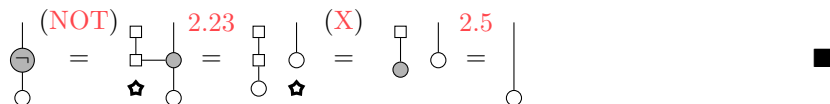
Proof of Lemma 2.22.



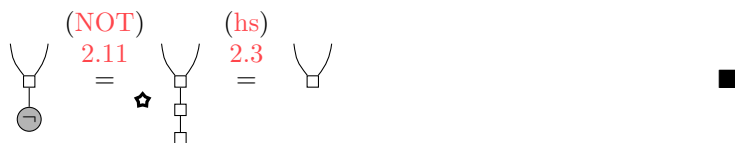
Proof of Lemma 2.24.



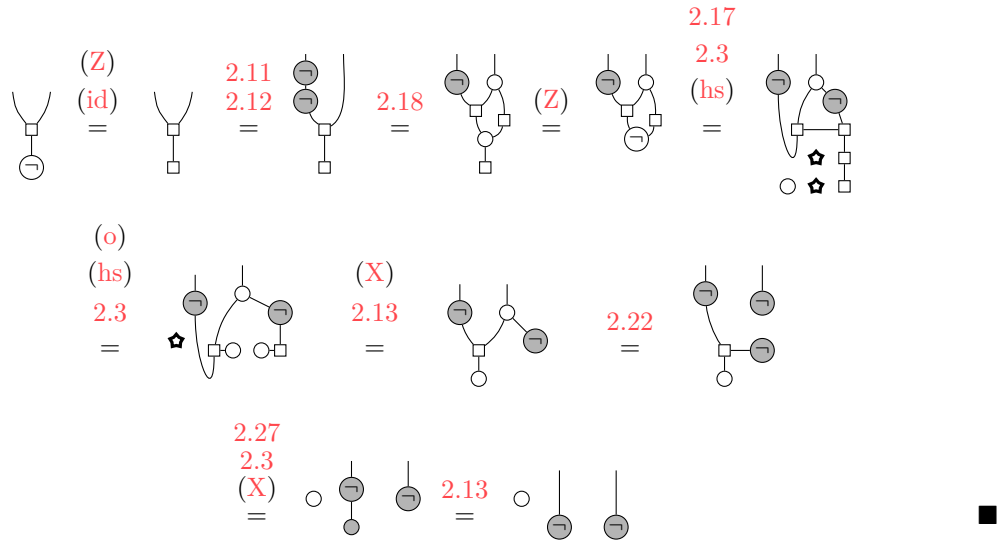
Proof of Lemma 2.25.



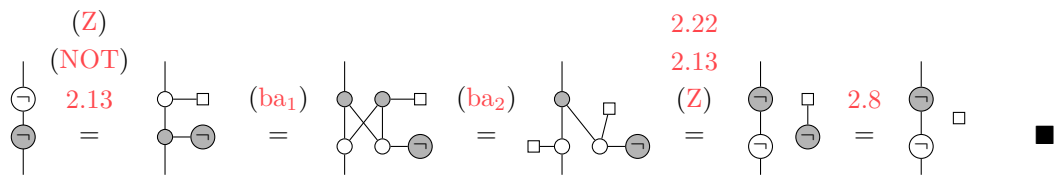
Proof of Lemma 2.27.



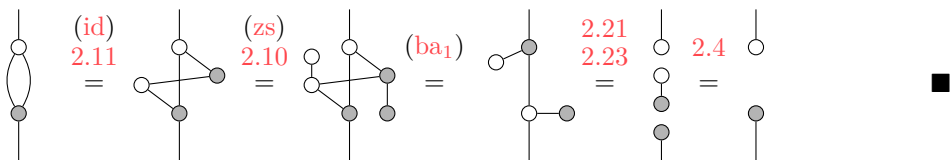
Proof of Lemma 2.28.



Proof of Lemma 2.29.



Proof of Lemma 2.30.



## 2.5 Labelled H-boxes

The matrix corresponding to an H-box is filled with 1's, except for the bottom right position where there is a  $-1$ . Calculating the matrix of some other diagrams in the ZH-calculus, we see that their matrix is similar, but instead of a  $-1$  in the bottom right corner there is some other number  $a \in \mathbb{Z}$ . In order to make this connection clearer, we introduce some new notation, where we write an H-box with a label of  $a$  inside it, to denote it is equal to a vector of 1's with an  $a$  in the bottom position. When talking about labelled H-boxes in text, we will often write  $H(a)$  to denote we are referring to an H-box labelled by  $a$  (leaving the arity implicit, which should be clear from context).

We begin with H-boxes of arity 1, corresponding to vectors  $(\frac{1}{a})$ , which we will define inductively for non-negative integers using a 'successor' gadget; H-boxes labelled by negative integers will then be defined from the positive ones using a 'negate' gadget.

**Definition 2.31.** Let

$$\boxed{-1} := \square \tag{9}$$

and for any  $a \in \mathbb{Z}$  such that  $a \geq 0$ , define:

$$\boxed{a} := \begin{array}{c} \ominus \star \\ \square \\ \boxed{a-1} \end{array} \tag{10}$$



The ‘successor gadget’ used in this definition might appear a bit strange. As a matrix, it is in fact equal to the previously considered *triangle generator* in the ZX-calculus [40, 47]:

$$\begin{array}{c} \triangle \\ \uparrow \end{array} := \star \begin{array}{c} \ominus \\ \square \\ \circ \end{array}$$

This triangle has the following interpretation as a matrix:

$$\left[ \begin{array}{c} \triangle \\ \uparrow \end{array} \right] = |0\rangle\langle 0| + |1\rangle\langle 0| + |1\rangle\langle 1|$$

We could have introduced the triangle as an additional derived generator. However, as can be seen from its matrix, it is not self-adjoint and hence the orientation of the node matters. To avoid the complications that come from that, we elect not to use the triangle and to simply write out its form as a ZH-diagram as in (10).

$H(-1)$  is not the only H-box with a particularly simple representation. Indeed, we have

$$\begin{array}{c} \boxed{0} \\ | \end{array} := \begin{array}{c} \ominus \\ \star \\ \square \\ \ominus \\ \boxed{-1} \end{array} \stackrel{(Z)}{=} \begin{array}{c} \ominus \\ \star \\ \square \\ \circ \\ \ominus \end{array} \stackrel{2.28}{=} \begin{array}{c} \ominus \\ \circ \\ \ominus \end{array} \stackrel{2.12}{=} \begin{array}{c} \circ \\ \ominus \end{array} \stackrel{2.7}{=} \begin{array}{c} | \end{array} \quad (11)$$

$$\begin{array}{c} | \\ \boxed{1} \end{array} := \begin{array}{c} \ominus \\ \star \\ \square \\ \circ \\ \boxed{0} \end{array} \stackrel{11}{=} \begin{array}{c} \ominus \\ \star \\ \square \\ \circ \\ \ominus \end{array} \stackrel{2.26}{=} \begin{array}{c} \ominus \\ \star \\ \square \\ \circ \\ \circ \end{array} \stackrel{2.25}{=} \begin{array}{c} \ominus \\ \circ \\ \circ \end{array} \stackrel{(zs)}{=} \begin{array}{c} | \\ \circ \end{array} \stackrel{2.3}{=} \begin{array}{c} | \\ \circ \end{array} \quad (12)$$

**Definition 2.32.** For any  $a \in \mathbb{Z}$  such that  $a < -1$ , we define the corresponding degree-1 H-box by applying a negate spider:

$$\begin{array}{c} | \\ \boxed{a} \end{array} := \begin{array}{c} \ominus \\ \square \\ \boxed{-a} \end{array} \quad (13)$$

The negation gadget plays nicely with the H-box labels. In particular by Lemma 2.14, H-box labels satisfy  $-(-a) = a$ . We also have consistency with the simpler representations of  $H(a)$  for  $a \in \{-1, 0, 1\}$ :

$$\begin{array}{c} \ominus \\ | \\ \boxed{1} \end{array} \stackrel{(Z)}{\stackrel{(12)}{=}} \begin{array}{c} \circ \\ \square \\ \circ \end{array} \stackrel{(zs)}{=} \begin{array}{c} \circ \\ \square \\ \circ \end{array} \stackrel{(id)}{\stackrel{(9)}{=}} \begin{array}{c} | \\ \boxed{-1} \end{array}$$

$$\begin{array}{c} \ominus \\ | \\ \boxed{0} \end{array} \stackrel{(Z)}{\stackrel{(11)}{=}} \begin{array}{c} \circ \\ \square \\ \circ \end{array} \stackrel{2.21}{=} \begin{array}{c} \circ \\ \circ \\ \square \end{array} \stackrel{2.5}{\stackrel{(11)}{=}} \begin{array}{c} | \\ \boxed{0} \end{array}$$

**Definition 2.33.** Labelled H-boxes of arbitrary arity are defined as:

$$\begin{array}{c} \dots \\ \triangle \\ \boxed{a} \end{array} := \star \begin{array}{c} \dots \\ \square \\ \square \\ \boxed{a} \end{array} \quad (14)$$

This is consistent with (hh) for arity-1 H-boxes of arbitrary label (using Lemma 2.3).

Certain higher-arity H-boxes have relatively simple label-free forms, in particular, for the  $-1$ -labelled H-box we have:

$$\begin{array}{c} \dots \\ \triangle \\ \boxed{-1} \end{array} \stackrel{(14)}{=} \star \begin{array}{c} \dots \\ \square \\ \square \\ \boxed{-1} \end{array} \stackrel{(9)}{=} \star \begin{array}{c} \dots \\ \square \\ \square \\ \square \end{array} \stackrel{(hs)}{=} \begin{array}{c} \dots \\ \square \end{array} \quad (15)$$

for the 0-labelled H-box we have:

$$\begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \boxed{0} \end{array} \stackrel{(14)}{=} \star \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \square \\ \square \\ \boxed{0} \end{array} \stackrel{(11)}{=} \star \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \square \\ \square \\ \bullet \end{array} \stackrel{2.16}{=} \star \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \square \\ \square \\ \circ \end{array} \quad (16)$$

for the 1-labelled H-box we have:

$$\begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \boxed{1} \end{array} \stackrel{(14)}{=} \star \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \square \\ \square \\ \boxed{1} \end{array} \stackrel{(12)}{=} \star \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \square \\ \square \\ \circ \end{array} \stackrel{(X)}{=} \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \square \\ \square \\ \bullet \end{array} \stackrel{(ba_2)}{=} \circ \dots \circ \quad (17)$$

and for the 2-labelled H-box we have:

$$\begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \boxed{2} \end{array} \stackrel{(14)}{=} \star \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \square \\ \square \\ \boxed{2} \end{array} \stackrel{(10)}{=} \star \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \square \\ \square \\ \bullet \\ \square \\ \square \\ \boxed{1} \end{array} \stackrel{(17)}{=} \star \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \square \\ \square \\ \bullet \\ \square \\ \square \\ \circ \quad \circ \end{array} \quad (18)$$

There are some scalar cancellation rules associated to these labelled H-boxes:

**Lemma 2.34.** For any  $a \in \mathbb{Z}$ :

$$\begin{array}{c} \bullet \\ | \\ \boxed{a} \end{array} = \text{---} \text{---} \text{---} \text{---}$$

**Lemma 2.35.**

$$\boxed{2} = \circ$$

**Lemma 2.36.**

$$\begin{array}{c} \bullet \\ | \\ \boxed{2} \end{array} = \circ$$

*Proof of Lemma 2.34.* First, note that if  $a < -1$ , we can change the sign:

$$\begin{array}{c} \bullet \\ | \\ \boxed{-a} \end{array} \stackrel{(13)}{=} \begin{array}{c} \bullet \\ | \\ \circ \end{array} \stackrel{(Z)}{=} \begin{array}{c} \bullet \\ | \\ \square \end{array} \stackrel{2.21}{=} \begin{array}{c} \bullet \\ | \\ \square \end{array} \begin{array}{c} \bullet \\ | \\ \square \end{array} \stackrel{2.5}{=} \begin{array}{c} \bullet \\ | \\ \boxed{a} \end{array}$$

It thus suffices to consider  $-1 \leq a$ . For these values, we prove the result by induction, with the base case  $a = -1$  being Lemma 2.5. The inductive step is

$$\begin{array}{c} \bullet \\ | \\ \boxed{a+1} \end{array} \stackrel{(10)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \boxed{a} \end{array} \stackrel{(NOT)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \boxed{a} \end{array} \stackrel{2.10}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \boxed{a} \end{array} \stackrel{(hs)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \boxed{a} \end{array} \stackrel{2.3}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \boxed{a} \end{array} \stackrel{(X)}{=} \begin{array}{c} \bullet \\ | \\ \boxed{a} \end{array} \quad \blacksquare$$

*Proof of Lemma 2.35.*

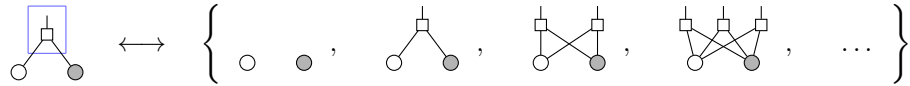
$$\boxed{2} \stackrel{(18)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{(Z)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{(Z)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{2.14}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{(X)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{2.26}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{(zs)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{2.3}{=} \circ \quad \blacksquare$$

*Proof of Lemma 2.36.*

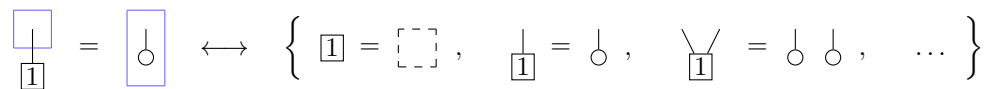
$$\begin{array}{c} \bullet \\ | \\ \boxed{2} \end{array} \stackrel{(18)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{(hh)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{2.12}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{2.26}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{(zs)}{=} \star \begin{array}{c} \bullet \\ | \\ \square \\ \square \\ \bullet \\ | \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \\ \square \end{array} \stackrel{2.3}{=} \circ \quad \blacksquare$$

## 2.6 !-box notation

Many of the calculations in the remainder of the paper are greatly simplified by the use of *!-box notation* [34]. A !-box (pronounced ‘bang box’) in a string diagram represents a part of the diagram that is able to fan out arbitrarily. That is, the contents of a !-box, along with any wires into or out of the !-box, can be copied  $n$  times for any non-negative integer  $n$ . For example, the !-box diagram below represents the following family of (concrete) string diagrams, one for each  $n$ :



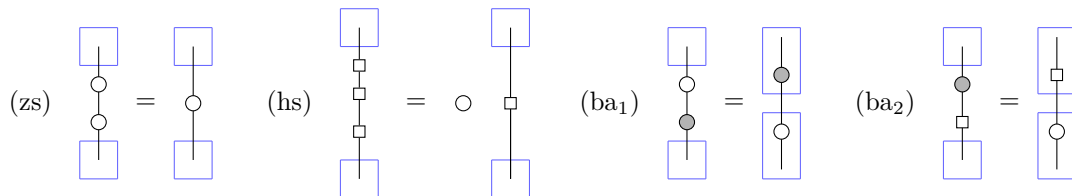
All of the resulting string diagrams are well-defined because all of our generators can have arbitrary arities. We can also use !-boxes in diagram equations, as long as each !-box on the LHS has a corresponding !-box on the RHS, and the inputs/outputs in each !-box match. Such a rule represents a family of equations where each *pair* of corresponding !-boxes is replicated  $n$  times, e.g. we can re-express (17) as:



Note the dashed box on the right-hand side of the first equation denotes an empty diagram. With this notation, the definition of grey spiders (1) becomes



Additionally, the rules (zs), (hs), (ba<sub>1</sub>), and (ba<sub>2</sub>) from Figure 1 become:



Note that the red dashed NOT-edges defined in (2) behave well when crossing !-box borders.

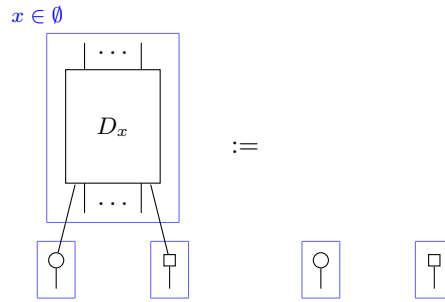


## 2.7 Annotated !-boxes

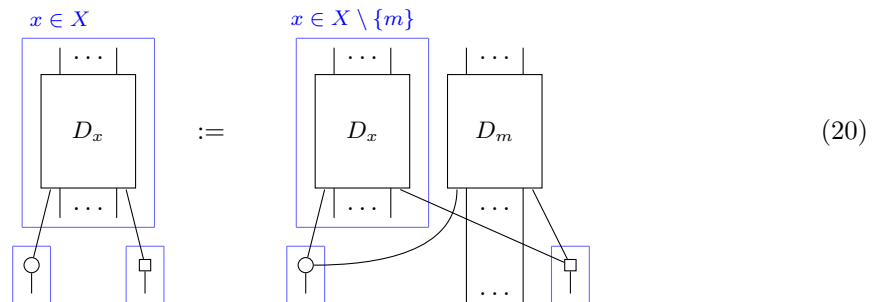
Many of our diagrams – in particular the normal forms that will be defined in Section 4.1 – have a repeating structure involving multiple H-boxes with different labels. To write such diagrams more concisely, we introduce a limited extension of the usual !-box notation, which allows !-boxes to be indexed by the elements of a totally ordered finite set. Standard !-boxes allow infinite families of diagrams to be represented in a single diagram, whereas annotated !-boxes simply offer a more concise representation of a single diagram with repeated structure, instantiated once for each element of the finite set.

**Definition 2.37.** An annotated !-box is a !-box with a label of the form ‘ $x \in X$ ’, where  $X$  is a totally ordered finite set and  $x$  is a variable that may appear in labels inside the !-box.

In the following, we will use a large box labelled  $D_x$  to denote an arbitrary ZH-diagram whose labels may contain the parameter  $x$ . Given a diagram containing an annotated !-box, we recursively define an equivalent diagram without the annotated !-box. The base case is that of a !-box indexed over the empty set, which is instantiated zero times:



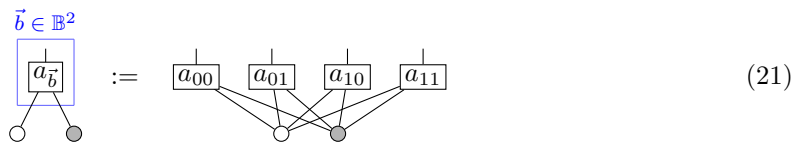
For the recursive case,  $X$  is non-empty. Thus it has a maximum element, which we denote  $m := \max X$ . We construct an equivalent diagram where the !-box is indexed by the smaller set  $X \setminus \{m\}$ . This is done by copying the contents of the !-box (including all external wires) once, immediately to the right of the !-box, and replacing each occurrence of  $x$  inside the new subdiagram by  $m$ :



**Remark 2.38.** This definition straightforwardly extends to diagrams containing multiple disjoint annotated !-boxes, which is all we need in this paper.

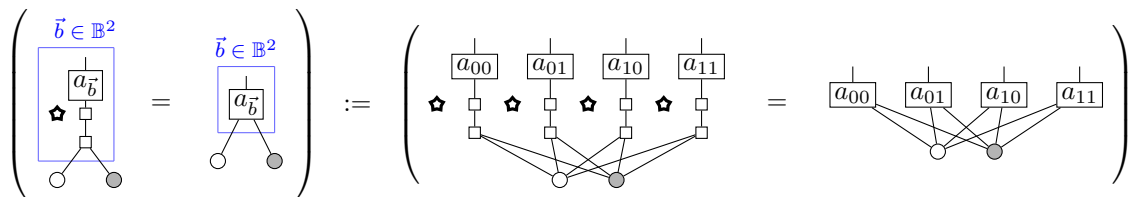
It can also be applied in reverse: any time a diagram contains multiple copies of the same subdiagram, up to changes in labels, these can be combined into an annotated !-box.

We will often use bit strings for indexing annotated !-boxes; the motivation for this will become clear with the introduction of normal-form diagrams in Section 4.1. As an example, indexing over the finite set  $\mathbb{B}^2 := \{00, 01, 10, 11\}$ , we can write expressions such as:



Since annotated !-boxes correspond to unique diagrams (rather than infinite families of diagrams), their appearance in equations is less constrained than that of unlabelled !-boxes. Nevertheless, some care is needed if the annotated !-boxes contain inputs or outputs of the diagram as a whole.

The simplest such equations correspond directly to allowed equations for unlabelled !-boxes: they have corresponding annotated !-boxes on the LHS and RHS, which are both indexed by the *same* finite set. Inputs and outputs coming out of a labelled !-box are matched to those of the same index on the other side of the equality. For example:

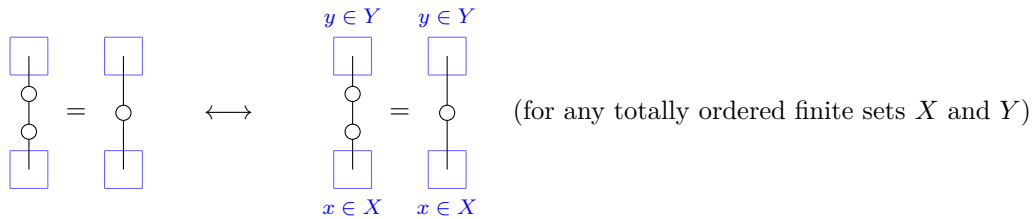


If an annotated !-box contains diagram inputs or outputs, it must either have a corresponding annotated !-box on the other side of the equality, or it must match a !-box labelled by a sub- or

superset (possibly the empty set) as in (20). This is to avoid complications caused by the need to keep the order of external wires consistent.

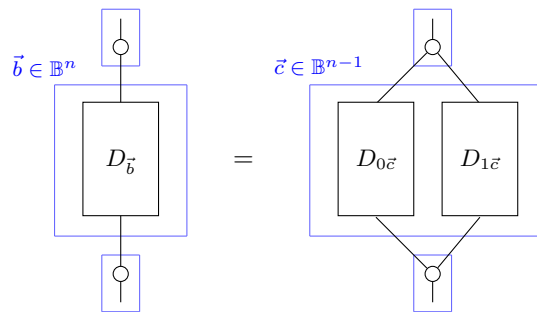
For annotated !-boxes that do not contain any diagram inputs or outputs, such ordering issues do not arise because of the meta rule ‘only topology matters’. Lemmas 2.39 and 2.40 below give examples of diagram equations where the annotated !-boxes on the two sides do not match in the same strict way.

Note that we can recover the behaviour of normal, un-labelled !-boxes by interpreting a !-box without a label as being indexed by an *arbitrary* totally ordered finite set, e.g.



In particular, this means that any derived rewrite rule involving un-labelled !-boxes can also be applied to annotated !-boxes with arbitrary labels.

**Lemma 2.39.** An annotated !-box indexed by a bit string can be ‘expanded’ according to one of the bits, as long as it does not contain any inputs or outputs of the diagram as a whole. Here, the box labelled  $D_{\vec{b}}$  denotes an arbitrary diagram parameterised by  $\vec{b}$ .

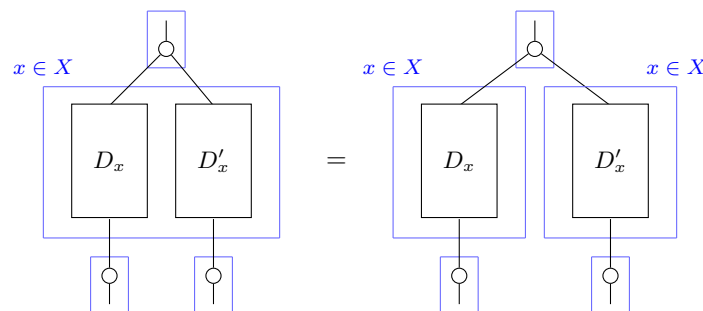


*Proof.* Note that the set  $\mathbb{B}^n$  can be split into two pieces, based on whether the most significant bit is 0 or 1:

$$\mathbb{B}^n = \{0\vec{c} \mid \vec{c} \in \mathbb{B}^{n-1}\} \uplus \{1\vec{c} \mid \vec{c} \in \mathbb{B}^{n-1}\}$$

The result therefore follows by completely expanding the annotated !-boxes on each side according to Definition 2.37 and applying the meta-rule ‘only topology matters’. ■

**Lemma 2.40.** An annotated !-box containing two disconnected diagram components can be split into two boxes indexed over the same set, as long as the original annotated !-box does not contain any inputs or outputs of the diagram as a whole. Here,  $X$  is an arbitrary totally ordered finite set and the boxes labelled  $D_x$  and  $D'_x$  denote arbitrary diagrams parameterised by  $x$ .



*Proof.* Note that for any set  $X$ ,

$$\bigsqcup_{x \in X} \{D_x, D'_x\} = \{D_x \mid x \in X\} \uplus \{D'_x \mid x \in X\}$$

The result therefore follows from Definition 2.37 and the meta-rule ‘only topology matters’. ■

The condition about annotated !-boxes in Lemmas 2.39 and 2.40 not containing any inputs or outputs of the diagram as a whole is to avoid issues caused by the need to keep the order of external wires consistent. We have drawn the contents of the annotated !-boxes as being connected only to Z-spiders, since this covers all our applications. Nevertheless, the result generalises straightforwardly to X-spiders and H-boxes as well.

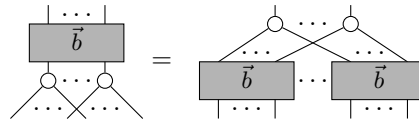
The following notation will be useful when working with annotated !-boxes.

**Definition 2.41.** Let  $\mathbb{B}^n$  be the set of all  $n$ -bitstrings. For any  $\vec{b} := b_1 \dots b_n \in \mathbb{B}^n$ , define the indexing map  $\iota_{\vec{b}}$  as follows:

$$\iota_{\vec{b}} = \boxed{\vec{b}} = \left( \begin{array}{c} | \\ \bullet \\ | \end{array} \right)^{1-b_1} \dots \left( \begin{array}{c} | \\ \bullet \\ | \end{array} \right)^{1-b_n},$$

where  $\left( \begin{array}{c} | \\ \bullet \\ | \end{array} \right)^1 = \begin{array}{c} | \\ \bullet \\ | \end{array}$  and  $\left( \begin{array}{c} | \\ \bullet \\ | \end{array} \right)^0 = |$ , analogous to how  $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ .

**Lemma 2.42.** The  $\iota_{\vec{b}}$  operator copies through white spiders, i.e. for any  $\vec{b} \in \mathbb{B}^n$ :



*Proof.* This follows immediately from Lemma 2.19 via Definition 2.41. ■

### 3 Special cases of multiply, intro and average

Now that we have all the necessary definitions and basic rewrite rules, we can start our proof of completeness. We will proceed as outlined in Section 1.2, first introducing three families of rewrite rules that use the labelled H-boxes of Section 2.5, which will play an important part in proving completeness. Proving that these rewrite rules hold in full generality is difficult however. In this section we will only prove these rewrite rules for certain small integers. The general proofs are postponed until Section 6, after we have introduced and studied the normal form diagrams in Section 4.

Note that starting from here, we will no longer necessarily write all the lemmas used in a rewrite step in a proof. In particular, we will often suppress uses of Lemma 2.3 and just treat  $\star$  and  $\circ$  as each other’s inverses. We will often also let uses of spider fusion (zs) be implicit.

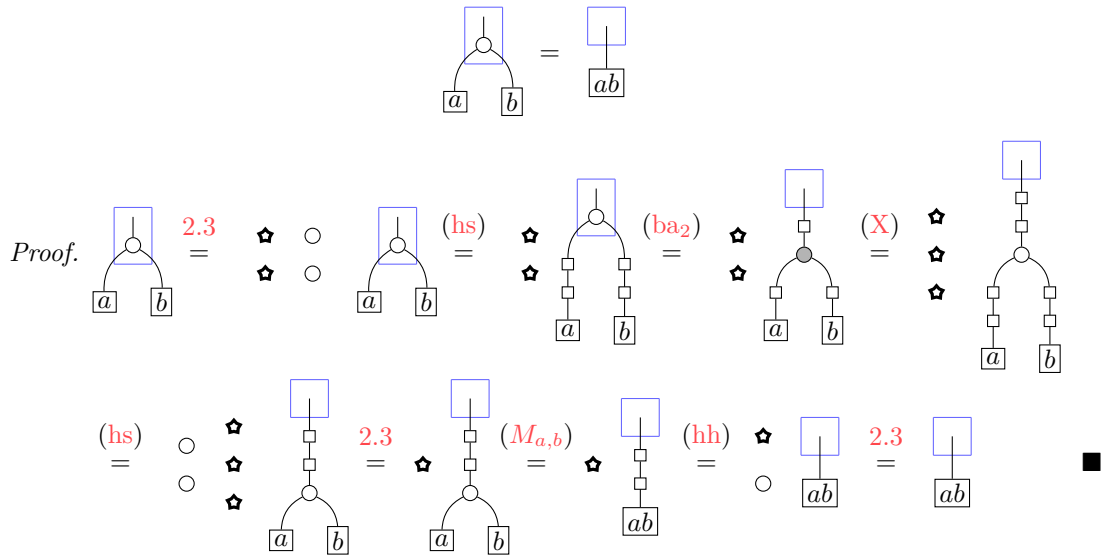
#### 3.1 The multiply rule

We claim that the following identity holds in the ZH-calculus for all integers  $a$  and  $b$ :

$$\begin{array}{c} | \\ \circ \\ \swarrow \quad \searrow \\ \boxed{a} \quad \boxed{b} \end{array} = \boxed{ab} \tag{M_{a,b}}$$

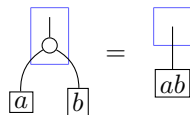
We refer to this as the *multiply* rule. Proving this is quite involved and will be postponed until Section 5. The usefulness of this rule comes from the following generalisation to arbitrary arity.

**Lemma 3.1.** If the ZH-calculus proves  $M_{a,b}$ , then it also proves the following:

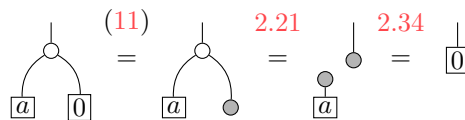


This states that if two labelled H-boxes are connected to exactly the same set of white spiders that we can fuse the H-boxes together by multiplying their labels. For certain small values of  $b$  we can easily prove this rule.

**Lemma 3.2.** Let  $b \in \{0, 1, -1\}$ . Then the ZH-calculus proves  $(M_{a,b})$  for all  $a \in \mathbb{Z}$ .



*Proof.* By Lemma 3.1 it suffices to show the case where the !-box is expanded a single time. For  $b = 1$ , this follows from (17) and (zs). For  $b = -1$ , this follows from (Z) and (13). For  $b = 0$  we calculate:



### 3.2 The intro rule

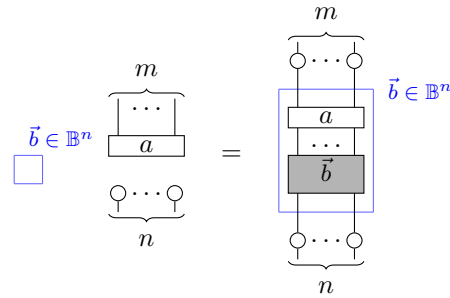
A second family of rules that is particularly useful is the following:



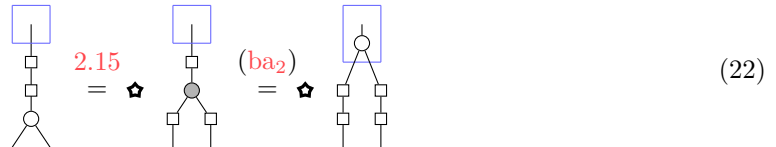
We refer to this as the *intro* rule as it allows us to introduce new wires to an H-box. As is the case for the multiply rule, this rule has a generalisation to arbitrary arity. But unlike the multiply rule, it will often be useful to apply the rule multiple times in succession to connect it to a larger set of wires.

**Lemma 3.3.** If the ZH-calculus proves  $I_a$ , then it also proves the following !-boxed version<sup>3</sup>. The natural numbers  $m$  and  $n$  are arbitrary, but note that the annotated !-box depends on  $n$ .

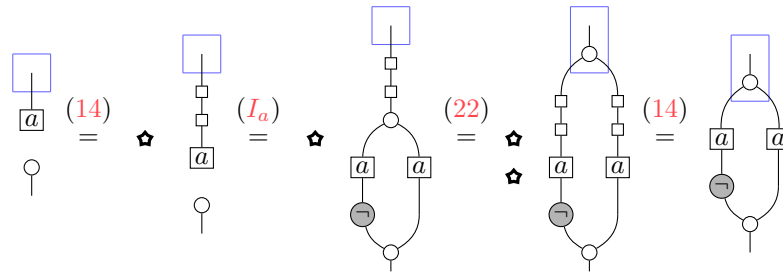
<sup>3</sup>We use a mixed notation without !-boxes for the inputs and outputs for easier intelligibility, but will occasionally use the more formal notation with further !-boxes where the extra rigour can be achieved without confusing the notation.



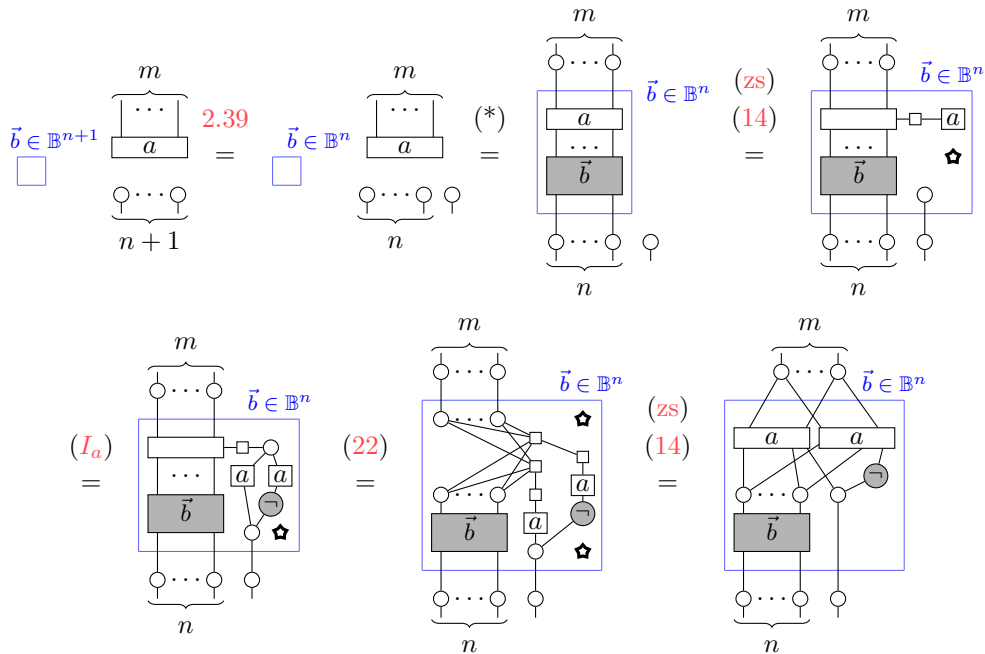
*Proof.* First, let us prove the following equation:



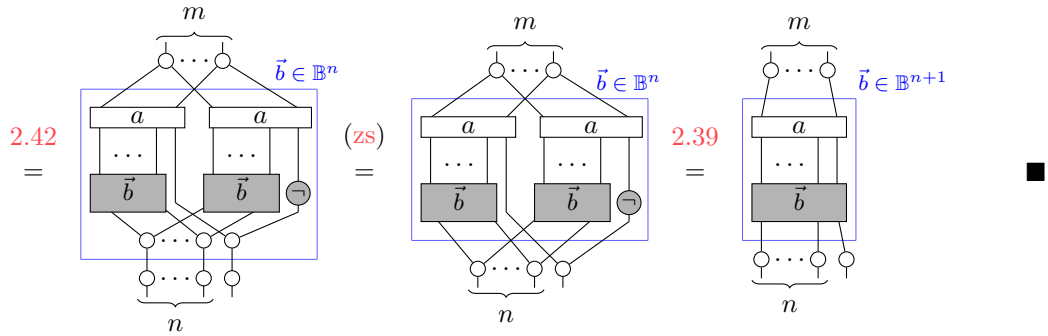
Now, we prove the claim by induction on  $n$ , beginning with the base case where  $n = 1$ . Note that  $n = 0$  is valid but trivial. In the derivation below we use a  $!$ -box to show that  $m$  is arbitrary.



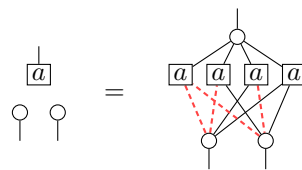
For the inductive step, assume the lemma holds for  $n$  copies of the lower white node. In the calculation below,  $(*)$  denotes the induction step.







**Example 3.4.** When applying the !-boxed intro rule to an H-box and  $n$  white spiders, the H-box is copied  $2^n$  times, with each copy being connected to the  $n$  white spiders via a different combination of NOT edges and plain edges. Every wire originally incident on the H-box is replaced by a white spider, which connects to all the new H-boxes. For example, with  $n = 2$  and a single wire incident on the original H-box:



This corresponds to the outer product

$$\begin{pmatrix} 1 \\ a \end{pmatrix} ((1 \ 1) \otimes (1 \ 1)) = \begin{pmatrix} 1 \\ a \end{pmatrix} (1 \ 1 \ 1 \ 1) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ a & a & a & a \end{pmatrix},$$

as will become clear from the normal form definition in Section 4.1.

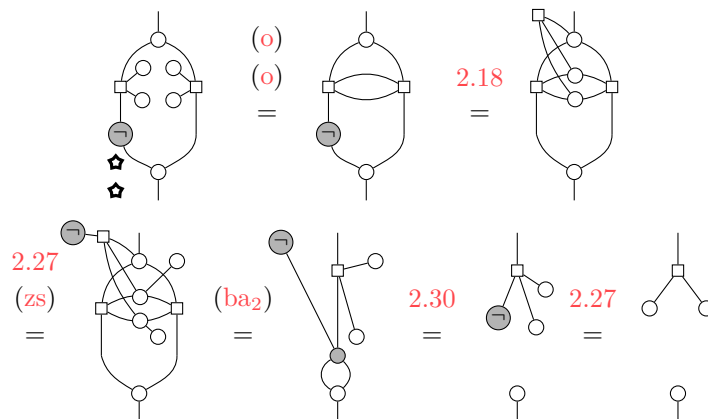
We will prove that the intro rule holds for all integers in Section 6. But first we will directly prove the intro rule for certain simple integers. To do this we will need two lemmas.

**Lemma 3.5.**

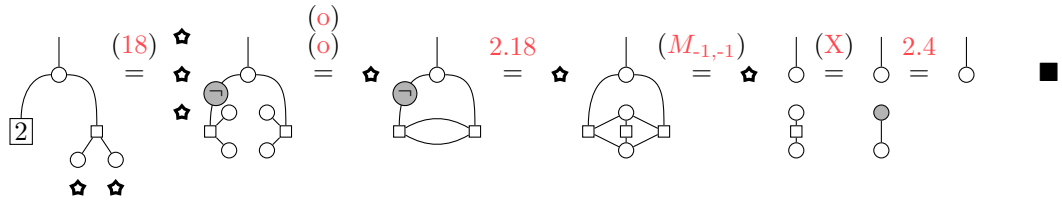
**Lemma 3.6.**



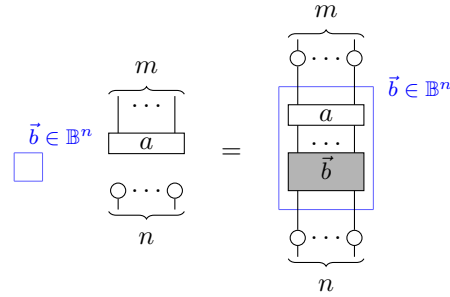
*Proof of Lemma 3.5.* Going from right to left:



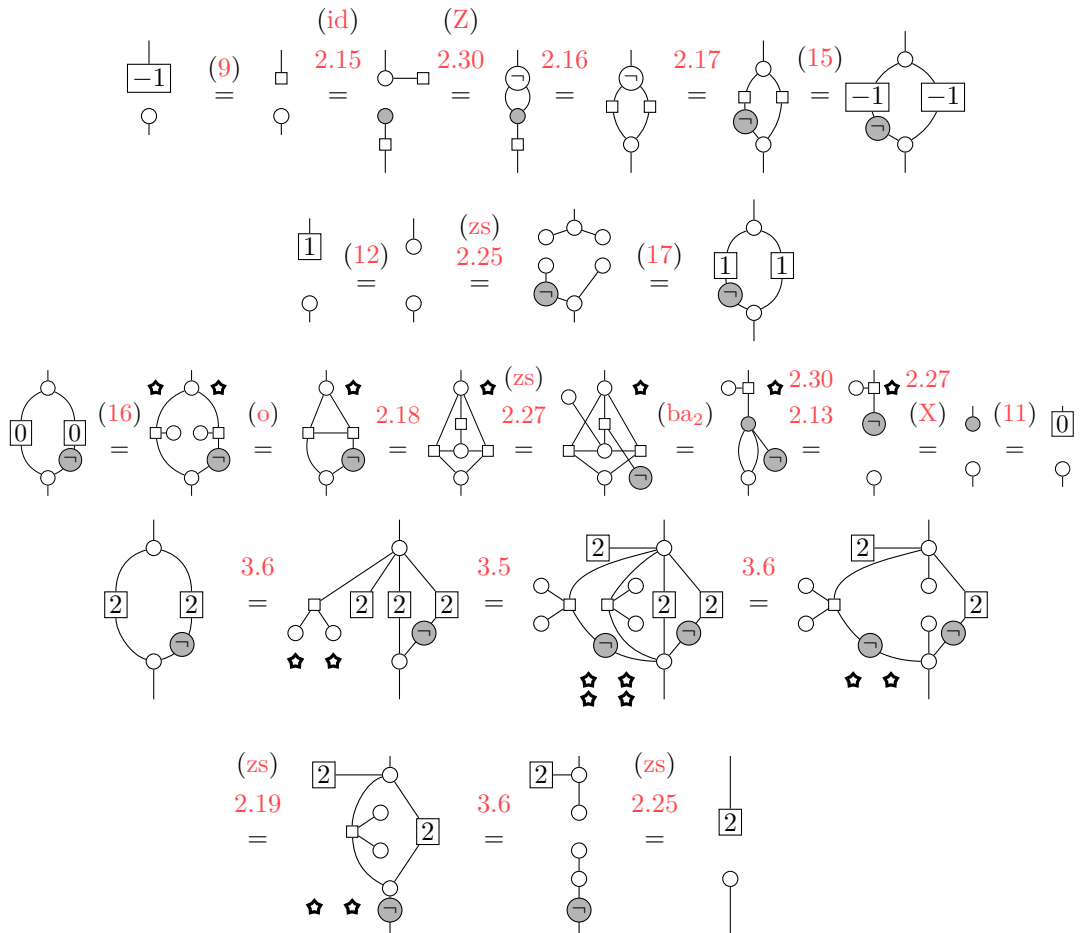
*Proof of Lemma 3.6.* We prove the !-box-free version, with the generalisation following the same proof as for Lemma 3.1.



**Lemma 3.7.** Let  $a \in \{-1, 0, 1, 2\}$ . Then the ZH-calculus proves  $I_a$  and its !-boxed generalisation:



*Proof.* By Lemma 3.3 it suffices to show Eq.  $(I_a)$ . We prove by case distinction on  $a \in \{-1, 0, 1, 2\}$ :



### 3.3 The average rule

The final family of rewrite rules we will need is the following.

$$\text{Diagram} = \text{Diagram} \quad (A_{a,b})$$

We call this rule the *average* rule, because on the right-hand side it adds the numbers together, but also multiplies it with a ‘NOT 2’ H-box, that acts like multiplying by a half; cf. Lemma 8.7. We will prove that this rule holds for all integers in Section 6. For now, we will prove it for certain combinations of integers.

**Lemma 3.8.** If the ZH-calculus proves  $(A_{a,b})$ , then it also proves the following:

$$\text{Diagram} = \star \text{Diagram} \quad (A_{a,b})$$

*Proof.*

$$\text{Diagram} \stackrel{(hs)}{=} \star \text{Diagram} \stackrel{(ba_2)}{=} \star \text{Diagram} \stackrel{(X)}{=} \star \text{Diagram} \stackrel{(hs)}{=} \star \text{Diagram} \stackrel{(A_{a,b})}{=} \star \text{Diagram} \quad \blacksquare$$

**Lemma 3.9.** The ZH-calculus proves  $(A_{a,0})$  for all  $a \in \mathbb{Z}$ .

$$\text{Diagram} = \text{Diagram}$$

*Proof.*

$$\text{Diagram} \stackrel{(14)}{=} \text{Diagram} \stackrel{(id)}{=} \star \text{Diagram} \stackrel{(I_0)}{=} \text{Diagram} \stackrel{(M_{0,-1})}{=} \text{Diagram} \stackrel{(I_1)}{=} \text{Diagram} \stackrel{(I_0)}{=} \text{Diagram} \stackrel{(id)}{=} \text{Diagram} \stackrel{(hh)}{=} \text{Diagram} \stackrel{(2.12)}{=} \text{Diagram} \stackrel{(16)}{=} \text{Diagram} \stackrel{(18)}{=} \text{Diagram} \quad \blacksquare$$

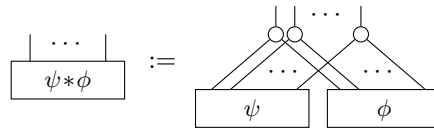
**Lemma 3.10.** Let  $a$  be an integer for which we have proven  $(I_a)$ . Then we can prove  $(A_{a,-a})$ .



### 4.1 Normal form diagrams

Our normal form diagrams will be ‘state-like’ in that all of the wires will be connected to the top of the diagram. This is because we can convert between states and operators by simply bending inputs to become outputs and vice versa. This transforming of an operator into a state is known as map-state duality or the Choi-Jamiołkowski isomorphism.

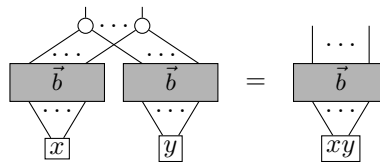
A key part of this normal form, is that we can easily represent the *Schur product* of two states, i.e. the pointwise product, in the ZH-calculus. For states  $\psi, \phi$ , write  $\psi * \phi$  for their Schur product. Then representing states by large white boxes we have the following identity:



Here the  $i$ -th output of  $\psi$  and  $\phi$  is plugged into a  $\circlearrowleft$  for each  $i$ . It follows from (zs) that  $*$  is associative and commutative, so we can write  $k$ -fold Schur products  $\psi_1 * \psi_2 * \dots * \psi_k$  without ambiguity. For any finite set  $J$  with  $|J| = k$ , let  $\prod_{j \in J} \psi_j$  be the  $k$ -fold Schur product.

As shown in the next lemma, the indexing maps  $\iota_{\vec{b}}$  from Definition 2.41 interact nicely with the Schur product.

**Lemma 4.1.** The ZH-calculus enables the computation of the Schur product of two maps of the form  $\iota_{\vec{b}} \circ H_n(x)$  and  $\iota_{\vec{b}} \circ H_n(y)$  for any  $\vec{b} \in \mathbb{B}^n$  and  $x, y$  for which we have proven  $(M_{x,y})$ :



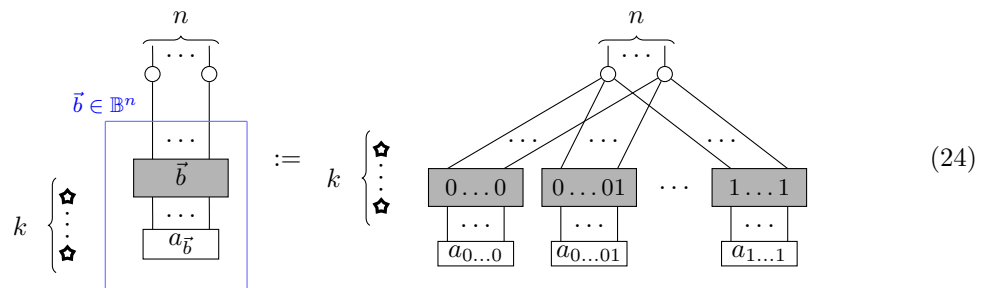
*Proof.* Apply Lemma 2.42, followed by Lemma 3.1. ■

**Definition 4.2.** A *normal-form diagram* in the ZH-calculus is a diagram of the form

$$\star^k \prod_{\vec{b} \in \mathbb{B}^n} (\iota_{\vec{b}} \circ H_n(a_{\vec{b}})) \tag{23}$$

where  $H_n(a_{\vec{b}})$  is the arity- $n$  H-box (considered as a state) labelled by some values  $a_{\vec{b}}$  for all  $\vec{b} \in \mathbb{B}^n$ , and  $k \in \mathbb{N}$ . We say the normal-form is *reduced* when either  $k = 0$  or at least one of the labels  $a_{\vec{b}}$  cannot be factored by two (i.e. is odd).

A normal form diagram can be seen as a collection of  $n$  spiders, fanning out to  $2^n$  H-boxes, each with a distinct configuration of NOT's corresponding to the  $2^n$  bitstrings in  $\mathbb{B}^n$ , together with a global scalar. Diagrammatically, normal forms are:



**Remark 4.3.** Note that the number of outputs in a normal form diagram is the same as the exponent of  $\mathbb{B}$  in the !-box annotation. To avoid cluttering the notation too much, we will only rarely make this explicit in diagrams.

The additional condition for being reduced ensures uniqueness of normal form diagrams.

**Theorem 4.4.** Reduced normal forms are unique. In particular:

$$\left[ \star^k \prod_{\vec{b} \in \mathbb{B}^n} (\iota_{\vec{b}} \circ H_n(a_{\vec{b}})) \right] = \sum_{\vec{b} \in \mathbb{B}^n} \frac{1}{2^k} a_{\vec{b}} |\vec{b}\rangle. \tag{25}$$

*Proof.* The map  $\iota_{\vec{b}}$  is a permutation that acts on computational basis elements as  $|\vec{c}\rangle \mapsto |\vec{c} \oplus \vec{b} \oplus \vec{1}\rangle$ . In particular, it sends the basis element  $|\vec{1}\rangle$  to  $|\vec{b}\rangle$ . Hence  $\iota_{\vec{b}} \circ H_n(a_{\vec{b}})$  is a vector with  $a_{\vec{b}}$  in the  $\vec{b}$ -th component and 1 everywhere else. The Schur product of all such vectors indeed gives the RHS of (25) up to the global scalar of  $\frac{1}{2^k}$  that is added by  $\star^k$ .

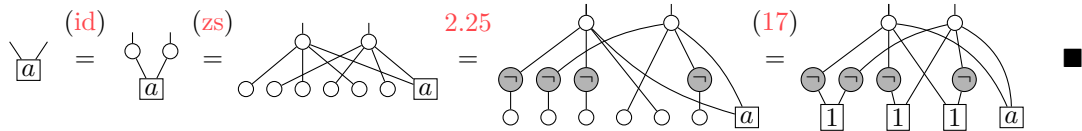
So now suppose two different normal form diagrams have equal interpretation (25). We need to show that the diagrams are then equal, which boils down to showing  $k = k'$  and  $a_{\vec{b}} = a'_{\vec{b}}$  for all  $\vec{b} \in \mathbb{B}^n$ . Suppose without loss of generality that  $k \leq k'$ . Then  $a_{\vec{b}} = \frac{2^k}{2^{k'}} a'_{\vec{b}}$ . The left-hand side is an integer, and so the right-hand side must be so as well. Hence, if  $k < k'$ , then all the  $a'_{\vec{b}}$  must be divisible by 2, which contradicts the assumption of being a reduced normal form (since in this case  $k' \neq 0$ ). Hence,  $k = k'$ , and hence  $a_{\vec{b}} = a'_{\vec{b}}$  as required. ■

Because reduced normal forms are unique, two diagrams in reduced normal form are equal if and only if the linear maps the diagrams represent are equal, and hence, the calculus is complete if we can bring all diagrams to reduced normal form.

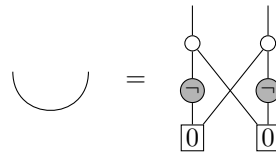
Some simple diagrams can readily be brought to reduced normal form:

**Lemma 4.5.** Any H-box can be brought into normal form using the rules of the ZH-calculus.

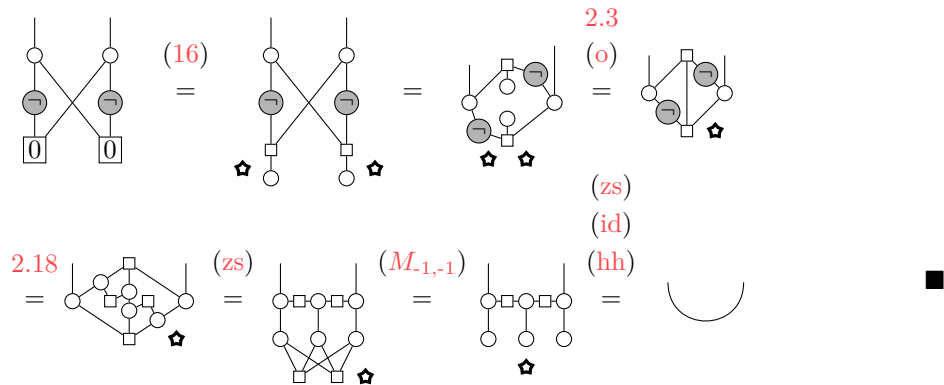
*Proof.* The matrix of an H-box  $H_n(a)$  has 1's in every entry but the very last one. Hence, to bring an H-box into normal form, we just need to introduce 'dummy' 1's for every other matrix entry. We demonstrate the principle using a binary H-box but the argument is analogous for any other arity:



**Lemma 4.6.**



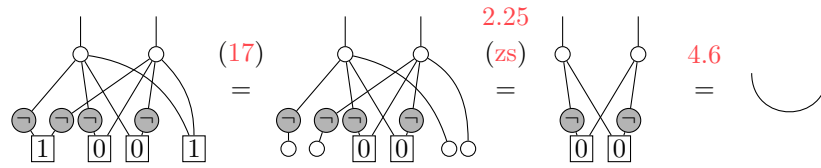
*Proof.*



**Lemma 4.7.** The diagram of a single cup can be brought into normal form:

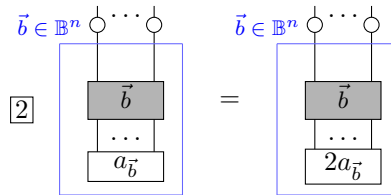


*Proof.* Starting from the normal form we work our way back:

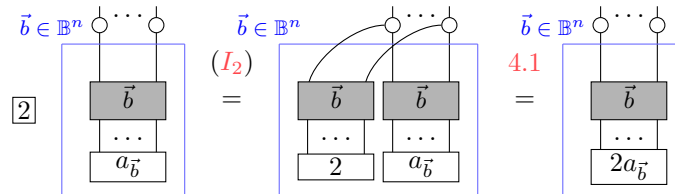


Before we continue to the more general procedure to reduce diagrams to normal form we have to prove some results about how to deal with scalars and factors of 2 in normal forms in order to reduce the diagram.

**Lemma 4.8.** Given integers  $a_{\vec{b}}$  for which we have proven  $(M_{2,a_{\vec{b}}})$  we can prove:



*Proof.*



**Proposition 4.9.** Suppose we have proven  $(M_{2,a})$  for all integers  $a \in \mathbb{Z}$ . Given a diagram in normal form, we can bring it to reduced normal form.

*Proof.* If there are no  $\star$ 's present we are done. Furthermore, if any of the  $a_{\vec{b}}$  is odd, we are also done. So suppose there is at least one  $\star$  and that all the  $a_{\vec{b}}$  are even. Then we can factor them all as  $2 \frac{a_{\vec{b}}}{2}$  and apply Lemma 4.8 in reverse to introduce a scalar 2 to cancel a star (using Lemma 2.35 and Lemma 2.3). We keep repeating this procedure until either the  $\star$ 's run out, or one of the  $a_{\vec{b}}$  becomes odd. ■

## 4.2 Conditional reduction to normal form

For bigger diagrams we will show that if a part of the diagram is in normal form, we can ‘consume’ the rest of the diagram generator by generator to write the entire diagram into normal form.

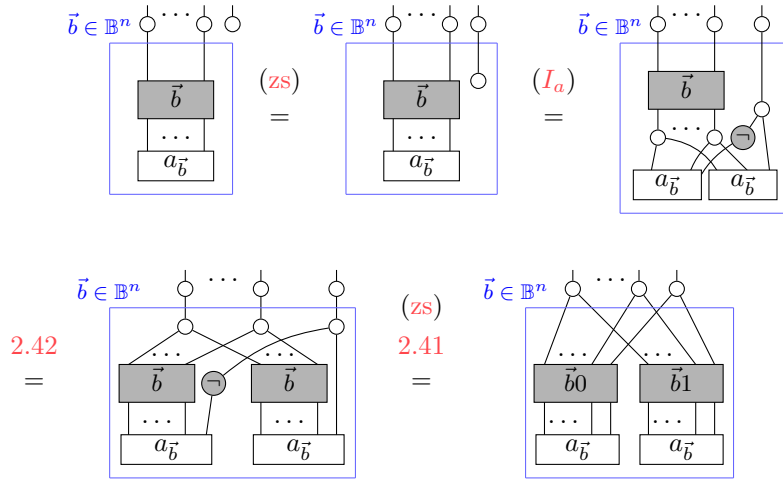
This procedure requires that we have proven the multiply, average and intro rule for all integers. Since this is difficult to do, we will first show the general procedure for bringing a diagram to normal form *assuming* we have proven these identities. We will then be able to ‘bootstrap’ the proof for the general multiply, average and intro rule using the simple cases of these rules we have already proven.

We need to show that any generator of the ZH-calculus — H-boxes, Z-spiders, and any type of wiring — can be brought to normal form, that tensor products of normal forms can be brought to normal form, and that any diagram consisting of a normal form composed with a generator in some way can also be brought to normal form. Once we have this we know that any ZH-diagram can be brought to normal form, and hence, if two ZH-diagrams represent the same linear map they must then be equal to the same ZH-diagram.

**Proposition 4.10.** Let  $D$  be a diagram consisting of a normal form diagram for which we have proven  $(I_a)$  for every  $a$  in the normal form, juxtaposed with  $\circlearrowleft$ . Then  $D$  can be brought into normal form using the rules of the ZH-calculus:

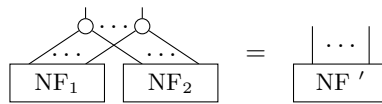
$$\overbrace{\text{NF}}^n \circlearrowleft = \overbrace{\text{NF}' }^{n+1}$$

*Proof.* Starting from the left-hand side, which we expand using the indexed !-box notation, we calculate:



The last diagram is a normal form diagram with  $n + 1$  outputs, i.e. the desired result. ■

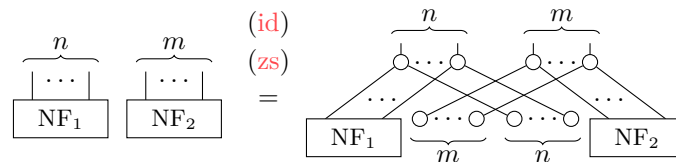
**Proposition 4.11.** The Schur product of two normal form diagrams can be brought into normal form using the rules of the ZH-calculus, when we have proven  $(M_{a,b})$  for every pair  $a$  and  $b$  occurring in the normal forms.



*Proof.* This follows from  $(zs)$  and Lemma 4.1. ■

**Corollary 4.12.** The tensor product of two normal form diagrams can be brought into normal form using the rules of the ZH-calculus, when we have proven  $(I_a)$  for all  $a$  occurring in the normal forms, and  $(M_{a,b})$  for all  $a$  and  $b$  in the normal forms.

*Proof.* A tensor product can be expressed as

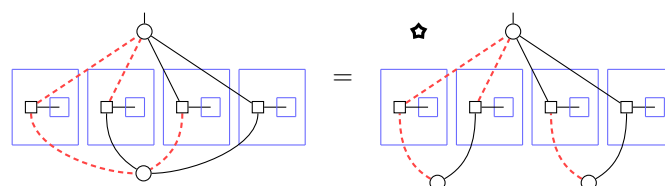


The diagram  $NF_1$  and the leftmost  $m$  copies of  $\circlearrowleft$  can be combined into one normal-form diagram with  $(n + m)$  outputs by successive applications of Proposition 4.10. Similarly, the rightmost  $n$  copies of  $\circlearrowleft$  and  $NF_2$  can be combined into one normal-form diagram with  $(n + m)$  outputs. The desired result then follows by Proposition 4.11. ■

The most difficult step is to show that contraction with a white dot preserves normal forms. This requires a couple of lemmas.

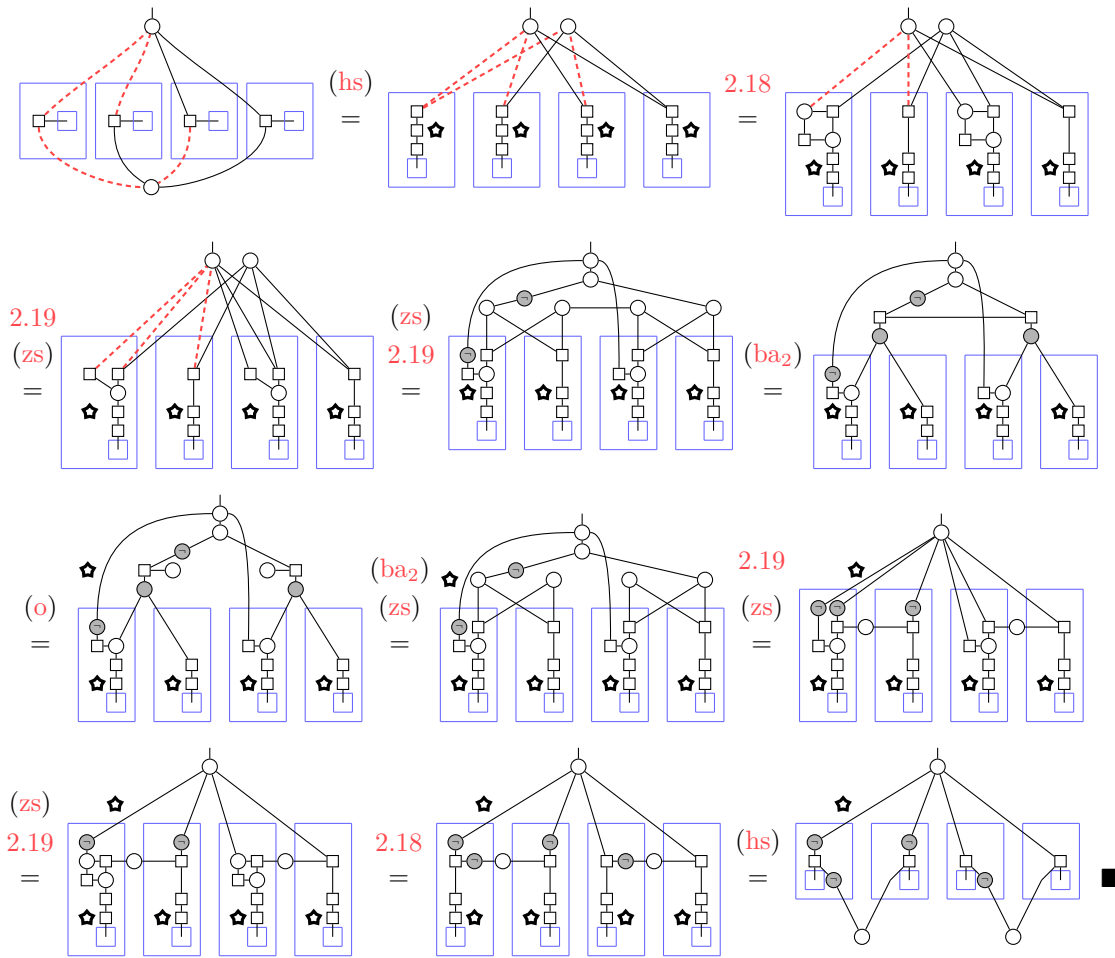
First, we present a more general form of the  $(o)$  rule that allows us to disconnect wires in a more general way.

**Lemma 4.13.** Instead of just wires, white spiders between an arbitrary number of H-boxes can be split using  $(o)$ :





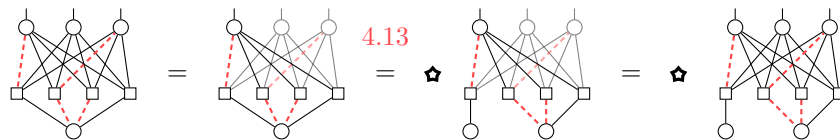
*Proof.*



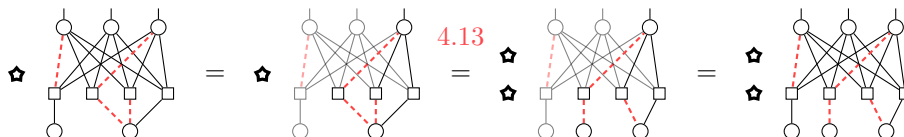
In practice, Lemma 4.13 is usually applied in diagrams that are close to normal form, so that the H-boxes are connected to multiple white spiders. This allows the lemma to be applied multiple times in succession.

**Example 4.14.** Consider the following derivation, which will arise in the proof of Proposition 6.2. For each rewrite step, the parts of the diagram that are uninvolved have been greyed out to clarify the process.

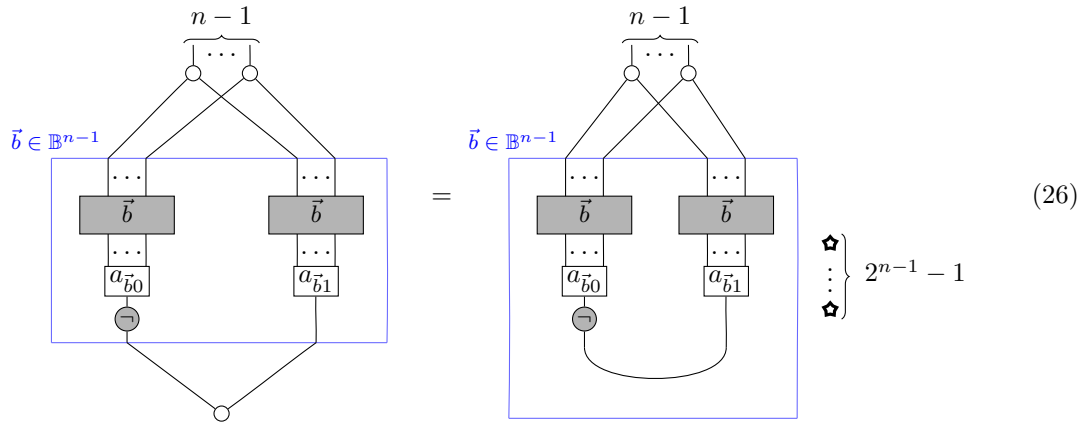
In the first application of Lemma 4.13, the top leftmost spider is the important one. The rewrite step splits the bottom spider to separate the first H-box from the others, as the first H-box is the only one that is connected to the leftmost spider via an X node:



Now, applying the lemma to the middle spider would have no effect as all H-boxes are connected to the middle spider via the same kind of wire (namely, simple ones). Thus, it remains to apply Lemma 4.13 according to the top rightmost spider. This step splits the bottom spider to separate the second H-box from the others because the second H-box is connected to the rightmost spider via an X node while the others are connected by simple wires:

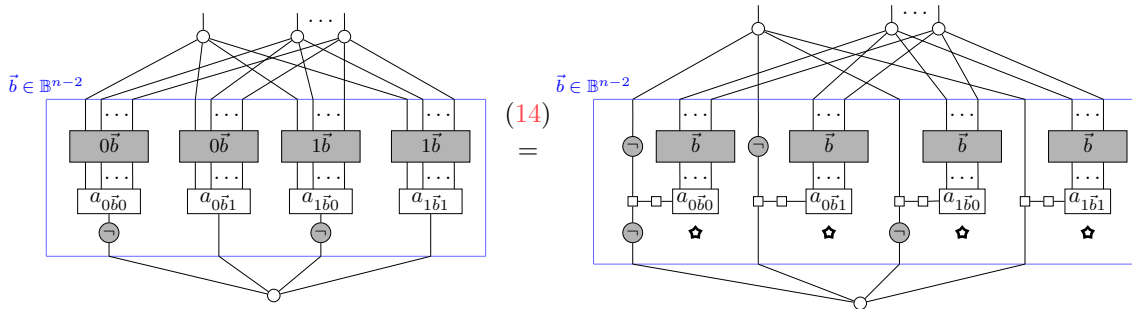


**Lemma 4.15.** Disconnect lemma:<sup>4</sup>

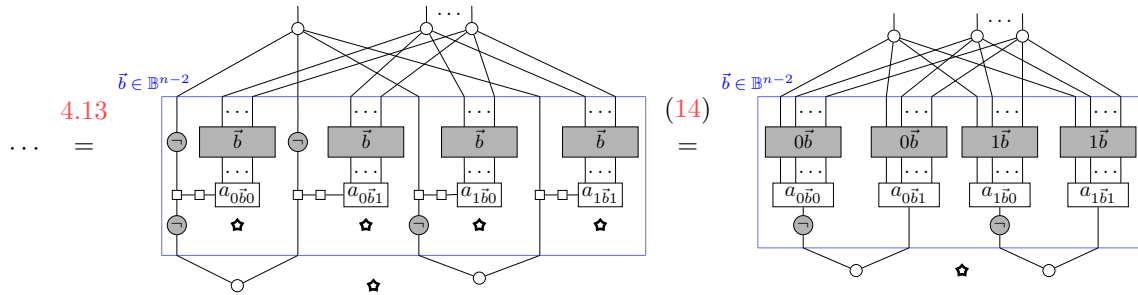


*Proof.* If  $n = 1$ , the  $\boxplus$ -boxes on both sides are indexed over the one-element set, so the result follows straightforwardly from the definition of annotated  $\boxplus$ -boxes in Section 2.7 and removal of the white spider via (id). Note that there are zero stars in this case.

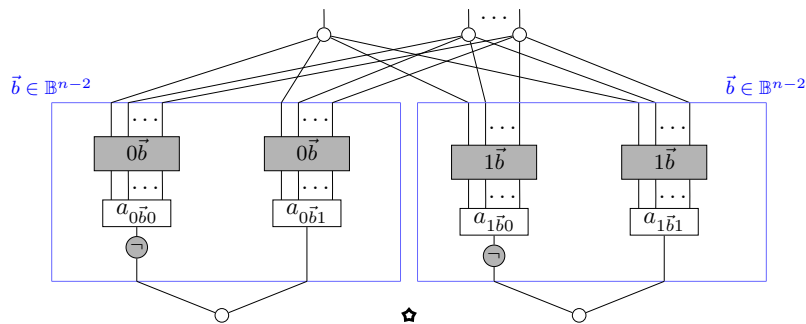
For  $n > 1$ , by Lemma 2.39, we can equivalently express the LHS as



We can then apply Lemma 4.13 to split the bottom spider in two:



By Lemma 2.40, we can then split the  $\boxplus$ -box into two parts, indexed over the same set to obtain:



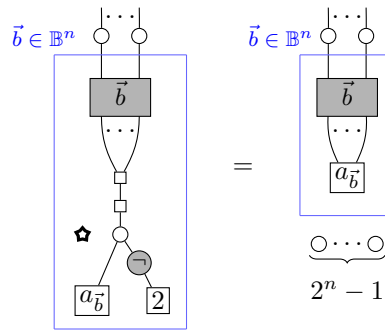
<sup>4</sup>A similar but incorrectly scaled result is given as Lemma B.3 in [5]. Since any application of the lemma in that paper is always combined with an also incorrectly-scaled application of the average rule, the scalars work out there overall anyway. We give the correct scalar factors here.

We now have two copies of a graph which is very similar to the LHS of (26), but for one fewer bit. We can thus repeat the process above to split each of the two spiders using the second bit of the bitstring, then the third, and so on, until  $\circ$ -spiders only connect pairs of  $H$ -spiders that disagree on the least significant bit.

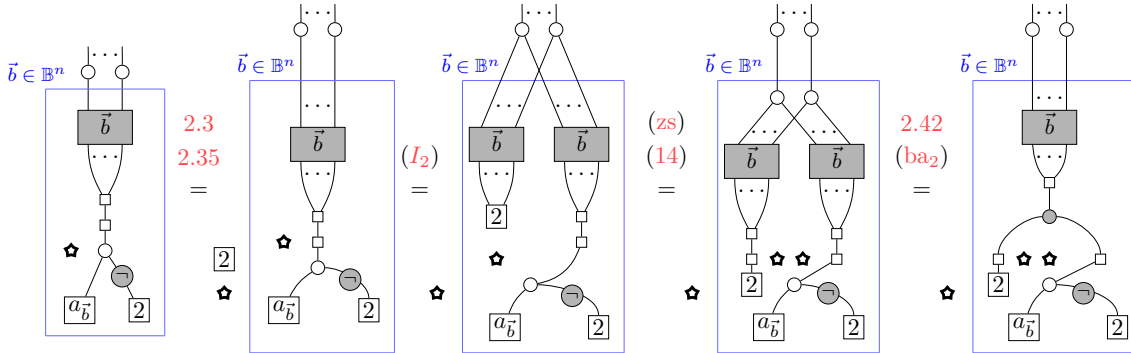
Each application of Lemma 4.13 introduces a star, so we need to count how many times that lemma is used in total. For the first bit of  $\mathbb{B}^{n-1}$ , the lemma is applied once. Afterwards, the diagram has separated into two indexed  $!$ -boxes and Lemma 4.13 needs to be applied to each part. Continuing in this way, the number of  $!$ -boxes which need to be split doubles in each step. Hence we introduce  $1 + 2 + \dots + 2^{n-2} = 2^{n-1} - 1$  stars.

Note that at the end of this process, each  $!$ -box will be indexed over a one-element set, so according to the definition of annotated  $!$ -boxes we can simply drop them. By replacing 2-legged  $\circ$ -spiders with cups using (id), and applying the definition of annotated  $!$ -boxes to re-introduce indexing over  $\mathbb{B}^{n-1}$ , we obtain the RHS of (26). ■

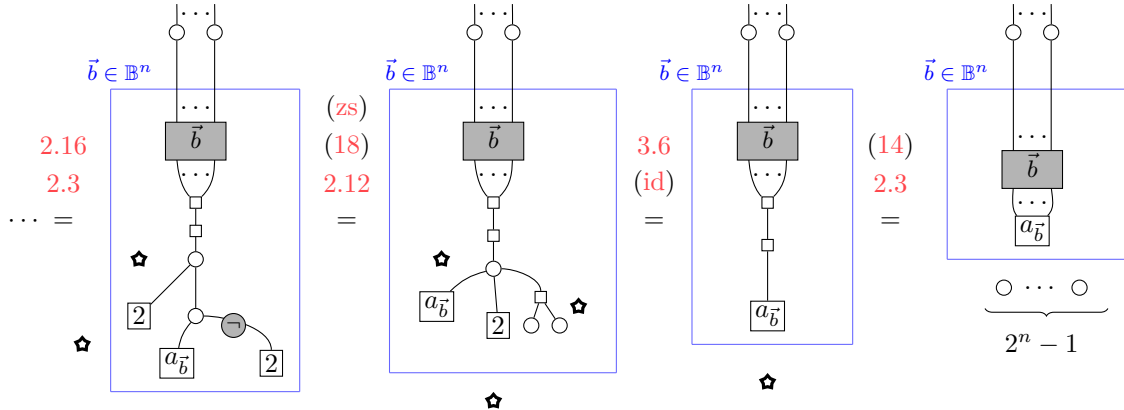
**Lemma 4.16.** For any  $a_{\vec{b}}$  we have



*Proof.* First, we introduce a scalar  $H(2)$  and get this in the  $!$ -box using Lemma 3.3 (the  $!$ -boxed form of the Intro rule):



And then we prepare the diagram to apply Lemma 3.6:

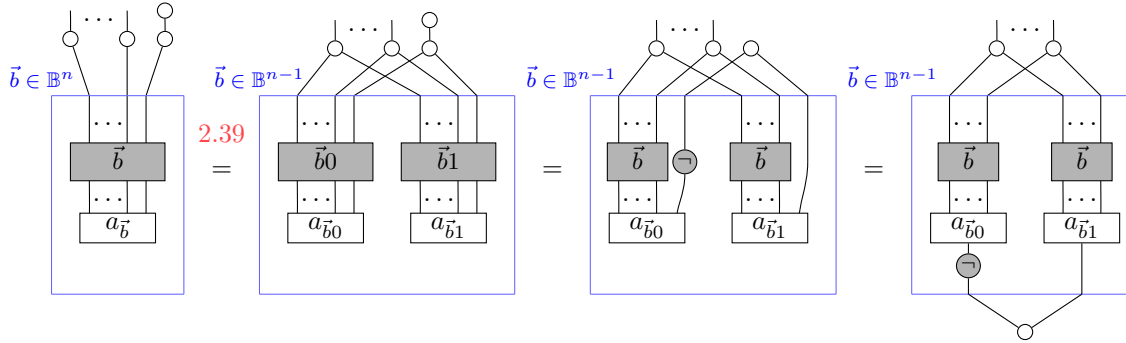


Note that, in the last step, (14) introduces a scalar white dot in each  $!$ -box, resulting in  $2^n$  white dots, one of which is cancelled by the  $\star$  outside the  $!$ -box. ■

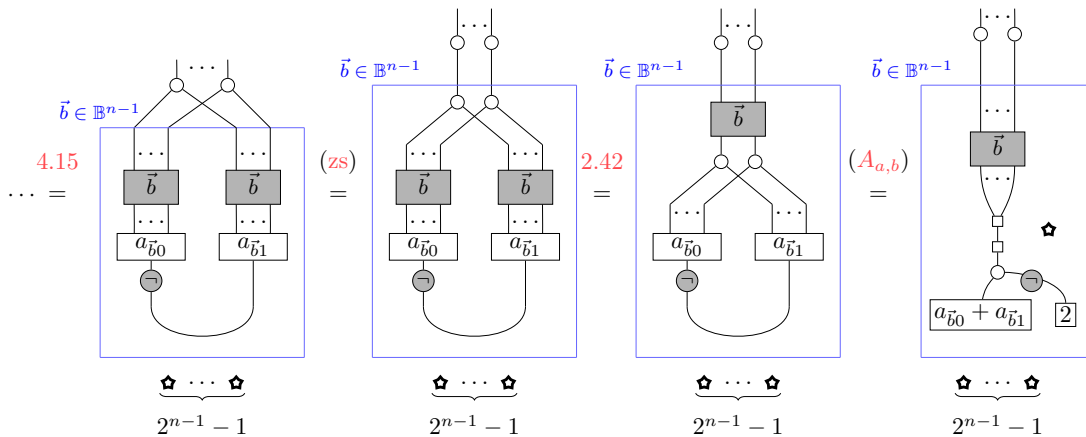
**Proposition 4.17.** The diagram resulting from applying  $\circlearrowleft$  to an output of a normal form diagram can be brought into normal form, when we have proven  $(A_{a,b})$  for all  $a$  and  $b$  in the normal form:

$$\begin{array}{c} \dots \\ | \\ \text{NF} \end{array} \circlearrowleft = \begin{array}{c} \dots \\ | \\ \text{NF}' \end{array}$$

*Proof.* Starting from an arbitrary normal form, with a  $\circlearrowleft$  plugged into the right most output, we first expand the annotated !-box:



Now the diagram is ready for application of Lemma 4.15, followed by the average rule:



This diagram can now be brought to normal form by application of Lemma 4.16. Note the stars exactly cancel the white dots introduced by that lemma. ■

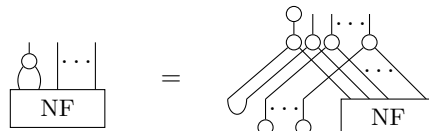
Our strategy will now be to show that any diagram can be decomposed into H-boxes, combined via the operations of extension, convolution, and contraction. This will give our completeness proof.

To simplify the decomposition of diagrams into H-boxes, we prove a few corollaries.

**Corollary 4.18.** The diagram resulting from applying  $\circlearrowright$  to a pair of outputs of a normal form diagram can be brought into normal form, when we have proven  $(I_a)$  and  $(A_{a,b})$  for all  $a$  and  $b$  in the normal form.

$$\begin{array}{c} \dots \\ | \\ \text{NF} \end{array} \circlearrowright = \begin{array}{c} \dots \\ | \\ \text{NF}' \end{array} \quad (27)$$

*Proof.* Applying a  $\circlearrowright$  to a pair of outputs has the same result as convolving with a cup, then contracting one of the outputs. That is, we can decompose (27) as follows:



then apply Lemma 4.7 and Propositions 4.10, 4.11, and 4.17. ■

**Corollary 4.19.** A diagram consisting of a cap applied to a normal form diagram can be transformed into a normal form diagram, when we have proven  $(I_a)$ ,  $(A_{a,b})$  and  $(A_{a',b'})$  for all  $a$  and  $b$  in the normal form and for every  $a'$  and  $b'$  that are sums of labels in the normal form:

$$\begin{array}{c} \text{cap} \\ \text{NF} \end{array} = \begin{array}{c} \text{NF}' \end{array}$$

*Proof.* Since the cap can be decomposed as  $\circlearrowleft \circ \circlearrowright$ , the result follows immediately from Corollary 4.18 and Proposition 4.17. ■

Thanks to Corollaries 4.12 and 4.19, we are able to turn any diagram of normal forms into a normal form. It only remains to show that the generators of the ZH-calculus can themselves be made into normal forms. We have already shown the result for H-boxes (Lemma 4.5), so the following will be sufficient:

**Lemma 4.20.** Any Z-spider can be brought into normal form using the rules of the ZH-calculus.

*Proof.* As shown in Eq. (12),  $\circlearrowleft$  is already a labelled H-box and thus is in normal form. By (zs),  $\circlearrowright = \circlearrowleft \circlearrowright$ , and hence it can be brought into normal form using Corollaries 4.12 and 4.19. This covers the cases of Z-spiders with 0 or 1 incident wires.

We can decompose any Z-spider with  $n \geq 2$  incident wires as a tensor product of  $(n - 1)$  cups, with each cup  $\circlearrowright$ -ed with its neighbours:

$$\text{Z-spider} \stackrel{(zs)}{=} \text{cup} \circlearrowright \text{cup} \circlearrowright \dots \circlearrowright \text{cup}$$

If  $n = 2$ , no  $\circlearrowright$  are needed and the equality is by (id) instead of (zs). In either case, the diagram can be brought into normal form by applying Lemma 4.7 and Corollaries 4.12 and 4.18. The intermediate normal forms only involve H-boxes labelled by a 0 and 1 and hence we have already proven the necessary cases of multiply, intro and average to prove this. ■

**Proposition 4.21.** If the ZH-calculus proves  $(M_{a,b})$ ,  $(I_a)$ , and  $(A_{a,b})$  for all integers  $a$  and  $b$ , then the ZH-calculus is complete, i.e. for any two ZH-diagrams  $D_1$  and  $D_2$ , if  $\llbracket D_1 \rrbracket = \llbracket D_2 \rrbracket$  then  $D_1$  is transformable into  $D_2$  using the rules of the ZH-calculus.

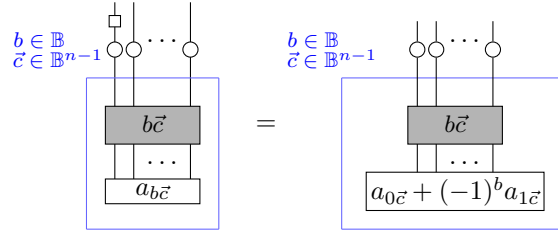
*Proof.* By Theorem 4.4, it suffices to show that any ZH diagram can be brought into reduced normal form. Lemmas 4.5 and 4.20 suffice to turn any generator into normal form. Then using Corollary 4.12 we can turn any tensor product of generators into a normal form and with Corollary 4.19 we can then apply any sort of wiring and reduce it to normal form. Finally, Proposition 4.9 allows us to bring the normal form to reduced normal form. ■

To prove completeness, it now remains to prove  $(M_{a,b})$ ,  $(I_a)$ , and  $(A_{a,b})$  for all integers  $a$  and  $b$ . To assist in this, let us prove a few lemmas that will help reduce diagrams to normal form when we have these rules for certain values of  $a$  and  $b$ . As we already have many individual cases of those rules — see Lemma 3.2 for  $(M_{a,b})$  and Lemma 3.7 for  $(I_a)$ , as well as Section 3.3 for  $(A_{a,b})$  — these lemmas can already be applied to many diagrams.

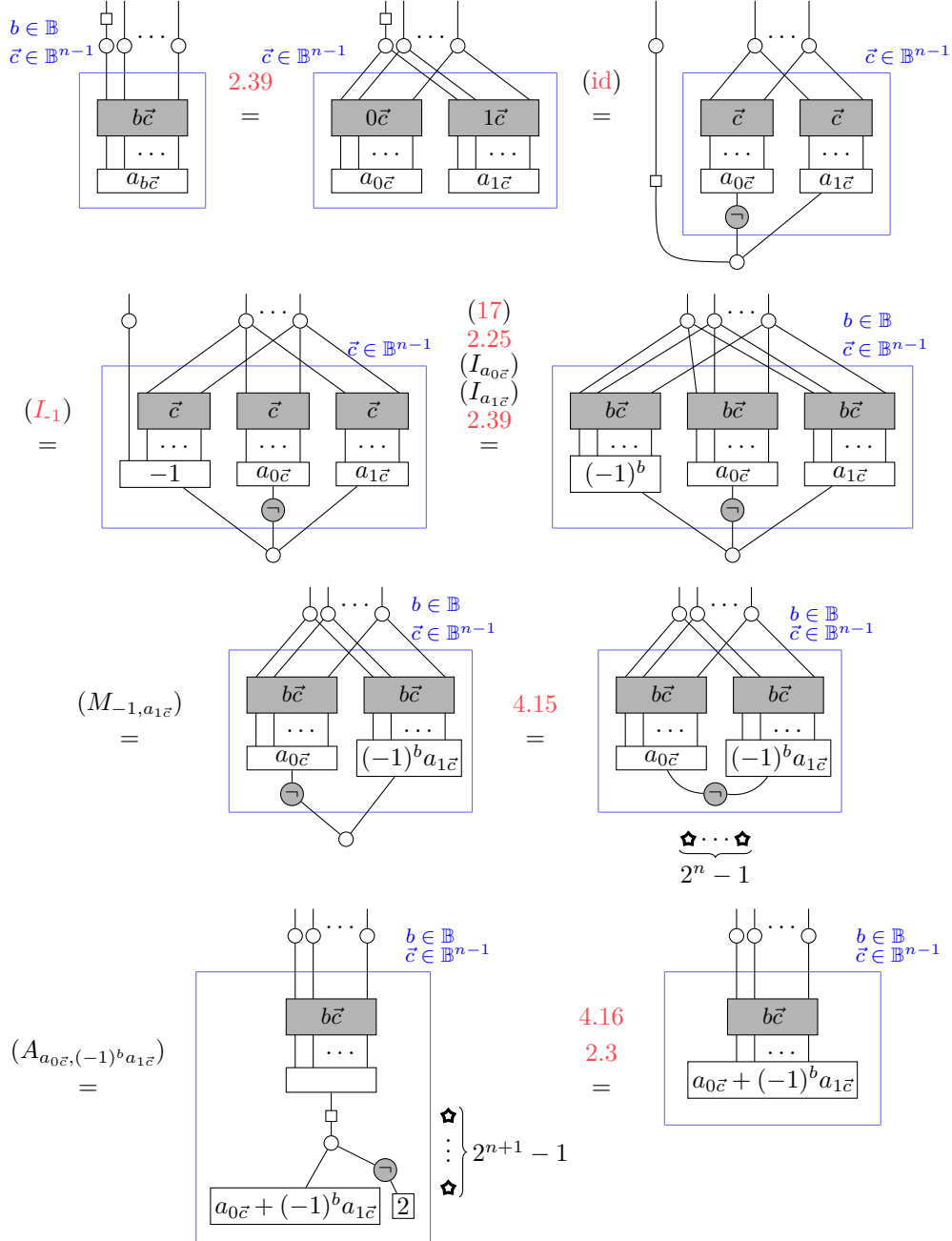
**Lemma 4.22.** Suppose we have  $(I_{a_b\vec{c}})$  for all  $b \in \mathbb{B}$ ,  $\vec{c} \in \mathbb{B}^{n-1}$ . Then we can consume a Hadamard gate into the normal form:

$$\begin{array}{c} b \in \mathbb{B} \\ \vec{c} \in \mathbb{B}^{n-1} \\ \text{Hadamard} \\ \vdots \\ a_{b\vec{c}} \end{array} = \begin{array}{c} b \in \mathbb{B} \\ \vec{c} \in \mathbb{B}^{n-1} \\ \text{Hadamard} \\ \vdots \\ a_{0\vec{c}} \quad (-1)^b a_{1\vec{c}} \end{array} \left. \begin{array}{l} \star \\ \vdots \\ \star \end{array} \right\} 2^n - 1$$

If we also have  $A_{a_{0\vec{c}}, a_{1\vec{c}}}$  and  $A_{a_{0\vec{c}}, -a_{1\vec{c}}}$  for all  $\vec{c} \in \mathbb{B}^{n-1}$ , then in fact:

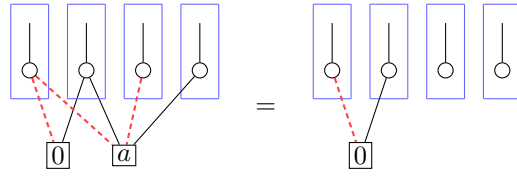


*Proof.* We start by manipulating the !-boxes and layout, and then perform the rewriting:

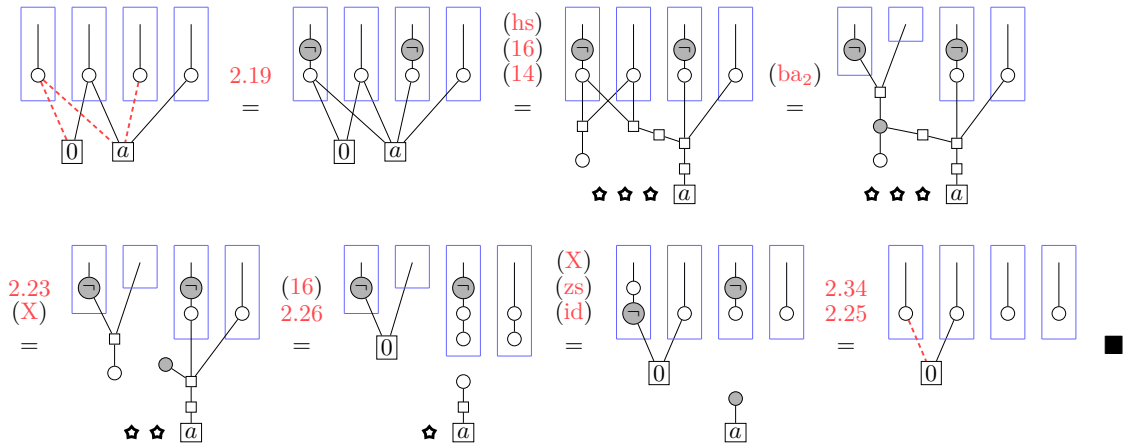


The following lemma allows us to (dis)connect integer-labelled H-boxes that are connected to a superset of a given 0-labelled H-box.

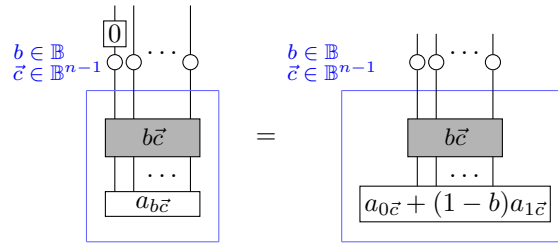
**Lemma 4.23.** For any integer  $a$ ,



*Proof.*



**Lemma 4.24.** Suppose we have proven  $(I_{a_b\vec{c}})$  for all  $b \in \mathbb{B}$ ,  $\vec{c} \in \mathbb{B}^{n-1}$  and  $(A_{a_0\vec{c}, a_1\vec{c}})$  for all  $\vec{c} \in \mathbb{B}^{n-1}$ , then we can consume an  $H(0)$  into the normal form:



*Proof.* The proof is analogous to that of Lemma 4.22, using Lemma 4.23 instead of  $(M_{-1, a_1\vec{c}})$ . ■

## 5 Arithmetic

In this section we show we can do simple arithmetic with labelled H-boxes. Namely we will show first that the following equation holds for all integers  $a$  and  $b$ .



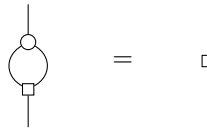
Hence, we can add H-box labels together. This will be crucial for when we will prove the general average rule in Section 6. Secondly, we will prove the multiply rule for all integers.

### 5.1 Proving addition for natural numbers

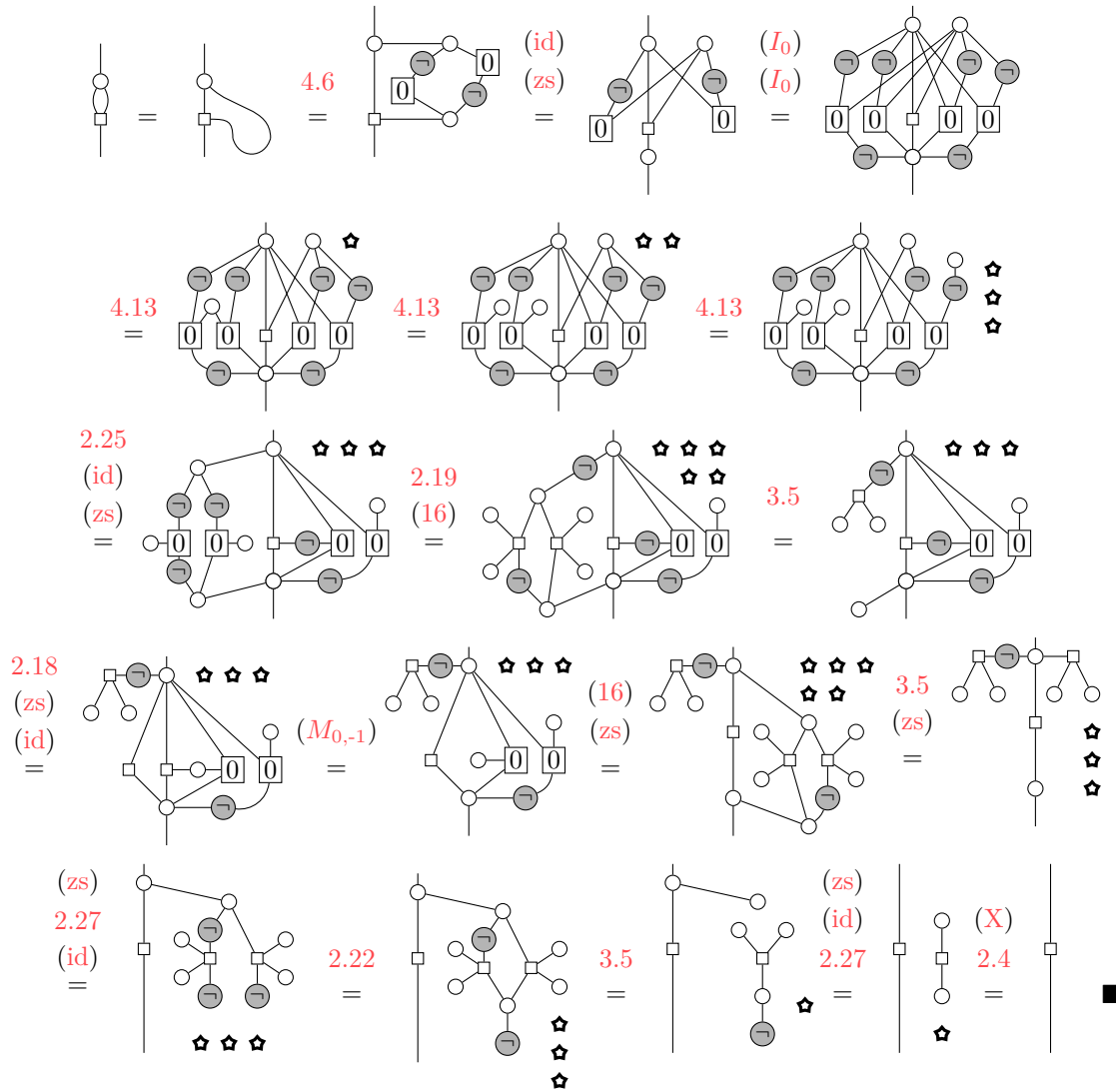
First we will prove (28) for natural numbers, which requires showing that (28) acts as expected when  $b = 1$ , and that furthermore the ‘addition gadget’ of (28) is associative. For this we need a couple of lemmas.

The next lemma, while looking deceptively simple, has a quite involved proof, requiring repeated use of Lemma 4.13. It was found by translating both sides to normal form, and then simplifying the intermediate steps.

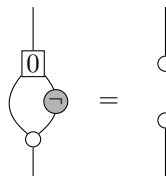
**Lemma 5.1.**



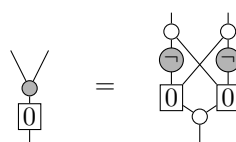
*Proof.*



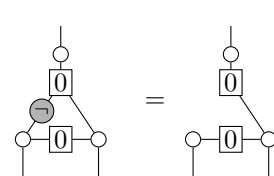
**Lemma 5.2.**



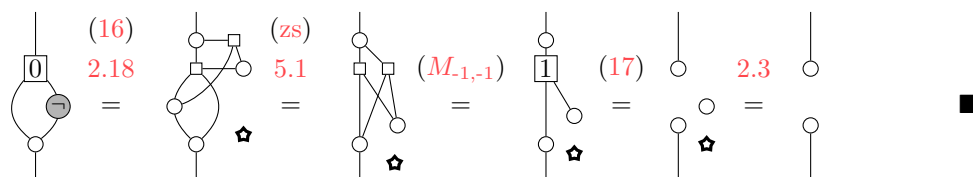
**Lemma 5.3.**



**Lemma 5.4.**

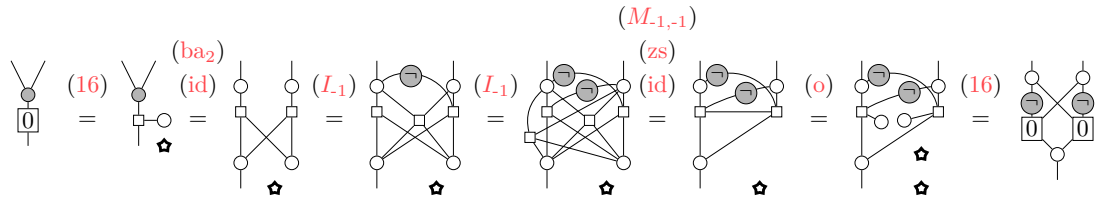


*Proof of Lemma 5.2.*



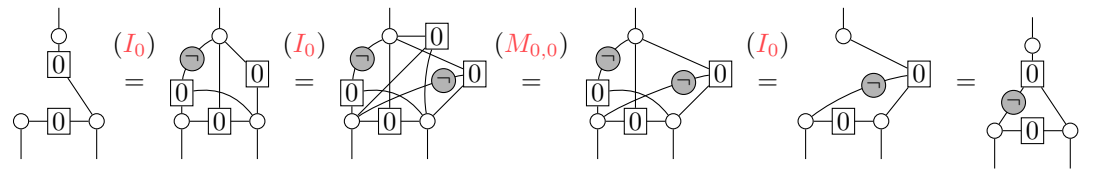


*Proof of Lemma 5.3.*



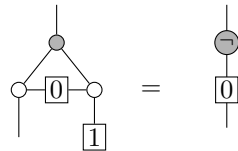
■

*Proof of Lemma 5.4.*

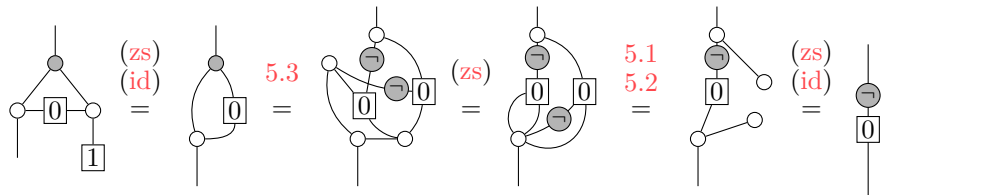


■

**Lemma 5.5.** Adding 1 is the same as applying the successor.

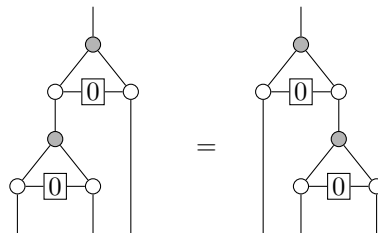


*Proof.*

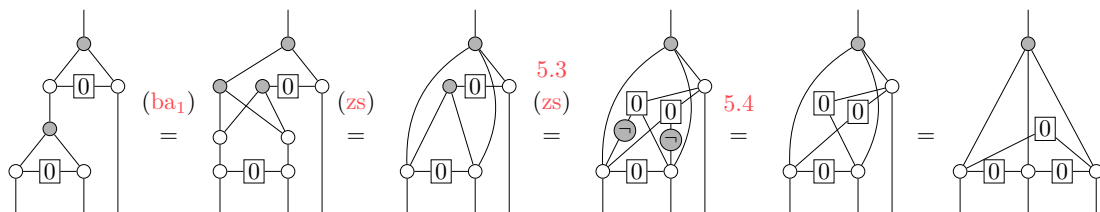


■

**Lemma 5.6.** Addition is associative.

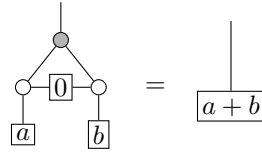


*Proof.* We can reduce the LHS to a diagram that is symmetric in the three inputs:

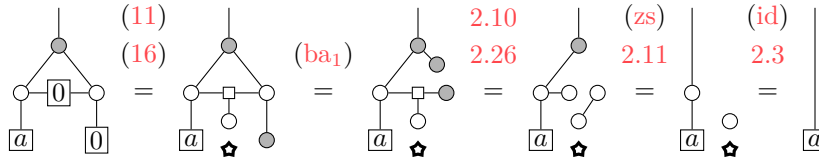


By symmetry we can also bring the RHS to this last diagram, and hence the LHS and RHS are equal. ■

**Lemma 5.7.** Let  $a, b \in \mathbb{N}$ . Then the following holds:

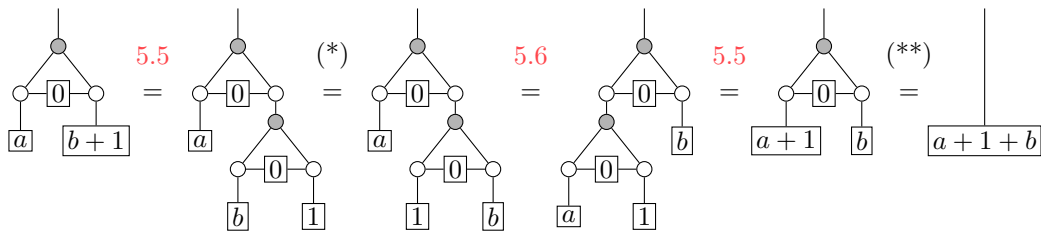


*Proof.* We prove this by induction on  $b$ . For any  $a$ , the case  $b = 0$  is straightforward:



Furthermore, for any  $a$ , Lemma 5.5 shows the case  $b = 1$ .

Now, suppose there is some  $b$  such that the desired result holds for any  $a$ , and consider  $b + 1$ . Then

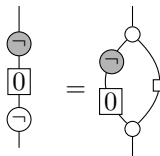


where the step marked (\*) is by ‘only topology matters’, and the step marked (\*\*) is the inductive hypothesis. Thus, the result follows by commutativity of addition. ■

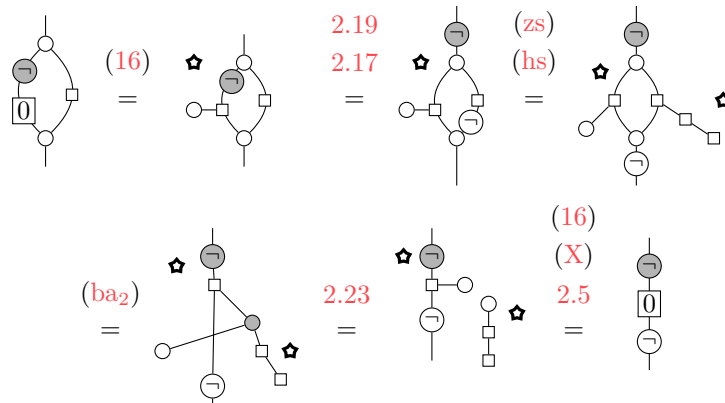
## 5.2 Proving addition for all integers

To generalise addition to arbitrary integers, we need two more lemmas.

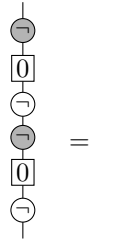
**Lemma 5.8.**



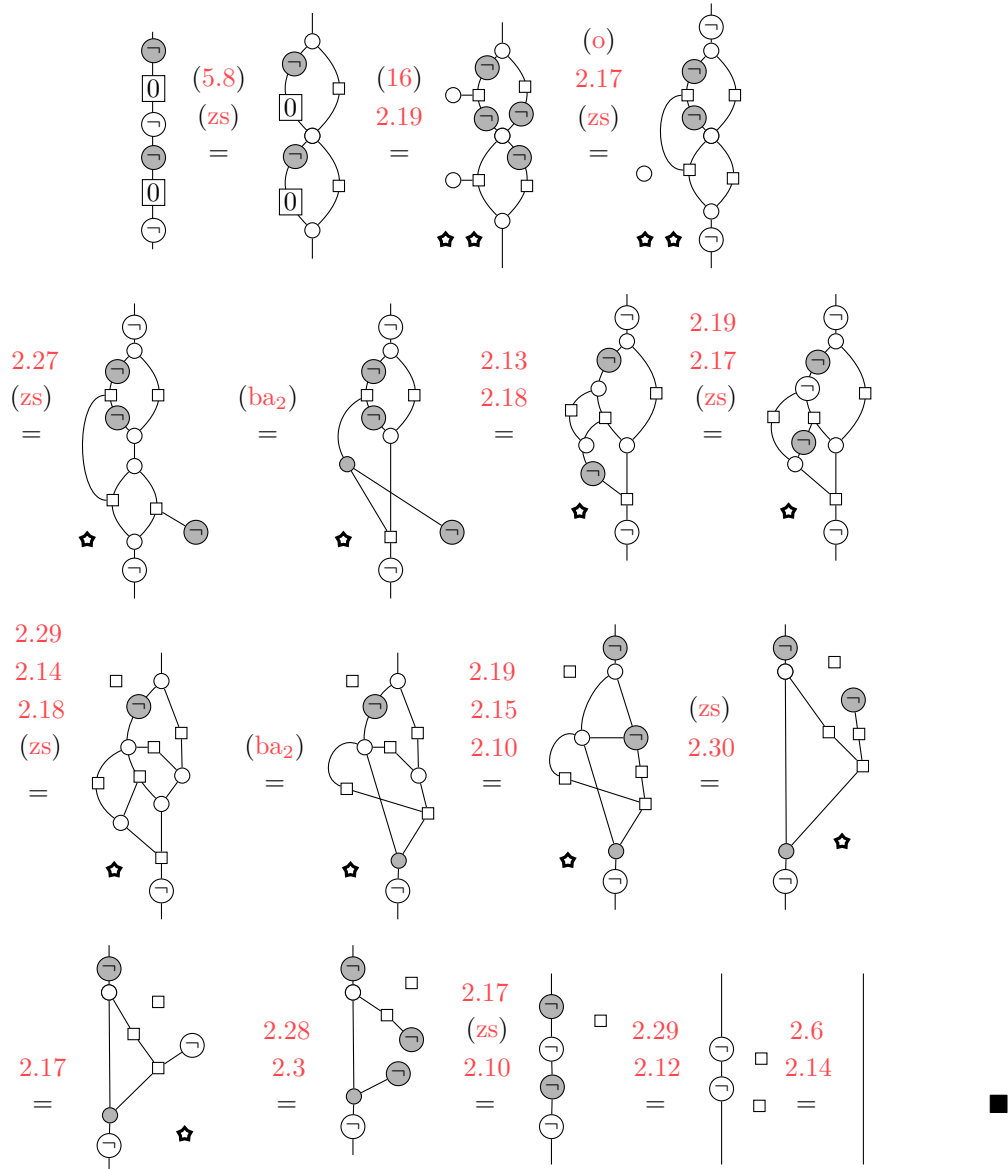
*Proof.*



**Lemma 5.9.** The successor has an inverse.

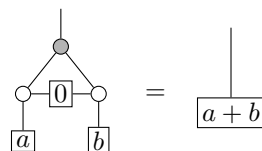


Proof.

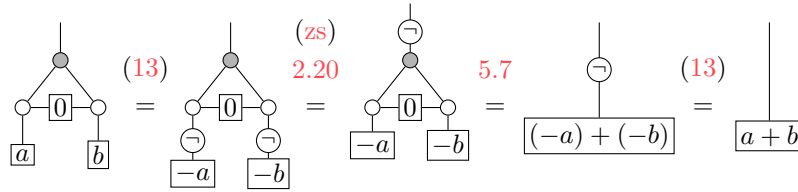


This lemma actually only shows that the successor operation has a one-sided inverse. But by conjugating the top and bottom using a negate spider, we see that this is in fact a two-sided inverse.

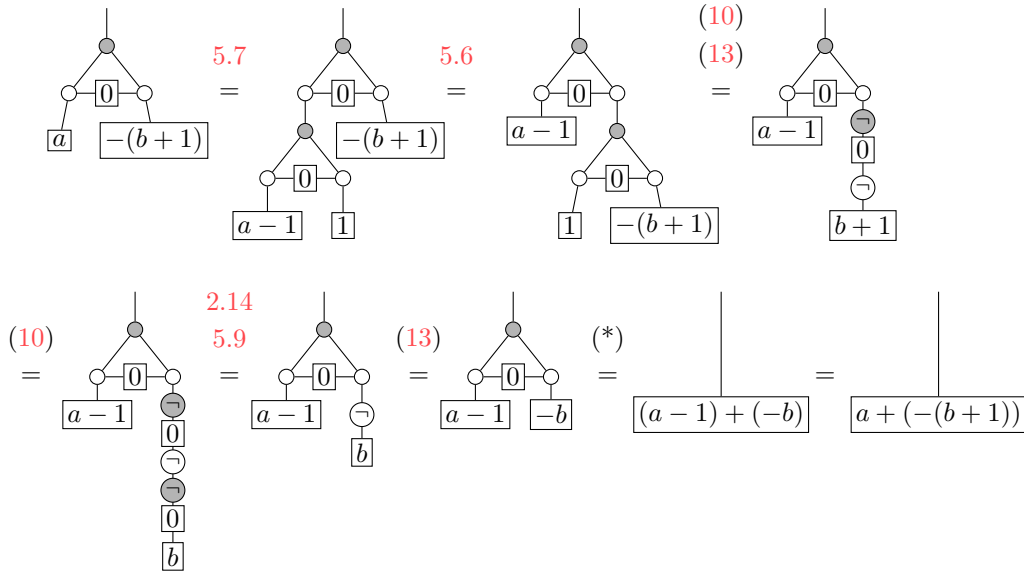
**Proposition 5.10.** Let  $a, b \in \mathbb{Z}$ . Then the following holds:



*Proof.* If  $a, b \geq 0$  this is just Lemma 5.7. If both  $a, b \leq 0$  we simply do:



Now suppose  $a > 0$  and  $b \leq 0$ , i.e.  $-b \geq 0$ . We prove by induction on  $-b \in \mathbb{N}$ . If  $-b = 0$  this is trivial. Suppose it holds for  $-b$ . In the calculation below we denote the induction step by (\*):



■

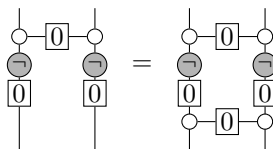
### 5.3 Proving multiplication

In this section we will show that the multiply rule introduced in Section 3.1 holds for all integers:

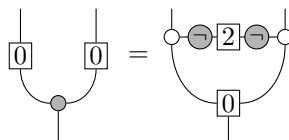
$$\begin{array}{c} \circ \\ \swarrow \quad \searrow \\ \boxed{a} \quad \boxed{b} \end{array} = \boxed{ab} \tag{29}$$

It is in any case clear that this operation is commutative, associative and the unit is the 1-labelled H-box. To prove that this indeed acts as multiplication it then suffices to prove it distributes over addition, which requires a bit of set-up to prove.

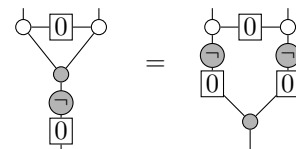
**Lemma 5.11.**



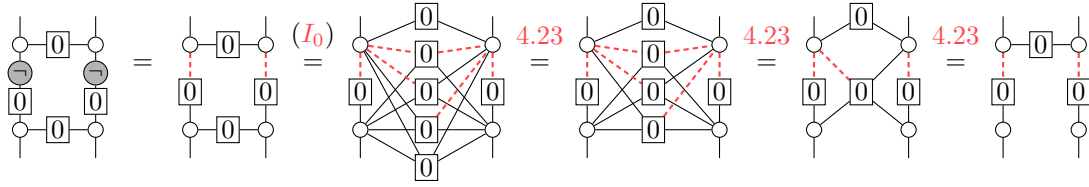
**Lemma 5.12.**



**Lemma 5.13.**

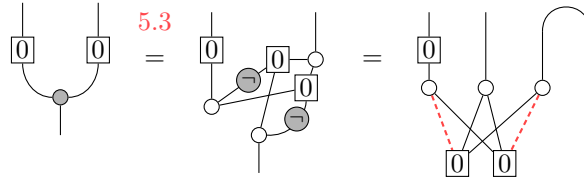


*Proof of Lemma 5.11.* Starting from the RHS, we apply  $(I_0)$  to the bottom  $H(0)$ -box. The newly-introduced  $H(0)$ -boxes can then be removed using Lemma 4.23: first with the top binary  $H(0)$ -box, then with the one on the right, and finally with the one on the left.



After replacing the red dashed lines with NOT gates and applying (id), this is equal to the LHS. ■

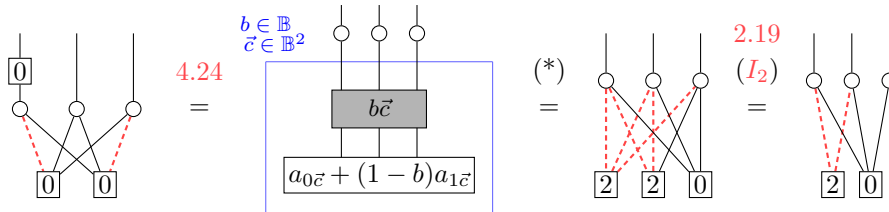
*Proof of Lemma 5.12.* First:



Note that – up to one bent wire and the introduction of trivial  $H(1)$ -boxes – the resulting diagram consists of an  $H(0)$ -box on the left output, applied to a normal form diagram. The H-box parameters in the normal-form diagram are

$$a_{b\vec{c}} = \begin{cases} 0 & \text{if } b = 0, \vec{c} = 11 \text{ or } b = 1, \vec{c} = 10 \\ 1 & \text{otherwise.} \end{cases}$$

By Lemma 3.7, we have both  $I_0$  and  $I_1$ , and by Lemma 3.12, we have  $A_{x,y}$  for any  $x, y \in \{0, 1\}$ . Thus we may apply Lemma 4.24 (where we have left out the bent wire for simplicity):



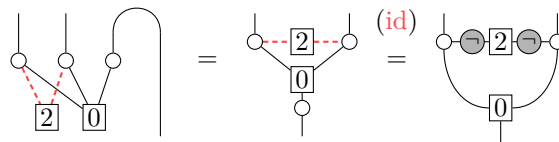
For the step denoted (\*), note that

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

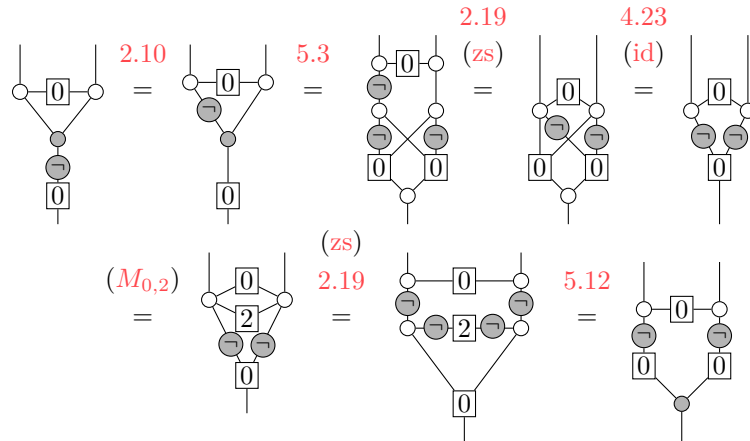
so

$$a_{0\vec{c}} + (1-b)a_{1\vec{c}} = \begin{cases} 2 & \text{if } b = 0 \text{ and } \vec{c} \in \{00, 01\} \\ 0 & \text{if } b = 1, \vec{c} = 11 \\ 1 & \text{otherwise,} \end{cases}$$

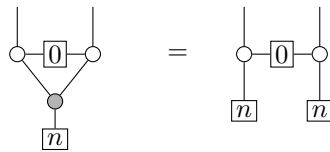
and that  $H(1)$ -boxes can be left out of the diagram by (17) and Lemma 2.25. Hence, re-introducing the bent wire, we find:



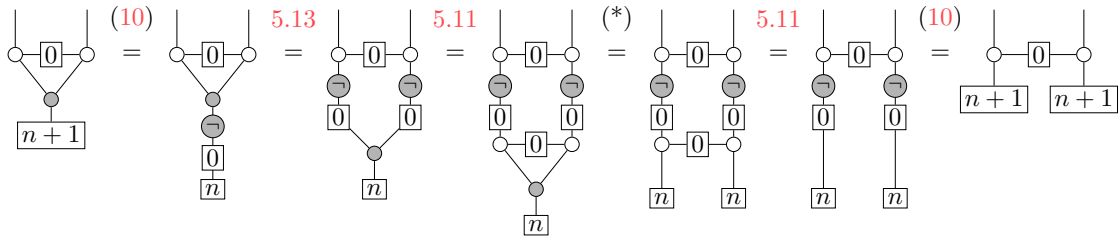
*Proof of Lemma 5.13.*



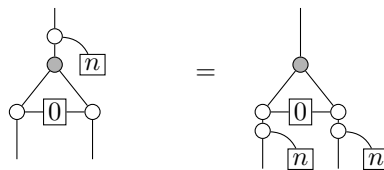
**Lemma 5.14.** For any natural number  $n$ :



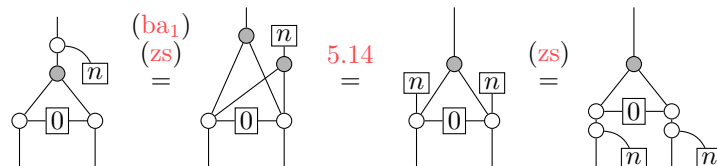
*Proof.* For  $n = 0$  this is straightforward. We prove by induction, so assume it holds for some  $n$ . In the calculation below we denote the induction step by (\*).



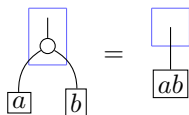
**Lemma 5.15.** The multiplication gadget distributes over addition, i.e. for any natural numbers  $n \in \mathbb{N}$ :



*Proof.*



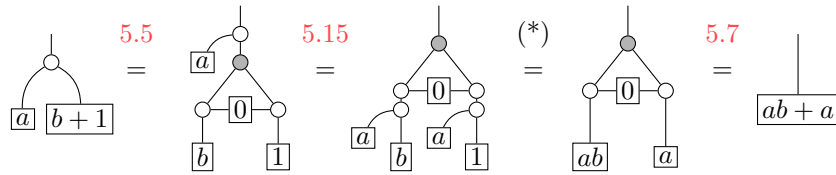
**Proposition 5.16.** The multiply rule holds for any integers  $a$  and  $b$ :



*Proof.* By Lemma 3.1 it suffices to show the non-!-boxed version, i.e.  $(M_{a,b})$ .

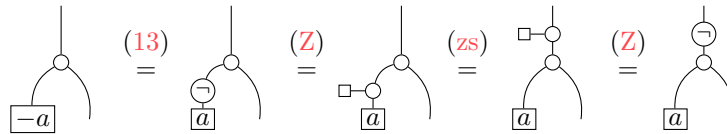
First we prove  $(M_{a,b})$  for  $a, b \in \mathbb{N}$ . The proof is by induction on  $b$  for any  $a \in \mathbb{N}$ . The base cases of  $b = 0$  and  $b = 1$  have already been shown in Lemma 3.2.

For the induction step, assume there exists some  $b$  such that  $(M_{a,b})$  holds for any  $a$ . Then:



where the step labelled (\*) uses the base case  $b = 1$  and the inductive hypothesis.

Now for negative numbers, we have:



and similarly for  $b$ , so the multiplication reduces to that of non-negative numbers. If there are two copies of  $\ominus$ , they cancel by Lemma 2.14. ■

## 6 Completeness

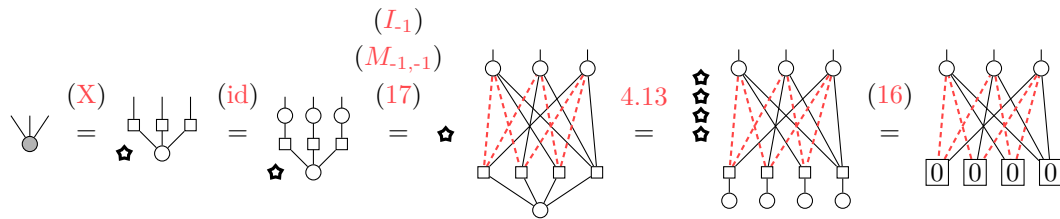
We write  $ZH \vdash D_1 = D_2$  when  $D_1$  and  $D_2$  can be proven to be equal using the rules of the ZH-calculus. The ZH-calculus is *complete* when  $\llbracket D_1 \rrbracket = \llbracket D_2 \rrbracket$  implies that  $ZH \vdash D_1 = D_2$ .

In Proposition 4.21, a ‘conditional’ completeness result was proven, that shows that if the ZH-calculus can prove the average, intro and multiply rule for all integers, then the ZH-calculus is complete. Proposition 5.16 shows that the multiply rule holds for all integers, so that it remains to prove the intro and average rule. That is what we will do in this section.

First, we will establish the average rule.

**Lemma 6.1.** The ternary X-spider can be brought to normal form.

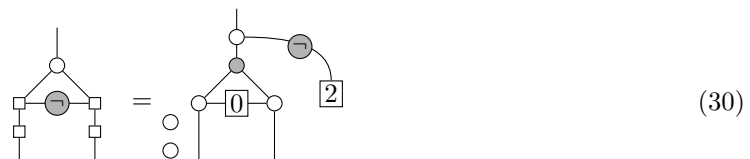
*Proof.*



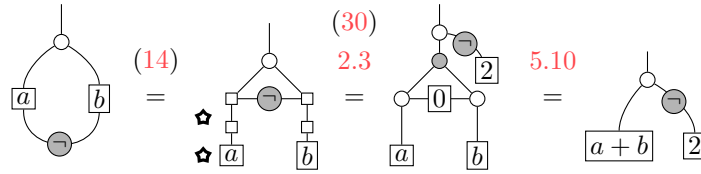
where there are three applications of Lemma 4.13 in the second-to-last step to break up the white spider. This last diagram is in normal form if we introduce the necessary trivial  $H(1)$  boxes. ■

**Proposition 6.2.** The ZH-calculus proves  $(A_{a,b})$  for all integers  $a$  and  $b$ .

*Proof.* It suffices to show that:

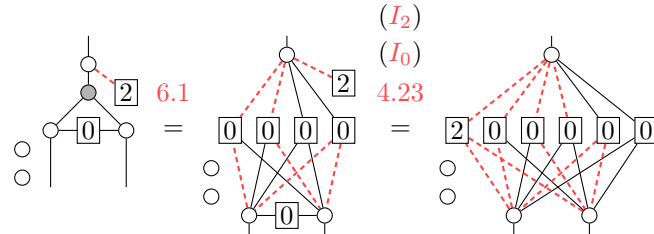


As then

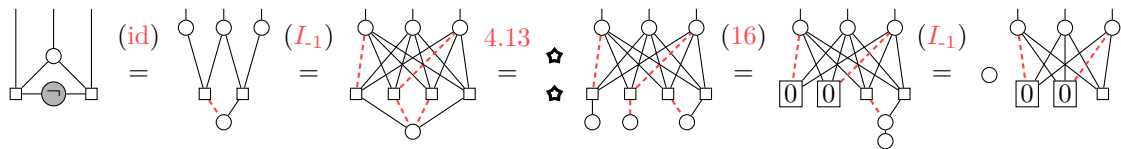


as required. We show this by bringing both sides of Eq. (30) to normal form.

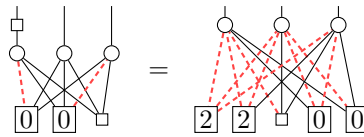
For the RHS of (30), first recall that we can bring the X-spider to normal form using Lemma 6.1, and hence:



For the LHS of (30), bending the wires up for simplicity, we first have



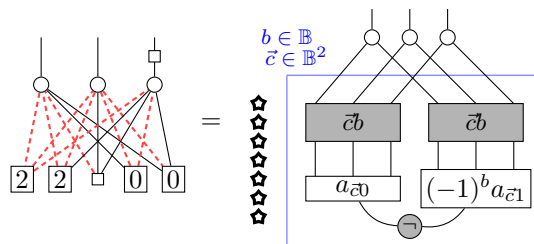
Ignoring the white dot, this is a normal form diagram with  $a_{011} = a_{110} = 0$ ,  $a_{111} = -1$  and  $a_{\bar{b}} = 1$  otherwise. Lemma 3.12 proves  $(A_{a,b})$  for all combinations of those values, thus we can apply the second part of Lemma 4.22 to show



Up to a partial transpose (for easier legibility of the matrices), this corresponds to

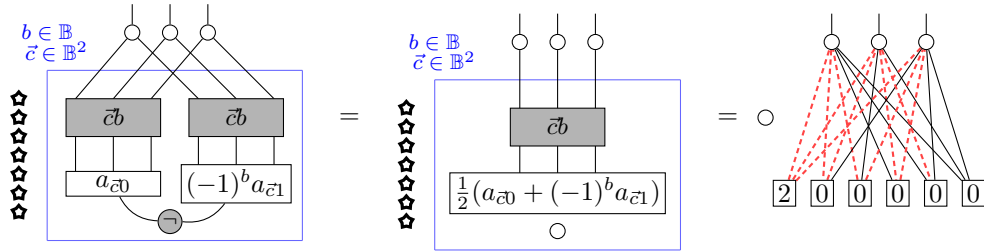
$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1+1 & 1+1 & 1+0 & 0-1 \\ 1-1 & 1-1 & 1+0 & 0+1 \end{pmatrix} = \begin{pmatrix} 2 & 2 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Again, this is a normal form diagram, this time with  $a_{000} = a_{001} = 2$ ,  $a_{011} = -1$ ,  $a_{100} = a_{101} = 0$  and  $a_{010} = a_{110} = a_{111} = 1$ . Now we apply the first part of Lemma 4.22 (note the changes due to the H-box being on the third output):



We have  $a_{c\bar{0}} = \pm a_{c\bar{1}}$  for all  $\vec{c} \in \mathbb{B}^2$ , so the two H-boxes in each copy of the !-box contain either equal or opposite integers. By Lemma 3.7 we have  $(I_a)$  for all  $a \in \{-1, 0, 1, 2\}$ , so we can use Lemma 3.10. Thus,

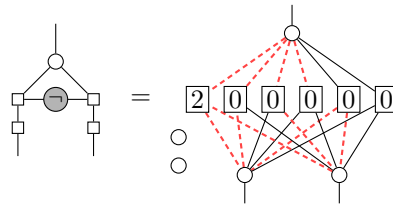




Up to a partial transpose, this sequence of rewriting steps corresponds to

$$\begin{pmatrix} 2 & 2 \\ 1 & -1 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 2+2 & 2-2 \\ 1-1 & 1+1 \\ 0 & 0 \\ 1+1 & 1-1 \end{pmatrix} = 2 \begin{pmatrix} 2 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$$

Combining everything and bending back the legs, we find

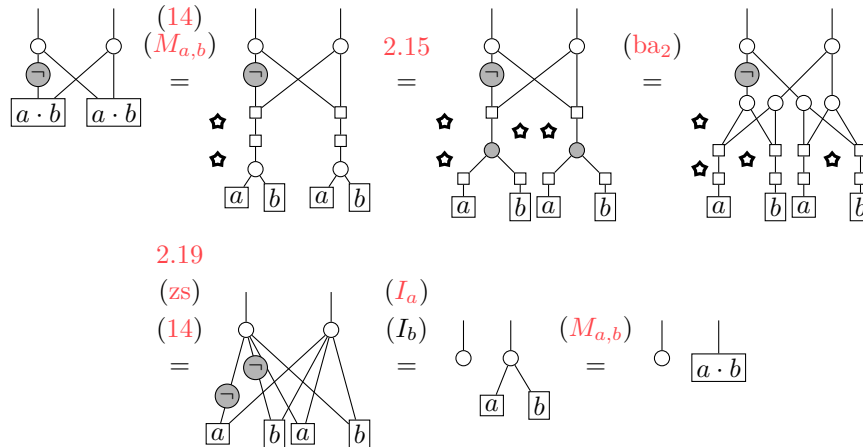


This is the same as the RHS. ■

It now remains to prove  $(I_a)$  for all integers  $a$ . We can reduce this problem to proving it for natural numbers using the following lemma.

**Lemma 6.3.** Suppose we have proven  $(I_a)$  and  $(I_b)$  for some integers  $a$  and  $b$ . Then we can also prove  $(I_{a \cdot b})$ . In particular, if we have  $(I_a)$ , then we also get  $(I_{-a})$ .

*Proof.* We bend all the wires up for a more easy presentation. The proof is then straightforward:



Because we have  $(I_{-1})$  (Lemma 3.7), we get  $(I_{-a})$  if we have  $(I_a)$ . ■

**Proposition 6.4.** The ZH-calculus proves  $(I_a)$  for all integers  $a$ .

*Proof.* By the previous lemma it suffices to prove  $(I_n)$  for all natural numbers. Note that we have  $(I_0)$  and  $(I_1)$  so that by induction it suffices to show that we can get  $(I_{n+1})$  out of  $(I_n)$ . We will

show that:

(31)

This is sufficient because then:

We prove Eq. (31) by reducing both sides of the equation to normal form, beginning with the part that is the same on both sides. To start, we bring the top part of the diagram into normal form and then apply the first part of Lemma 4.22 to the H-box on the bottom left wire.

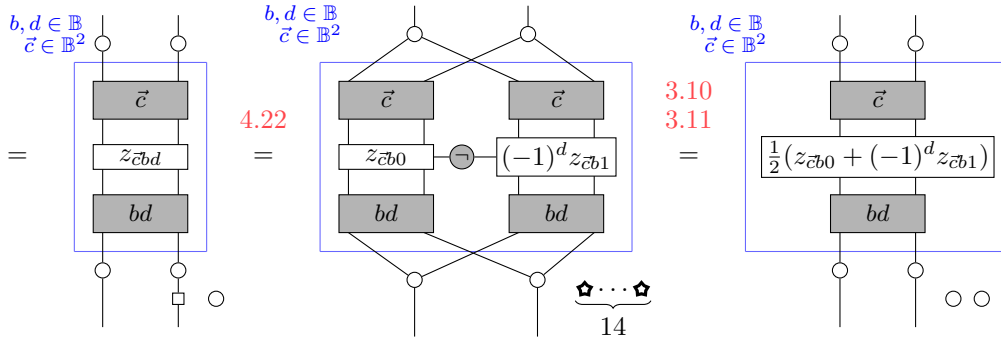
For the final step, note that the coefficients  $a_{\vec{c}bd}$  of the normal-form diagram are  $\pm 1$ , indeed  $a_{0101} = a_{0111} = a_{1101} = a_{1111} = -1$  and  $a_{\vec{c}bd} = 1$  otherwise. We have  $(I_1)$  and  $(I_{-1})$  by Lemma 3.7. Thus we can use the first equalities from Lemmas 3.10 and 3.11. This transformation corresponds to

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1+1 & 1+1 & 1-1 & 1-1 \\ 1-1 & 1-1 & 1+1 & 1+1 \\ 1+1 & 1+1 & 1-1 & 1-1 \\ 1+1 & -1-1 & 1-1 & -1+1 \end{pmatrix} = 2 \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}$$

Let  $z_{\vec{c}bd} := \frac{1}{2} (a_{\vec{c}0d} + (-1)^b a_{\vec{c}1d})$ , then

$$z_{\vec{c}bd} = \begin{cases} -1 & \text{if } \vec{c}bd = 1101 \\ 0 & \text{if } (\vec{c} \in \{00, 10, 11\} \wedge b = 1) \vee (\vec{c} = 01 \wedge b = 0) \\ 1 & \text{otherwise.} \end{cases}$$

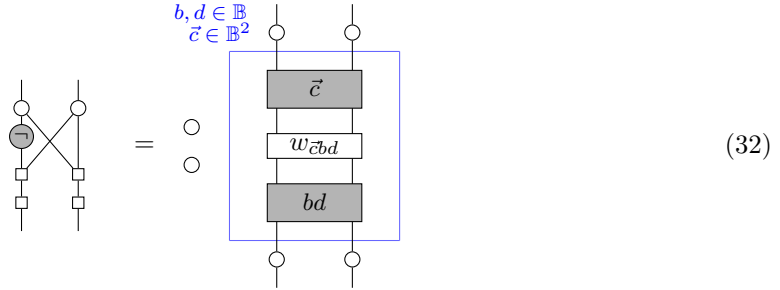
For each  $\vec{c}$  and  $b$ , we have  $z_{\vec{c}b0} = \pm z_{\vec{c}b1}$ , and by Lemma 3.7 we have  $(I_0)$ ,  $(I_1)$  and  $(I_{-1})$ . Hence we can again apply the first part of Lemma 4.22, followed by the first equalities of Lemmas 3.10 and 3.11.



This step corresponds to:

$$2 \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} = 2 \begin{pmatrix} 1+1 & 1-1 & 0+0 & 0+0 \\ 0+0 & 0+0 & 1+1 & 1-1 \\ 1+1 & 1-1 & 0+0 & 0+0 \\ 1-1 & 1+1 & 0+0 & 0+0 \end{pmatrix} = 4 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Combining everything, we have



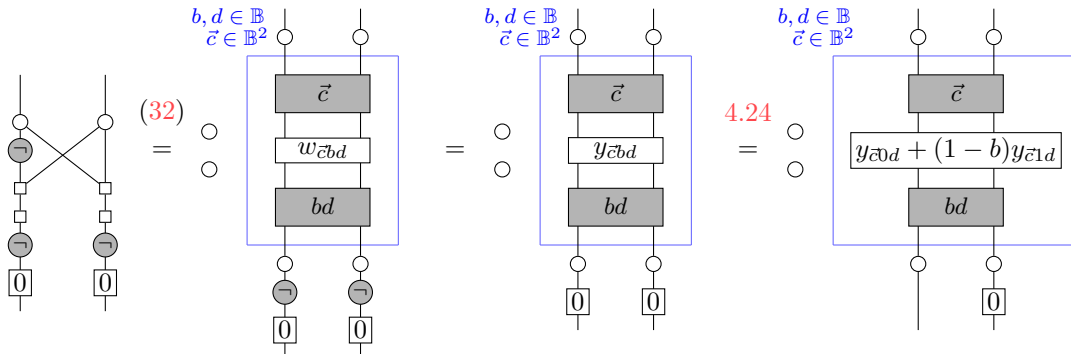
where

$$w_{\vec{c}bd} := \frac{1}{2} (z_{\vec{c}b0} + (-1)^d z_{\vec{c}b1}) = \begin{cases} 1 & \text{if } \vec{c}bd \in \{0000, 0110, 1000, 1101\} \\ 0 & \text{otherwise.} \end{cases}$$

Now, for the left-hand side of Eq. (31), let  $y_{\vec{c}bd} := w_{\vec{c}(1-b)(1-d)}$ , so that

$$y_{\vec{c}bd} = \begin{cases} 1 & \text{if } \vec{c}bd \in \{0011, 0101, 1011, 1110\} \\ 0 & \text{otherwise.} \end{cases}$$

Note that  $y_{\vec{c}0d}y_{\vec{c}1d} = 0$  for all  $\vec{c}, d$ , so we will be able to apply Lemma 4.24 because we have already proved  $(A_{y_{\vec{c}0d}, y_{\vec{c}1d}})$  for all  $\vec{c}, d$ . By absorbing the two NOTs on the bottom wires and relabelling, we find:



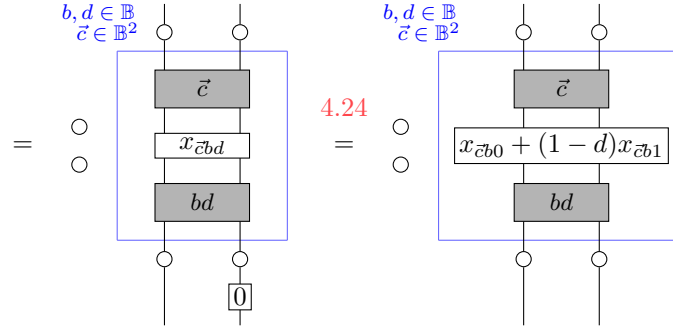
This corresponds to

$$4 \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = 4 \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Finally, let  $x_{\vec{c}bd} := y_{\vec{c}bd} + (1-b)y_{\vec{c}1d}$ , then

$$x_{\vec{c}bd} = \begin{cases} 1 & \text{if } \vec{c}bd \in \{0001, 0101, 0111, 1001, 1100\} \\ 0 & \text{otherwise.} \end{cases}$$

Note that  $x_{\vec{c}b0}x_{\vec{c}b1} = 0$  for all  $\vec{c}b$ , so we use the same process for the final 0-labelled H-box.

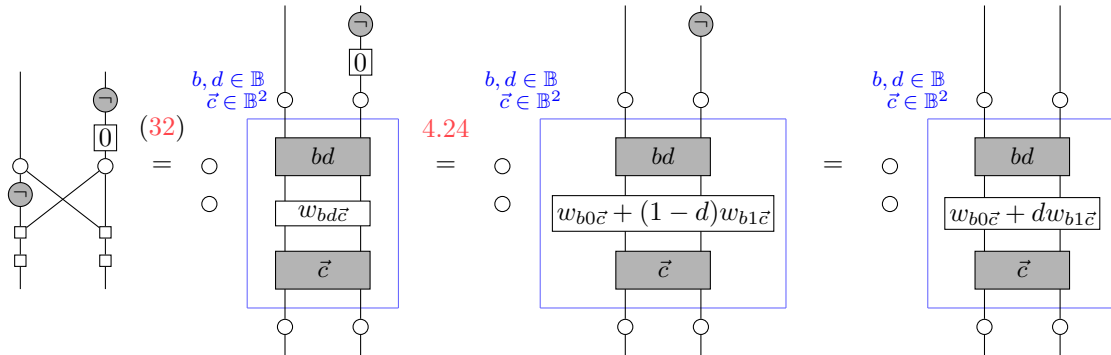


This rewrite step corresponds to:

$$4 \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = 4 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Apart from the two white dots, the left-hand side diagram of Eq. (31) has now been brought into normal form.

For the right-hand side of Eq. (31), note that  $w_{b0\vec{c}}w_{b1\vec{c}} = 0$  for all  $b, \vec{c}$ , so we can again use Lemma 4.24, before absorbing the NOT and relabelling:



This corresponds to

$$4 \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = 4 \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \\ = 4 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

which is equal to the LHS, so (31) is proved.  $\blacksquare$

**Theorem 6.5.** The ZH-calculus is complete.

*Proof.* Propositions 5.16, 6.2 and 6.4 show that  $(M_{a,b})$ ,  $(A_{a,b})$  and  $(I_a)$  are provable in the ZH-calculus for all integers  $a$  and  $b$ . Hence Proposition 4.21 shows that the ZH-calculus is complete.  $\blacksquare$

## 7 The ZH-calculus over arbitrary rings

The ZH-calculus as defined above represents matrices over  $\mathbb{Z}[\frac{1}{2}]$ . To handle these efficiently, we introduced integer labels on H-boxes. In this section, we extend this idea: we allow H-boxes to be labelled *a priori* by elements of some ring  $R$ . As a result, diagrams can express matrices in  $R\text{-bit}$ , the full subcategory of  $R$ -modules where the objects are of the form  $(R \oplus R)^{\otimes n}$ . This means our diagrams are interpreted as matrices of shape  $2^n \times 2^m$  with entries in  $R$ . The conditions on  $R$  and the ways in which we show completeness will be the core of this section. We will dub this new calculus  $ZH_R$ , or ‘ZH over  $R$ ’, with the original phase-free calculus simply called ‘ZH’ without a subscript. The ruleset of  $ZH_R$  will be a strict superset of the rules of Figure 1, and hence anything we have already proven for ZH will remain true in  $ZH_R$ .

The original ZH-calculus as introduced in [5] represented matrices over the complex numbers, and hence in our notation is called  $ZH_{\mathbb{C}}$ . The utility of  $ZH_{\mathbb{C}}$  is that it can describe evolutions of qubits. This does not need to be the only useful choice of ring: The ZW-calculus is similarly parametrised by a commutative ring [27], and this flexibility is what allowed for the proof of completeness for Clifford+T ZX via  $ZW_{\mathbb{Z}[\frac{1}{2}]}$  [29]<sup>5</sup>.

We will show that we can make  $ZH_R$  sound, complete, and universal into  $R\text{-bit}$  provided that  $R$  is commutative with  $1 \neq 0$ , and the element 2 is not a zero-divisor. Unitarity of the ring is required so that we can preserve the interpretation of the  $\curvearrowright$  generator, which uses 0 and 1. Commutativity of  $R$  is required because we require the following isometry of diagrams to be sound:

$$\boxed{a} \boxed{b} \simeq \boxed{b} \boxed{a} \implies \llbracket \boxed{a} \boxed{b} \rrbracket = \llbracket \boxed{b} \boxed{a} \rrbracket \implies a \times b = b \times a$$

The condition on the element 2 is needed since otherwise we would no longer have a well-defined relationship allowing us to go between white spiders and grey spiders as in (NOT) and (X). Likewise we would no longer be able to freely introduce pairs of Hadamard gates via the (hh) rule. These relationships are so important that we exclude the case where 2 is a zero-divisor from this paper and leave it to future work.

We will first deal with the case where  $\frac{1}{2} \in R$  in Section 7.1, and afterwards we will adapt the results to work when  $\frac{1}{2} \notin R$  in Section 7.2. When  $\frac{1}{2} \in R$  we will show that  $\heartsuit$  is equal to the  $H(\frac{1}{2})$  box with no inputs or outputs, and hence the  $\heartsuit$  generator is superfluous.

The completeness result for when  $\frac{1}{2} \in R$  uses the rules given in Figure 3, which are a superset of the rules given in Figure 1. If the ring  $R$  does not contain  $\frac{1}{2}$  but does contain 2, which is not a zero-divisor, then the ring  $R[\frac{1}{2}]$  is well defined. We can then use the completeness of  $ZH_{R[\frac{1}{2}]}$  and the introduction of a meta-rule to eliminate all instances of the  $\heartsuit$  generator to give completeness of  $ZH_R$  for  $\frac{1}{2} \notin R$ .

Note that the ZH-calculus as introduced in Section 2 is not the same as  $ZH_R$  for any  $R$ , since in the ZH-calculus the labels are from  $\mathbb{Z}$  but the interpretation is into  $\mathbb{Z}[\frac{1}{2}]\text{-bit}$ .

### 7.1 ZH-calculus over rings with a half

In this section we will take  $R$  to be a commutative unital ring that contains  $\frac{1}{2}$  (and so 2 is not a zero-divisor). We generalise the labelled H-boxes of Section 2.5 by letting the H-boxes in  $ZH_R$  be labelled by any element  $r \in R$ :

$$\left[ \begin{array}{c} \overbrace{\dots}^n \\ \curvearrowright \\ \underbrace{\dots}^m \end{array} \right] := \sum r^{i_1 \dots i_m j_1 \dots j_n} |j_1 \dots j_n\rangle \langle i_1 \dots i_m|$$

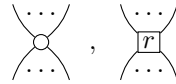
As in Section 2.1, the sum in the second equation is over all  $i_1, \dots, i_m, j_1, \dots, j_n \in \{0, 1\}$  so that the H-box above represents a matrix with all entries equal to 1, except the bottom right element, which is  $r$ . If the label is  $-1$  (so that it matches the usual H-box of Section 2.1) we will usually

<sup>5</sup>Due to these ideas emerging concurrently the authors of [29] do not explicitly use  $ZW_{\mathbb{Z}[\frac{1}{2}]}$  but something very similar.

leave out the label. Besides these more general H-boxes, the other generators are identical to those found in Section 2.1. Note that because  $\frac{1}{2} \in R$ , the star generator is redundant:

$$\llbracket \star \rrbracket = \frac{1}{2} = \llbracket \left[ \frac{1}{2} \right] \rrbracket$$

We will derive the corresponding diagrammatic equality later in Lemma 7.9. We could therefore use just the following generators when  $\frac{1}{2} \in R$ :



But in order to retain compatibility with the previous results, we will continue to write  $\star$  to represent the scalar  $\frac{1}{2}$ .

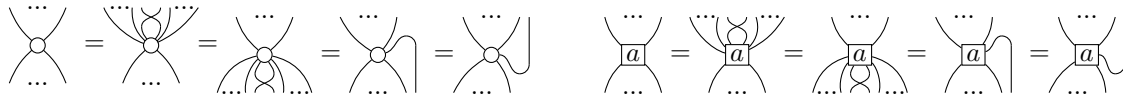
As before, we define the grey spiders and NOT gate of (1) as derived generators. With these generators and derived generators established we can state our theorem of universality.

**Theorem 7.1.** The calculus  $ZH_R$  is universal over  $R$ -bit, shown using the unique normal form diagram:

$$\left[ \prod_{\vec{b} \in \mathbb{B}^n} (t_{\vec{b}} \circ H_n(a_{\vec{b}})) \right] = \sum_{\vec{b} \in \mathbb{B}^n} a_{\vec{b}} |\vec{b}\rangle. \tag{33}$$

*Proof.* The equation above was demonstrated in the proof of Theorem 4.4. The lack of stars in this version comes from noting that every element in  $R$  is divisible by 2 (as  $\frac{1}{2} \in R$ ) and so stars are always subsumed into the ring elements  $a_{\vec{b}}$  in the reduced normal form. ■

Now let us establish our ruleset for the  $ZH$ -calculus over  $R$ . The rules for  $ZH_R$  are given in Figure 3 along with the same ‘only topology matters’ meta rule and all the same symmetry conditions as before:



Here,  $a \in R$  is arbitrary.

In comparison to the rules of Figure 1, (HS) has been generalised from (hs) to allow arbitrary labelled H-boxes and (U) has been added to relate the white dot to an H-box labelled by the 1 element of the ring. The rules (M), (A) and (I) already appeared in Section 3 as  $(M_{a,b})$ ,  $(I_a)$  and  $(A_{a,b})$ , but there they were derived rules that applied to integer labelled H-boxes, while here they are axioms that relate the ring elements of  $R$ . Note that it is not immediately obvious that the  $\mathbb{Z}$ -labelled H-boxes of Section 2.5 correspond to the  $R$ -labelled H-boxes of the  $ZH_R$ -calculus. We will prove this imminently, though first we observe the soundness of  $ZH_R$ .

**Proposition 7.2.** The rules of Figure 3 are sound.

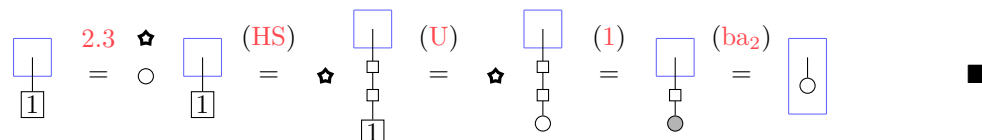
*Proof.* Because of Proposition 2.2, we simply need to check that the rules not already present in  $ZH$  are sound; these are (HS), (M), (A), (I), and (U). This is done by straightforward calculation. ■

To show how the labelled H-boxes of  $ZH$  and  $ZH_R$  relate, we first show that  $H(1)$  indeed satisfies (17):

**Lemma 7.3.** The phase-1 H-box of any arity decomposes:



*Proof.*



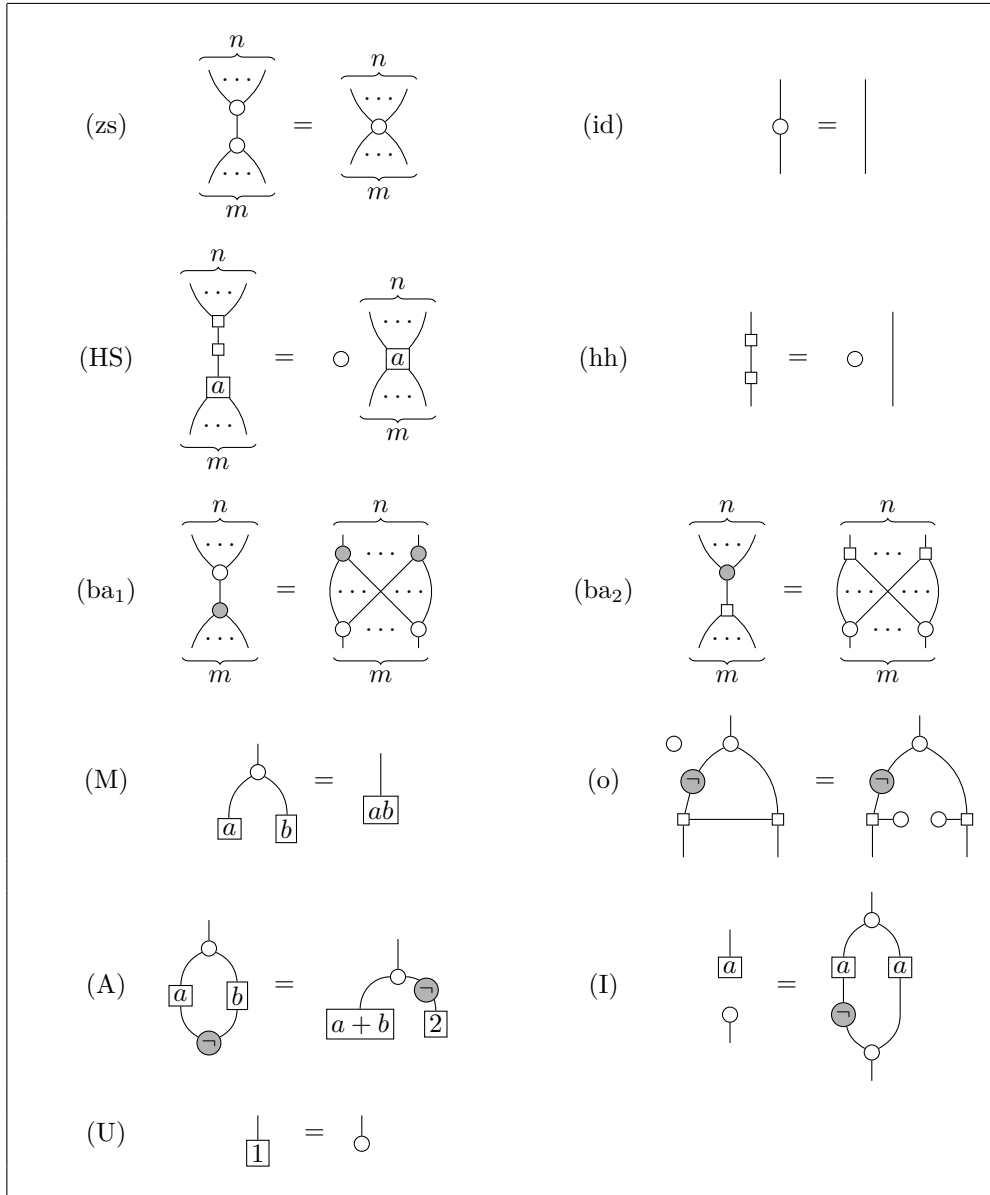
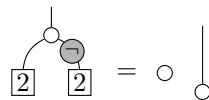
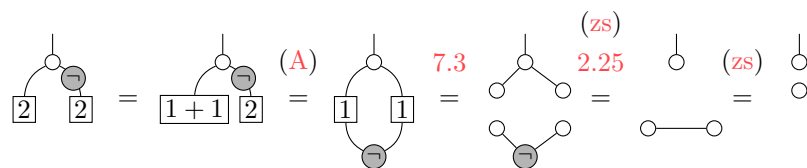


Figure 3: The set of rules for the  $ZH_R$ -calculus. Throughout,  $m, n$  are nonnegative integers and  $a, b$  are arbitrary elements of the commutative unital ring  $R$  that contains a half. The lowercased rules are the same as those in Figure 1. The rules (M), (A), (U), and (I) are pronounced *multiply*, *average*, *unit*, and *intro* respectively.

**Lemma 7.4.**



*Proof.*



Now we can also establish that  $H(0)$  and  $H(2)$  are as in Section 2.5.

**Lemma 7.5.**

$$\boxed{0} = \bullet$$

**Lemma 7.6.**

$$\boxed{2} = \star \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array}$$

**Lemma 7.7.**

$$\boxed{2} = \circ$$

*Proof of Lemma 7.5.*

$$\begin{array}{c} \text{2.3} \\ \text{(id)} \\ \boxed{0} \end{array} = \begin{array}{c} \text{(zs)} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{7.4} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{(zs)} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \text{(M)} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{(A)} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \text{7.3} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{(id)} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{2.25} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \bullet \quad \blacksquare$$

*Proof of Lemma 7.6.*

$$\begin{array}{c} \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{(id)} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \text{2.3} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{7.4} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \text{(I)} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{(16)} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \text{(M)} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{(id)} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \text{(A)} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{2.3} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{(id)} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \text{2.12} \\ \star \\ \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \boxed{2} \end{array} \quad \blacksquare$$

*Proof of Lemma 7.7.* Identical to the proof of Lemma 2.35. \blacksquare

With these equivalences established, we will now use induction to show that the labelled H-boxes in ZH are equivalent (using the rules of  $ZH_R$ ) to the corresponding H-boxes in  $ZH_R$ .

**Proposition 7.8.** All the integer labelled H-boxes of the ZH-calculus (Section 2.5) are provably equal in  $ZH_R$  to the corresponding labelled H-box in the  $ZH_R$ -calculus.

*Proof.* We will denote a labelled ZH H-box of Section 2.5 by  $H(n')$  to distinguish it from a generator of the  $ZH_R$ -calculus. We have for every  $n \in \{1, 2, 3, \dots\} \subset R$ :

$$\begin{array}{c} \boxed{n+1} \end{array} = \begin{array}{c} \text{(id)} \\ \text{2.3} \\ \text{7.4} \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{(A)} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \text{(*)} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{(6.2)} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \text{7.4} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{2.3} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} \begin{array}{c} \text{(id)} \\ \star \\ \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \circ \quad \circ \end{array} \end{array} = \begin{array}{c} \boxed{(n+1)'} \end{array}$$

Here the induction step is denoted by (\*). Note that we first used the average rule as an axiom of  $ZH_R$  acting on the generators (written (A)), and then as a derived rule from the original ZH-calculus (Proposition 6.2). For negative integers we simply note that the definition of negation (13) corresponds precisely to multiplying by  $-1$  in (M). \blacksquare

As in Section 3, we can prove !-boxed versions of the axioms (M), (A) and (I). In particular, the !-boxed version of (M) and (U) (i.e. Lemma 7.3) where the !-box is expanded 0 times give:

$$\boxed{a} \boxed{b} = \boxed{ab} \qquad \boxed{1} = \boxed{\quad} \qquad (34)$$

These rules enable us to multiply scalars at will, and in particular eliminate scalars by multiplying by the inverse (when the inverse exists in  $R$ ). Finally we will prove that the star generator is equal to the scalar  $\frac{1}{2}$  in  $ZH_R$ .



**Lemma 7.9.** If  $\frac{1}{2} \in R$ , then  $\star = \boxed{\frac{1}{2}}$ .

*Proof.*

$$\boxed{\frac{1}{2}} \stackrel{2.3}{=} \star \circ \boxed{\frac{1}{2}} \stackrel{7.7}{=} \star \boxed{2} \boxed{\frac{1}{2}} \stackrel{(34)}{=} \star \boxed{1} \stackrel{(34)}{=} \star \quad \blacksquare$$

With the correspondence to the standard ZH-calculus now firmly established, we can easily adapt our previous results to prove that  $ZH_R$  is complete.

**Theorem 7.10.** Let  $R$  be a commutative unital ring with  $\frac{1}{2} \in R$ . Then the  $ZH_R$ -calculus is complete, i.e. for any  $ZH_R$ -diagrams  $D_1$  and  $D_2$ , if  $\llbracket D_1 \rrbracket = \llbracket D_2 \rrbracket$ , then  $D_1$  can be transformed into  $D_2$  using the rules of Figure 3.

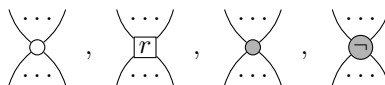
*Proof.* As the rules of Figure 3 are a superset of those of Figure 1, anything we have proven for ZH remains provable in  $ZH_R$ . We have also shown in Proposition 7.8 that all the labelled (ZH) H-boxes used in the completeness proof are equivalent to their  $ZH_R$ -calculus counterparts. In particular, the reduction to normal form conditional on having proved  $(M_{a,b})$ ,  $(I_a)$  and  $(A_{a,b})$  of Section 4.2 remains valid. But now, instead of having to prove the multiplication, introduction and average rules, we have them assumed as axioms. Thus, we can transform any  $ZH_R$ -diagram into reduced normal form, and hence the calculus is complete. ■

## 7.2 The ZH-calculus over rings without a half

The  $ZH_R$ -calculus when  $R$  does not contain a half is slightly trickier to deal with. As before, we will assume that  $R$  is a commutative unital ring. But instead of assuming that  $\frac{1}{2} \in R$ , we will merely assume that 2 is not a zero-divisor.

With these conditions,  $R$  embeds faithfully into  $R[\frac{1}{2}]$ , and so  $ZH_R$  as defined in Section 7.1 has a non-universal interpretation into  $R[\frac{1}{2}]$ -bit. We could then add the  $\star$  generator creating a universal interpretation into  $R[\frac{1}{2}]$ -bit. This is essentially the process one follows to go from  $ZH_Z$  to our ‘phase-free’ ZH, but also raises the question of ‘why not just start with  $ZH_{R[\frac{1}{2}]}$  instead?’ We present in this subsection a definition of  $ZH_R$ , with  $\frac{1}{2} \notin R$ , that is universal for  $R$ -bit and is complete using the same rules as  $ZH_R$  when  $\frac{1}{2} \in R$ , with the exception of an additional ‘scalar cancellation’ meta-rule.

Since we want to have an interpretation into  $R$ -bit we will no longer be able to use the  $\star$  generator, which in turn makes the definition of the derived generators (X) and (NOT) invalid. Instead we will promote the grey spiders from being *derived* generators to, simply, generators. Hence, our list of generators becomes:



The ruleset of the  $ZH_R$ -calculus when  $\frac{1}{2} \notin R$  consists of the rules of Figure 3, where now the grey spiders are considered to be actual generators, and the usual symmetries of the generators. In addition, as the grey spiders are now no longer defined in terms of the other generators, we introduce the following rules to relate the generators to one another, which are scaled versions of the definitions of the grey spiders (1):

$$\circ \text{ (white spider) } = \text{ (white spider with boxes) } \quad (2X)$$

$$\circ \text{ (grey spider) } = \text{ (grey spider with boxes) } \quad (2NOT)$$

Finally, we will also need the following cancellation meta-rule to get completeness when  $\frac{1}{2} \notin R$ :

$$\circ D_1 = \circ D_2 \implies D_1 = D_2 \quad (2Cancel)$$

In words, this rule says that if two diagrams are provably equal in  $ZH_R$  and they both contain a scalar white dot, then we can cancel the white dot on both sides and retain an equality. (When  $\star$  is in the calculus, this follows easily by an application of Lemma 2.3.) We call this implication a meta-rule because it is a statement about admissible deductive logic rather than diagram equality. Note that since 2 is not a zero-divisor in  $R$  this rule is sound. A similar meta-rule was also used for proving completeness of rational-angle fragments of the ZX calculus [30].

The plan for showing completeness of  $ZH_R$  when  $\frac{1}{2} \notin R$  is to show that the  $ZH_{R[\frac{1}{2}]}$ -derivation from a  $ZH_R$ -diagram  $D$  to its normal form can be used as a  $ZH_R$ -derivation, up to rescaling. While this idea is straightforward, formalising it takes some care.

**Remark 7.11.** Every  $ZH_R$  diagram can be seen as a  $ZH_{R[\frac{1}{2}]}$  diagram by translating each diagram component to their obvious counterpart. For grey spiders we need to be careful about this translation, since they are generators in  $ZH_R$ , but derived generators in  $ZH_{R[\frac{1}{2}]}$ . We will be using this embedding in the lemmas that follow, noting that we will still explicitly ‘unpack’ grey spiders using the rules (NOT), (X), (2NOT) and (2X) when we need to (and not leave such unpacking implicit). We are treating the  $\star$  element as syntactic sugar to represent the 0-arity  $H(\frac{1}{2})$  box when it arises.

**Lemma 7.12.** Suppose  $D_1$  and  $D_2$  are  $ZH_R$ -diagrams. Let  $*$  denote either of the  $ZH_{R[\frac{1}{2}]}$ -rules (NOT) or (X), with  $2*$  denoting the corresponding rescaled  $ZH_R$  version (2NOT) or (2X). Then if the  $ZH_{R[\frac{1}{2}]}$  rule application

$$* \\ D_1 = \star D_2$$

is a valid diagrammatic rewrite, so is the  $ZH_R$  rule application

$$2* \\ \circ D_1 = D_2$$

*Proof.* This follows immediately from the definitions. ■

**Definition 7.13.** We say that a  $ZH_{R[\frac{1}{2}]}$  diagram is a  $ZH_R^\star$  diagram (pronounced ‘ZHR with stars’ diagram) if it is of the form  $D \otimes (\star)^{\otimes n}$ , where  $D$  is a  $ZH_R$  diagram.

Recall that the rules of  $ZH_{R[\frac{1}{2}]}$  are the rules given in Figure 3 and the rules relating the generators to derived generators: (X), (NOT), and (Z).

**Lemma 7.14.** The only  $ZH_{R[\frac{1}{2}]}$  rules that can be applied to a  $ZH_R^\star$  diagram to give a diagram that is no longer a  $ZH_R^\star$  diagram are (M), (A), and (HS).

*Proof.* The (M) and (A) rules (applied right-to-left) can both result in H-box phases that are no longer in  $R$ . Furthermore, recall from Remark 7.11 that the star element is used here as shorthand for the 0-arity H-box labelled by  $\frac{1}{2}$ . Hence  $\frac{1}{2}$ -labelled H-boxes of arity 0 are allowed in  $ZH_R^\star$  diagrams, while  $\frac{1}{2}$ -labelled H-boxes of higher arity are not allowed. Yet the (HS) rule, when applied right-to-left to a  $\frac{1}{2}$ -labelled H-box of arity 0, yields a  $\frac{1}{2}$ -labelled H-box with arity greater than 0.

All other rules preserve  $ZH_R^\star$  diagrams: the only remaining rules that affect H-box labels are (U), which can only introduce the integer label 1, and (I), which does not change any labels and does not match H-boxes of arity 0. The other rules do not involve H-box labels at all. ■

**Lemma 7.15.** The  $ZH_{R[\frac{1}{2}]}$  derivation from a  $ZH_R$  diagram  $D$  to its normal form  $N$  of Theorem 7.10 is a  $ZH_R^\star$  diagram at every step of the derivation.

*Proof.* The starting diagram  $D$  is a  $ZH_R$  diagram (and therefore also a  $ZH_R^\star$  diagram). Each step of the derivation to its normal form  $N$  is a rule application, and we will show that every rule application used preserves the condition of being a  $ZH_R^\star$  diagram. Intuitively, this is because the process for rewriting to normal form was derived in the phase-free ZH-calculus, which differs from  $ZH_Z^\star$  only in that grey spiders are syntactic sugar rather than generators.

More formally: by Lemma 7.14 the only rule applications that could result in a diagram that is no longer in  $ZH_R^\star$  form are (M), (A), and (HS). As can be verified by going through the steps

of the derivation in Section 4, the derivation from  $D$  to  $N$  only uses (M) and (A) with phases that are elements of  $R$ , and hence they preserve the  $\text{ZH}_R^\star$  form.

Also note that at no point in the derivation is the rule (HS) applied to a star, since in ZH that would result in an invalid  $\frac{1}{2}$ -phased H-box. ■

The previous lemma shows that the derivation from a  $\text{ZH}_R$ -diagram  $D$  to its normal form in  $\text{ZH}_{R[\frac{1}{2}]}$  consists of diagrams that are close to being  $\text{ZH}_R$ -diagrams, except for the star generators. We shall next show that the only rules that interact with  $\star$  are (NOT) and (X), which are precisely the rules we replace when using  $\text{ZH}_R$  instead of  $\text{ZH}_{R[\frac{1}{2}]}$ .

**Lemma 7.16.** In the  $\text{ZH}_{R[\frac{1}{2}]}$  derivation from a  $\text{ZH}_R$  diagram  $D$  to its normal form  $N$  of Theorem 7.10, the element  $\star$  is only ever involved in a rewrite when the rule being applied is (NOT) or (X).

*Proof.* The only  $\text{ZH}_{R[\frac{1}{2}]}$  rules that interact with the  $\star$  element are (NOT), (X), and (HS) (viewing the star as a 0-arity H-box). As noted in the proof of Lemma 7.15 the (HS) rule is never applied to a star in the course of the derivation, since the derivation was designed to be applied to ZH diagrams where such a rewrite would be invalid. ■

**Lemma 7.17.** If the  $\text{ZH}_R$  diagram  $D$  has normal form  $N$ , then there is a  $m \in \mathbb{N}$  such that the equation  $D \otimes (\circ)^{\otimes m} = N \otimes (\circ)^{\otimes m}$  is derivable in  $\text{ZH}_R$ .

*Proof.* By Lemma 7.15 there is a  $\text{ZH}_{R[\frac{1}{2}]}$ -derivation from  $D$  to  $N$  such that every diagram in the chain is a  $\text{ZH}_R^\star$  diagram. As noted in Remark 7.11, we are keeping the unpacking of grey spiders explicit, and so applications of the  $\text{ZH}_{R[\frac{1}{2}]}$ -rules (NOT) and (X) are explicit steps in this derivation.

We then simultaneously compose every diagram in the derivation by the same number of  $\circ$  generators such that by Lemma 7.12 we can replace all applications of (NOT) with (2NOT) and all applications of (X) with (2X), and then cancel all remaining instances of a  $\star$  with a  $\circ$ . This leaves us with a sound  $\text{ZH}_R$ -derivation of  $\text{ZH}_R$  diagrams from  $D \otimes (\circ)^{\otimes m}$  to  $N \otimes (\circ)^{\otimes m}$  for some  $m \geq 0$ . ■

**Theorem 7.18.** Let  $R$  be a commutative unital ring where 2 is a zero-divisor and  $\frac{1}{2} \notin R$ . Then the  $\text{ZH}_R$ -calculus is complete. In other words, for any  $\text{ZH}_R$ -diagrams  $D_1$  and  $D_2$ , if  $\llbracket D_1 \rrbracket = \llbracket D_2 \rrbracket$  then  $D_1$  can be transformed into  $D_2$  using the rules of Figure 3, the rules (2NOT) and (2X), and the meta-rule (2Cancel).

*Proof.* The completeness of  $\text{ZH}_{R[\frac{1}{2}]}$  gives us the  $\text{ZH}_{R[\frac{1}{2}]}$ -derivations  $D_1 = N$  and  $N = D_2$ . By Lemma 7.17 there are  $\text{ZH}_R$ -derivations  $(\circ)^{\otimes a} \otimes D_1 = (\circ)^{\otimes a} \otimes N$  and  $(\circ)^{\otimes b} \otimes N = (\circ)^{\otimes b} \otimes D_2$  for some  $a, b \in \mathbb{N}$ . We can therefore derive  $(\circ)^{\otimes a+b} \otimes D_1 = (\circ)^{\otimes a+b} \otimes N = (\circ)^{\otimes a+b} \otimes D_2$  in  $\text{ZH}_R$  by multiplying every diagram in the derivations by a suitable number of  $\otimes$ -products of  $\circ$ .

By the (2Cancel) meta-rule we can then assert that  $D_1 = D_2$  is provable in  $\text{ZH}_R$ . ■

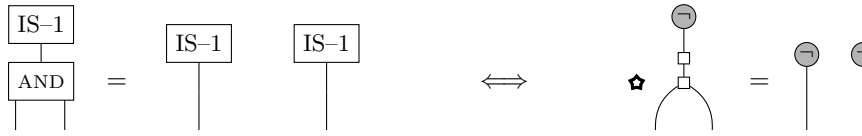
We have now shown that  $\text{ZH}_R$  is universal, sound, and complete whenever  $R$  is a commutative, unital ring where 2 is not a zero-divisor. In the case  $\frac{1}{2} \in R$ , the calculus and ruleset is very similar to our basic ZH. When  $\frac{1}{2} \notin R$  we needed to tweak the generators to avoid using the  $\star$  generator, and we needed to introduce a meta-rule for cancelling  $\circ$  generators.

## 8 Modifications to the ZH-calculus

In this section, we will discuss a couple of variations on the ZH-calculus. In particular, we present an alternative to (o) in Section 8.1, and an alternative to (A) and (I) in Section 8.3. We show that (hh) can actually be proved from the other rules in Section 8.2. In Section 8.4 we present an alternative parameter-free ZH-calculus which is complete over a subring of matrices over  $\mathbb{Z}[\frac{1}{\sqrt{2}}]$  which exactly corresponds to those that can be implemented by the Toffoli+Hadamard gate set.

### 8.1 Replacing the (o) rule

Of all the basic rules of the ZH-calculus in Figure 1 the ortho rule (o) seems a bit out of place. In this section we will see that we can replace it with two other rules, namely Lemmas 2.28 and 5.1. These two rules can be rephrased as statements about the AND gate. Namely, up to some simple rewriting, Lemma 2.28 says that if we post-select the AND gate with  $\langle 1|$  that this post-selection copies through:



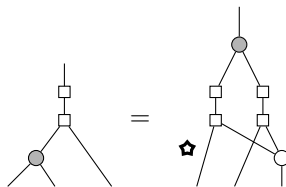
Lemma 5.1 can be rephrased as stating that a COPY gate followed by an AND gate applied to its outputs is just the identity:



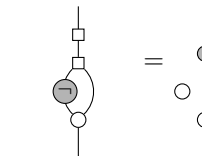
Lemmas 2.28 and 5.1 show that these two rules can be proved by the basic ZH-calculus rules including (o). In this section we will show the converse: that assuming just the rules of Figure 1 except for (o), together with the Lemmas 2.28 and 5.1 we can prove (o). Note that all the basic lemmas of Section 2.4 are proven without the usage of (o) (except for Lemma 2.28), and hence can be used in this section.

Before proving (o), we need to prove a number of lemmas that are essentially statements in Boolean logic.

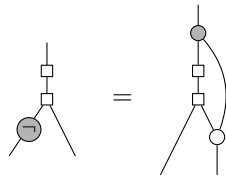
**Lemma 8.1.**  $(x \oplus y) \wedge z = (x \wedge z) \oplus (y \wedge z)$ :



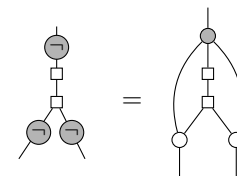
**Lemma 8.3.**  $(\neg x) \wedge x = 0$ :



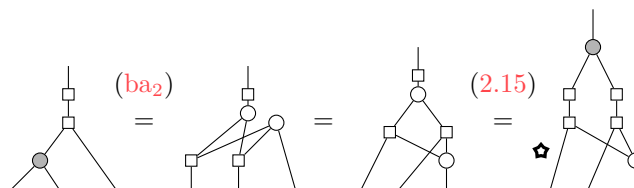
**Lemma 8.2.**  $(\neg x) \wedge y = (x \wedge y) \oplus y$ :



**Lemma 8.4.**  $x \vee y = x \oplus (x \wedge y) \oplus y$ :

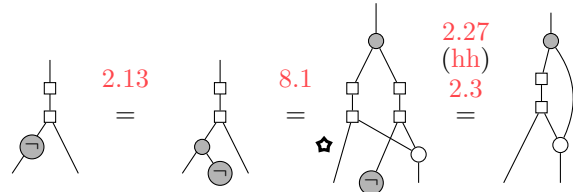


*Proof of Lemma 8.1.*



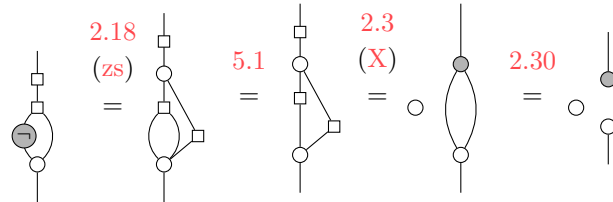
■

Proof of Lemma 8.2.



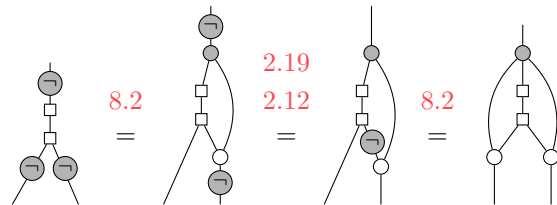
■

Proof of Lemma 8.3.



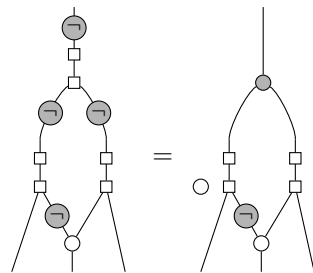
■

Proof of Lemma 8.4.

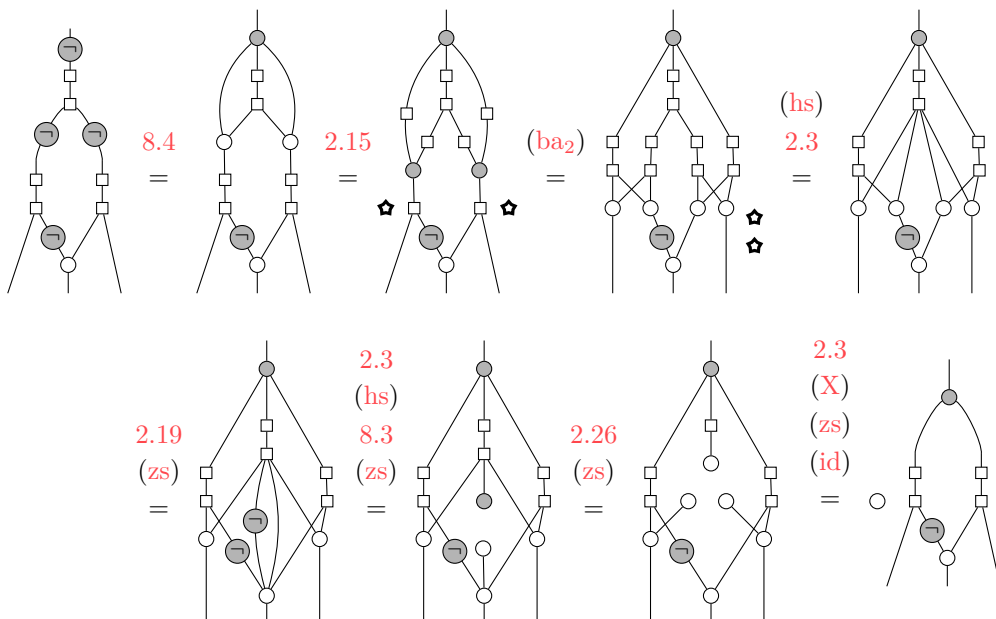


■

**Lemma 8.5.** We have  $(x \wedge (\neg y)) \vee (y \wedge z) = (x \wedge (\neg y)) \oplus (y \wedge z)$ :



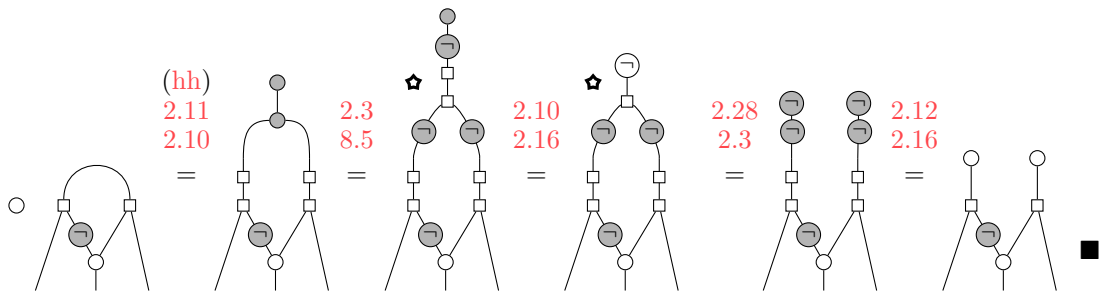
Proof.



■

**Theorem 8.6.** In the presence of the other rules of the ZH-calculus, (o) is equivalent to the union of Lemmas 2.28 and 5.1.

*Proof.* The proof of Lemmas 2.28 and 5.1 shows the forward direction: that the standard ruleset of the ZH-calculus, including (o), implies the two lemmas. For the converse direction, recall that the only one of the basic derived rules in Section 2.4 whose proof required (o) was Lemma 2.28, which is now one of our axioms. Note that we have proven Lemma 8.5 using just Lemma 5.1, ZH-calculus axioms other than (o), and the basic derived rules. Proving (o) from Lemma 8.5, other ZH axioms, and basic derived rules will therefore give the desired result. To make the application of intermediate rules clearer, we re-arrange the diagrams of (o) so the wire which is usually an output is now the middle input instead:



### 8.2 The (hh) rule is not necessary

It turns out that (hh) can actually be derived from the other rules. Before proving this, since the proof of Lemma 2.3 uses (hh), we will need to find a different way to cancel scalars first. The following turns out to suffice:

$$\begin{array}{c} \square \end{array} \stackrel{\text{(id)}}{=} \begin{array}{c} \square \\ | \\ \circ \end{array} \stackrel{\text{(ba}_2\text{)}}{=} \begin{array}{c} \bullet \\ | \\ \square \end{array} \stackrel{\text{(X)}}{=} \begin{array}{c} \square \\ | \\ \star \\ | \\ \square \end{array} \stackrel{\text{(id)}}{=} \begin{array}{c} \star \\ | \\ \square \end{array} \stackrel{\text{(hs)}}{=} \begin{array}{c} \circ \\ | \\ \square \end{array} \tag{35}$$

We can now prove (hh):

$$\begin{array}{c} \circ \end{array} \Big| \stackrel{\text{(id)}}{=} \begin{array}{c} \circ \\ | \\ \circ \end{array} \stackrel{\text{(ba}_1\text{)}}{=} \begin{array}{c} \bullet \\ | \\ \circ \end{array} \stackrel{\text{(ba}_1\text{)}}{=} \begin{array}{c} \bullet \\ | \\ \bullet \\ | \\ \circ \end{array} \stackrel{\text{(id)}}{=} \begin{array}{c} \bullet \\ | \\ \bullet \\ | \\ \circ \end{array} \stackrel{\text{(X)}}{=} \begin{array}{c} \star \\ | \\ \bullet \\ | \\ \star \\ | \\ \circ \end{array} \stackrel{\text{(hs)}}{=} \begin{array}{c} \star \\ | \\ \circ \\ | \\ \star \\ | \\ \circ \end{array} \stackrel{\text{(35)}}{=} \begin{array}{c} \square \end{array} \tag{36}$$

The fact that we can derive (hh) from the other rules raises the question whether any other rules are also not necessary. We hypothesise that they are in fact all needed. It is at least the case that (zs) and (hh) are necessary: they are the only rules that relate higher-arity spiders, respectively H-boxes, to lower arity ones (this argument can be formalised by presenting alternative interpretations into linear maps that change what higher-arity spiders/H-boxes correspond to). (id) is also necessary as it is the only rule that relates a generator to the identity wire. At least one of (ba<sub>1</sub>) and (ba<sub>2</sub>) is necessary as they are the only ones that relate an empty diagram to a non-empty diagram (the  $n = m = 0$  case). We do not know of any argument for the necessity of the other rules (m) and (o), although it seems likely that they are both necessary.

### 8.3 Merging the intro and average rule

The intro rule (I) and the average rule (A) can be subsumed by a single larger rule. Before we present this rule, let us first present an alternative to (A) that holds when the ring contains a half.

**Lemma 8.7.** Let  $R$  be a ring with a half. In the presence of the other rules in Figure 3, (A) is equivalent in  $ZH_R$  to the following rule:

$$\begin{array}{c} \circ \\ | \\ \boxed{a} \quad \boxed{b} \\ | \\ \ominus \end{array} = \boxed{2} \quad \begin{array}{c} | \\ \boxed{\frac{a+b}{2}} \end{array} \tag{37}$$

*Proof.* As the LHS of (A) agrees with the LHS of (37), it suffices to show that the RHS of both are equal. Note first that:

$$\begin{array}{c} \ominus \\ | \\ \boxed{2} \end{array} \stackrel{\text{(U)}}{=} \begin{array}{c} \circ \\ | \\ \boxed{\frac{1}{2}} \quad \boxed{2} \quad \boxed{2} \quad \ominus \end{array} \stackrel{7.4}{=} \begin{array}{c} \circ \\ | \\ \boxed{\frac{1}{2}} \end{array} \tag{38}$$

Hence:

$$\begin{array}{c} \circ \\ | \\ \boxed{a+b} \quad \boxed{2} \quad \ominus \end{array} \stackrel{\text{(38)}}{=} \begin{array}{c} \circ \\ | \\ \boxed{a+b} \quad \boxed{\frac{1}{2}} \end{array} \stackrel{\text{(m)}}{=} \begin{array}{c} \circ \\ | \\ \boxed{\frac{a+b}{2}} \end{array} \tag{39}$$

Note that this only shows that (A) implies (37), as we proved Lemma 7.4 using (A).

For the converse direction, we have

$$\begin{array}{c} \boxed{2} \quad \begin{array}{c} | \\ \boxed{\frac{a+b}{2}} \end{array} \end{array} \stackrel{\text{(id)}}{=} \begin{array}{c} \ominus \\ | \\ \boxed{2} \quad \begin{array}{c} \circ \\ | \\ \boxed{\frac{a+b}{2}} \end{array} \end{array} \stackrel{\text{(zs)}}{=} \begin{array}{c} \ominus \\ | \\ \boxed{2} \quad \begin{array}{c} \circ \\ | \\ \boxed{2} \quad \begin{array}{c} \circ \\ | \\ \boxed{\frac{a+b}{2}} \end{array} \end{array} \end{array} \stackrel{\text{(I)}}{=} \begin{array}{c} \ominus \\ | \\ \boxed{2} \quad \begin{array}{c} \ominus \\ | \\ \boxed{2} \quad \begin{array}{c} \circ \\ | \\ \boxed{\frac{a+b}{2}} \end{array} \end{array} \end{array} \stackrel{\text{(zs)}}{=} \begin{array}{c} \ominus \\ | \\ \boxed{2} \quad \begin{array}{c} \ominus \\ | \\ \boxed{2} \quad \begin{array}{c} \circ \\ | \\ \boxed{\frac{a+b}{2}} \end{array} \end{array} \end{array} \stackrel{2.27}{=} \begin{array}{c} \ominus \\ | \\ \boxed{2} \quad \begin{array}{c} \circ \\ | \\ \boxed{2} \quad \begin{array}{c} \circ \\ | \\ \boxed{\frac{a+b}{2}} \end{array} \end{array} \end{array} \stackrel{2.27}{=} \begin{array}{c} \ominus \\ | \\ \boxed{2} \quad \begin{array}{c} \circ \\ | \\ \boxed{2} \quad \begin{array}{c} \circ \\ | \\ \boxed{\frac{a+b}{2}} \end{array} \end{array} \end{array} \stackrel{\text{(M)}}{=} \begin{array}{c} \ominus \\ | \\ \boxed{2} \quad \boxed{a+b} \end{array}$$

which implicitly uses a straightforward generalisation of Lemma 2.27 to H-boxes with arities other than 3 and with phase labels. None of the lemmas invoked in this proof, nor their antecedents, rely on (A). ■

**Theorem 8.8** ([48]). Let  $R$  be a ring with a half. In the presence of the other rules of Figure 3, the combination of the rules (I) and (A) is equivalent to the following rule:

$$\begin{array}{c} \square \\ | \\ \boxed{a} \quad \boxed{0} \quad \boxed{b} \\ | \\ \ominus \end{array} = \begin{array}{c} \circ \\ | \\ \boxed{a+b} \quad \boxed{a-b} \\ | \\ \ominus \end{array} \tag{39}$$

*Proof.* As the  $ZH_R$ -calculus is complete with the rules presented in Figure 3, and (39) is easily shown to be sound for any  $a$  and  $b$ , the equation (39) can be proven using (I) and (A) for any particular choice of  $a$  and  $b$ . Alternatively, by using the ZH-calculus over the polynomial ring in two variables  $R[a, b]$ , there is also a proof of it using the variables  $a$  and  $b$  directly.

It hence remains to show that (39) together with (Figure 3)  $\setminus \{(A), (I)\}$  implies both (I) and (A). We will first prove (37). By plugging a white dot into the top wire of (39) we get this, except for a scalar factor that will be shown to cancel afterwards:

$$\begin{array}{c} \circ \\ | \\ \boxed{a+b} \quad \boxed{a-b} \\ | \\ \ominus \end{array} \stackrel{\text{(id)}}{=} \begin{array}{c} \circ \\ | \\ \boxed{a+b} \quad \boxed{a-b} \\ | \\ \ominus \end{array} \stackrel{\text{(39)}}{=} \begin{array}{c} \square \\ | \\ \boxed{a} \quad \boxed{0} \quad \boxed{b} \\ | \\ \ominus \end{array} \stackrel{\text{(X)}}{=} \begin{array}{c} \square \\ | \\ \boxed{a} \quad \boxed{0} \quad \boxed{b} \\ | \\ \ominus \end{array} \stackrel{2.21}{=} \begin{array}{c} \square \\ | \\ \boxed{a} \quad \boxed{0} \quad \boxed{b} \\ | \\ \ominus \end{array} \stackrel{\text{(HS)}}{=} \begin{array}{c} \square \\ | \\ \boxed{a} \quad \boxed{0} \quad \boxed{b} \\ | \\ \ominus \end{array} \stackrel{2.26}{=} \begin{array}{c} \square \\ | \\ \boxed{a} \quad \boxed{0} \quad \boxed{b} \\ | \\ \ominus \end{array} \stackrel{\text{(X)}}{=} \begin{array}{c} \square \\ | \\ \boxed{a} \quad \boxed{0} \quad \boxed{b} \\ | \\ \ominus \end{array} \stackrel{\text{(zs)}}{=} \begin{array}{c} \circ \\ | \\ \boxed{0} \quad \boxed{b} \\ | \\ \ominus \end{array} \stackrel{\text{(id)}}{=} \begin{array}{c} \circ \\ | \\ \boxed{0} \quad \boxed{b} \\ | \\ \ominus \end{array} \stackrel{\text{(M)}}{=} \begin{array}{c} \circ \\ | \\ \boxed{0} \\ | \\ \ominus \end{array} \tag{40}$$





represent matrices  $M = (\frac{1}{\sqrt{2}})^n A$  where  $A$  is a matrix over  $\mathbb{Z}$ , which correspond precisely to the matrices generated by the Toffoli+Hadamard gate set [3]. We do this by redefining the  $\star$  generator to represent  $\frac{1}{\sqrt{2}}$  instead of  $\frac{1}{2}$ . By replacing every occurrence of  $\star$  by two stars in the definitions of Section 2, and in all the derivations of the preceding sections, all the derivations remain sound under this new interpretation. In particular, the derivation that each diagram can be reduced to normal form remains true. For almost any diagram we can also bring the diagram to reduced normal form, with one class of exceptions: if the diagram represents the zero matrix then the reduced normal form should contain no stars. However, when we reinterpret the star (and thus double the number of occurrences in the rewrite rules) we see that all rewrite rules preserve the parity of the number of stars in the diagram. Hence, the following equality that says that  $\frac{1}{\sqrt{2}} \cdot 0 = 0$  is not derivable with those rules:

$$\text{Using } \llbracket \star \rrbracket = \frac{1}{\sqrt{2}} \quad \begin{array}{c} \circ \\ | \\ \square \end{array} \star = \begin{array}{c} \circ \\ | \\ \square \end{array} \quad (44)$$

Adding this as another rewrite rule solves this edge case and we still get a complete calculus, but now for matrices of the form  $(\frac{1}{\sqrt{2}})^n A$  where  $A$  only contains integers.

## 9 Conclusion

We studied the ZH-calculus, a graphical language for reasoning about qubit quantum computing. We found a small set of rewrite rules motivated by basic identities between Boolean functions that allowed us to prove completeness of the fragment that corresponds to the Toffoli-Hadamard gateset. We then extended the ZH-calculus so that it can represent matrices over arbitrary rings where 2 is not a zero divisor. We found an extended ruleset that is complete for any such ring. We have argued that our calculus, both the parameter-free one, as well as the one complete over any ring, is the simplest complete graphical calculus for an approximately universal fragment of quantum computing found so far.

Of the rules of the ZH-calculus as presented in Figure 1, we showed that (hh) is actually not necessary, as it can be proven from the others. We believe that the rest are all necessary for completeness. In Section 8.1 we showed that (o) can be replaced by two smaller and simpler to understand rules. It might perhaps be possible to show that only one of these suffices, which would give us a simpler axiomatisation. For the  $ZH_R$ -calculus we required a meta-rule to cancel scalars when  $\frac{1}{2} \notin R$ . It is not clear at the moment whether the calculus is complete without this meta-rule.

The phase-free calculus with its translation to Boolean functions elucidates the relationship between classical computation and universal quantum computation using the Toffoli-Hadamard gate set. Correspondingly, the ZH-calculus over arbitrary rings should be useful for analysing universal quantum computation with multiply-controlled gates as well as quantum computation or error-correction schemes involving hypergraph states. The Fourier-transform relationship between the ZH-calculus over  $\mathbb{C}$  and the ZX-calculus [36] also points towards the usefulness of combining the ZX- and ZH-calculus, and translating between them. Indeed, [17] already exhibits a combined calculus called the ZXH-calculus. Whether on its own or in combination with other graphical languages, the ZH-calculus has applications in a broad range of areas of quantum computing.

**Acknowledgements** The authors wish to thank Renaud Vilmart for pointing out (39) as an alternative to the intro and average rule, and Patrick Roy for pointing out the derivation (36) that showed that (hh) is superfluous. The authors would also like to extend their gratitude to the anonymous reviewers that read the paper closely and found several small errors and oversights that we could subsequently fix. JvdW and AK are supported in part by AFOSR grant FA2386-18-1-4028. For the majority of the work in this article, JvdW is supported by a Rubicon fellowship financed by the Dutch Research Council (NWO). Additionally, JvdW acknowledges that this project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101018390.

## References

- [1] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004. <https://doi.org/10.1103/PhysRevA.70.052328>. 3
- [2] Matthew Amy. Towards large-scale functional verification of universal quantum circuits. In Peter Selinger and Giulio Chiribella, editors, *Proceedings of the 15th International Conference on Quantum Physics and Logic, Halifax, Canada, 3-7th June 2018*, volume 287 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–21. Open Publishing Association, 2019. <https://doi.org/10.4204/EPTCS.287.1>. 4
- [3] Matthew Amy, Andrew N. Glauddell, and Neil J. Ross. Number-Theoretic Characterizations of Some Restricted Clifford+T Circuits. *Quantum*, 4:252, 4 2020. ISSN 2521-327X. <https://doi.org/10.22331/q-2020-04-06-252>. 8, 63, 64
- [4] Miriam Backens. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*, 16(9):093021, 2014. <https://doi.org/10.1088/1367-2630/16/9/093021>. 3
- [5] Miriam Backens and Aleks Kissinger. ZH: A complete graphical calculus for quantum computations involving classical non-linearity. In Peter Selinger and Giulio Chiribella, editors, *Proceedings of the 15th International Conference on Quantum Physics and Logic, Halifax, Canada, 3-7th June 2018*, volume 287 of *Electronic Proceedings in Theoretical Computer Science*, pages 23–42. Open Publishing Association, 2019. <https://doi.org/10.4204/EPTCS.287.2>. 4, 33, 52
- [6] Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski, and John van de Wetering. There and back again: A circuit extraction tale. *Quantum*, 5:421, 3 2021. ISSN 2521-327X. <https://doi.org/10.22331/q-2021-03-25-421>. 2
- [7] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179. IEEE, 1984. 3
- [8] Titouan Carette. When Only Topology Matters. 2021. <https://doi.org/10.48550/arXiv.2102.03178>. 8
- [9] Titouan Carette and Emmanuel Jeandel. A Recipe for Quantum Graphical Languages. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 118:1–118:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. ISBN 978-3-95977-138-2. <https://doi.org/10.4230/LIPIcs.ICALP.2020.118>. 4
- [10] Nicholas Chancellor, Aleks Kissinger, Joschka Roffe, Stefan Zohren, and Dominic Horsman. Graphical structures for design and verification of quantum error correction. 2016. <https://doi.org/10.48550/arXiv.1611.08012>. 2
- [11] Bob Coecke and Ross Duncan. Interacting quantum observables. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science, 2008. [https://doi.org/10.1007/978-3-540-70583-3\\_25](https://doi.org/10.1007/978-3-540-70583-3_25). 2
- [12] Bob Coecke and Ross Duncan. Interacting quantum observables: Categorical algebra and diagrammatics. *New Journal of Physics*, 13:043016, 2011. <https://doi.org/10.1088/1367-2630/13/4/043016>. 2
- [13] Bob Coecke and Aleks Kissinger. The compositional structure of multipartite quantum entanglement. In *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 297–308. Springer, 2010. [https://doi.org/10.1007/978-3-642-14162-1\\_25](https://doi.org/10.1007/978-3-642-14162-1_25). Extended version: arXiv:1002.2540. 4
- [14] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017. <https://doi.org/10.1017/9781316219317>. 2, 3, 8, 10
- [15] Bob Coecke, Dusko Pavlovic, and Jamie Vicary. A new description of orthogonal bases. *Mathematical Structures in Computer Science*, 23(3):555–567, 2013. <https://doi.org/10.1017/S0960129512000047>. 8
- [16] Cole Comfort. The ZX-calculus: A complete graphical calculus for classical circuits using spiders. In Benoît Valiron, Shane Mansfield, Pablo Arrighi, and Prakash Panangaden, editors, *Proceedings 17th International Conference on Quantum Physics and Logic, Paris, France*,

- June 2 - 6, 2020, volume 340 of *Electronic Proceedings in Theoretical Computer Science*, pages 60–90. Open Publishing Association, 2021. <https://doi.org/10.4204/EPTCS.340.4>. 4, 10
- [17] Richard D. P. East, John van de Wetering, Nicholas Chancellor, and Adolfo G. Grushin. AKLT-states as ZX-diagrams: diagrammatic reasoning for quantum states. *PRX Quantum*, 3:010302, Jan 2022. <https://doi.org/10.1103/PRXQuantum.3.010302>. 4, 64
- [18] Niel de Beaudrap and Dominic Horsman. The ZX calculus is a language for surface code lattice surgery. *Quantum*, 4, 2020. <https://doi.org/10.22331/q-2020-01-09-218>. 2
- [19] Ross Duncan and Simon Perdrix. Rewriting measurement-based quantum computations with generalised flow. In *Proceedings of ICALP*, Lecture Notes in Computer Science, pages 285–296. Springer, 2010. [https://doi.org/10.1007/978-3-642-14162-1\\_24](https://doi.org/10.1007/978-3-642-14162-1_24). 2
- [20] Ross Duncan, Aleks Kissinger, Simon Perdrix, and John van de Wetering. Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus. *Quantum*, 4:279, 6 2020. ISSN 2521-327X. <https://doi.org/10.22331/q-2020-06-04-279>. 2
- [21] Richard D. P. East, Pierre Martin-Dussaud, and John van de Wetering. Spin-networks in the ZX-calculus. 2021. <https://doi.org/10.48550/arXiv.2111.03114>. 4
- [22] Andrew Fagan and Ross Duncan. Optimising Clifford circuits with Quantomatic. In Peter Selinger and Giulio Chiribella, editors, *Proceedings of the 15th International Conference on Quantum Physics and Logic (QPL)*, volume 287 of *Electronic Proceedings in Theoretical Computer Science*, pages 85–105. Open Publishing Association, 2019. <https://doi.org/10.4204/EPTCS.287.5>. 2
- [23] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012. <https://doi.org/10.1103/PhysRevA.86.032324>. 3
- [24] Craig Gidney and Austin G. Fowler. Efficient magic state factories with a catalyzed  $|CCZ\rangle$  to  $2|T\rangle$  transformation. *Quantum*, 3:135, 4 2019. ISSN 2521-327X. <https://doi.org/10.22331/q-2019-04-30-135>. 2
- [25] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation. In *Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics*, volume 68, pages 13–58, 2010. <https://doi.org/10.1090/psapm/068/2762145>. 3
- [26] 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. <https://doi.org/10.1109/LICS.2015.59>. 4, 11
- [27] Amar Hadzihasanovic. *The algebra of entanglement and the geometry of composition*. DPhil thesis, University of Oxford, 2017. <https://doi.org/10.48550/arXiv.1709.08086>. 52
- [28] Michael Hanks, Marta P. Estarellas, William J. Munro, and Kae Nemoto. Effective compression of quantum braided circuits aided by ZX-calculus. *Physical Review X*, 10:041030, 2020. <https://doi.org/10.1103/PhysRevX.10.041030>. 2
- [29] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A complete axiomatisation of the ZX-calculus for Clifford+T quantum mechanics. In *33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 559–568, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5583-4. <https://doi.org/10.1145/3209108.3209131>. 3, 11, 52
- [30] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A generic normal form for ZX-diagrams and application to the rational angle completeness. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, LICS '19, pages 1–10. ACM, 2019. <https://doi.org/10.1109/LICS.2019.8785754>. 57
- [31] Aleks Kissinger and John van de Wetering. Universal MBQC with generalised parity-phase interactions and Pauli measurements. *Quantum*, 3:134, 4 2019. ISSN 2521-327X. <https://doi.org/10.22331/q-2019-04-26-134>. 2
- [32] Aleks Kissinger and John van de Wetering. Reducing the number of non-Clifford gates in quantum circuits. *Physical Review A*, 102:022406, 8 2020. <https://doi.org/10.1103/PhysRevA.102.022406>. 2
- [33] Aleks Kissinger and John van de Wetering. Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions. *Quantum Science and Technology*, 2022. <https://doi.org/10.1088/2058-9565/ac5d20>. 2

- [34] Aleks Kissinger, Alex Merry, and Matvey Soloviev. Pattern graph rewrite systems. *Electronic Proceedings in Theoretical Computer Science*, 143:54–66, 3 2014. ISSN 2075-2180. <https://doi.org/10.4204/EPTCS.143.5>. 18
- [35] Aleks Kissinger, John van de Wetering, and Renaud Vilmart. Classical simulation of quantum circuits with partial and graphical stabiliser decompositions. In François Le Gall and Tomoyuki Morimae, editors, *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, volume 232 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:13, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-237-2. <https://doi.org/10.4230/LIPIcs.TQC.2022.5.2>
- [36] Stach Kuijpers, John van de Wetering, and Aleks Kissinger. Graphical Fourier theory and the cost of quantum addition. 2019. <https://doi.org/10.48550/arXiv.1904.07551>. 64
- [37] Louis Lemonnier, John van de Wetering, and Aleks Kissinger. Hypergraph simplification: Linking the path-sum approach to the ZH-calculus. 2020. <https://doi.org/10.48550/arXiv.2003.13564>. 4
- [38] Hector Miller-Bakewell. Entanglement and Quaternions: The graphical calculus ZQ. 2020. <https://doi.org/10.48550/arXiv.2003.09999>. 11
- [39] Hector Miller-Bakewell. *Graphical Calculi and Their Conjecture Synthesis*. DPhil thesis, University of Oxford, 2020. <https://doi.org/10.48550/arXiv.2010.03914>. 4, 11
- [40] Kang Feng Ng and Quanlong Wang. A universal completion of the ZX-calculus. 2017. <https://doi.org/10.48550/arXiv.1706.09877>. 3, 4, 16
- [41] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010. <https://doi.org/10.1119/1.1463744>. 2, 3
- [42] Peter Selinger. Dagger compact closed categories and completely positive maps. *Electronic Notes in Theoretical computer science*, 170:139–163, 2007. <https://doi.org/10.1016/j.entcs.2006.12.018>. 8
- [43] Yaoyun Shi. Both Toffoli and controlled-not need little help to do universal quantum computing. *Quantum Information & Computation*, 3(1):84–92, 2003. 4
- [44] John van de Wetering. ZX-calculus for the working quantum computer scientist. 2020. <https://doi.org/10.48550/arXiv.2012.13966>. 2
- [45] John van de Wetering and Sal Wolffs. Completeness of the phase-free ZH-calculus. 2019. <https://doi.org/10.48550/arXiv.1904.07545>. 4
- [46] Renaud Vilmart. A near-minimal axiomatisation of ZX-calculus for pure qubit quantum mechanics. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10, 2019. <https://doi.org/10.1109/LICS.2019.8785765>. 11
- [47] Renaud Vilmart. A ZX-calculus with triangles for Toffoli-Hadamard, Clifford+T, and beyond. In Peter Selinger and Giulio Chiribella, editors, *Proceedings of the 15th International Conference on Quantum Physics and Logic, Halifax, Canada, 3-7th June 2018*, volume 287 of *Electronic Proceedings in Theoretical Computer Science*, pages 313–344. Open Publishing Association, 2019. <https://doi.org/10.4204/EPTCS.287.18>. 4, 11, 16
- [48] Renaud Vilmart, 2020. Personal communication. 62
- [49] Renaud Vilmart. The structure of sum-over-paths, its consequences, and completeness for Clifford. In Stefan Kiefer and Christine Tasson, editors, *Foundations of Software Science and Computation Structures*, pages 531–550, Cham, 2021. Springer International Publishing. ISBN 978-3-030-71995-1. [https://doi.org/10.1007/978-3-030-71995-1\\_27](https://doi.org/10.1007/978-3-030-71995-1_27). 4
- [50] Renaud Vilmart. Quantum multiple-valued decision diagrams in graphical calculi. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*, volume 202 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 89:1–89:15, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-201-3. <https://doi.org/10.4230/LIPIcs.MFCS.2021.89>. 4
- [51] Renaud Vilmart. Completeness of sum-over-paths for Toffoli-Hadamard and the Clifford Hierarchy. *HAL preprint: hal-03654438*, 2022. <https://hal.archives-ouvertes.fr/hal-03654438>. 4
- [52] Quanlong Wang. Algebraic complete axiomatisation of ZX-calculus with a normal form via elementary matrix operations. 2020. <https://doi.org/10.48550/arXiv.2007.13739>. 4

- [53] Fabio Zanasi. *Interacting Hopf Algebras—The Theory of Linear Systems*. PhD thesis, Lyon, École normale supérieure, 2015. <https://doi.org/10.48550/arXiv.1805.03032>. 5