Survey on the state of the ZX-calculus

H. Miller-Bakewell

Dept. of Comput. Sci., Univ. of Oxford, UK
Hector.Miller-Bakewell@cs.ox.ac.uk

March 7, 2019

Contents

1 Introduction

This document is currently being updated to show the advances made since late 2017.

1.1 This Document

This document is designed to be a comprehensive survey of the ZX-calculus. It should contain all major results, along with descriptions of the methods used (where useful) and references to where one can find full proofs. It should also contain all the information needed to understand and use the calculus, the only necessary prior knowledge being basic linear algebra.

1.2 A brief history of the ZX-calculus

Diagrammatic languages are comprised of three parts:

- Diagrams,
- Diagram-manipulation rules, and
- an Interpretation for the diagrams.

ZX-calculus is broadly defined as the diagrammatic language composed of green (Z) and red (X) spiders, a simple set of rules governing how such diagrams can be manipulated, and a quantum mechanical interpretation of the diagrams. We could also say that ZX-calculus is the abstraction of only the necessary information from certain fundamental quantum interactions. Our reason for giving such a broad definition first is that the ZX-calculus as a concept has evolved over time. As research progressed we showed that more rules were needed and so added them in, or that rules were now redundant and took them away.

Thanks to recent results we can now say "this set of rules completes this segment of the language" (more on that in section §5,) but even now there is still work being done on finding *better* collections of rules, where better can mean fewer in number, simpler, or just better suited for your purpose. This section will show how the calculus has evolved since its inception, and the concepts mentioned are left intentionally vague until we give concrete and current definitions in section §2.

The idea that would become the ZX-calculus first appeared in the paper "Interacting Quantum Observables" [CD08] by Bob Coecke and Ross Duncan. The fundamental observation appears as their Theorem 1: "Any morphism built from a classical structure and the underlying structure of the category depends only on the arities involved." We will make this a precise statement below, but for now take it as the birth of the concept of "spiders"; morphisms that can take any number of inputs and outputs. The graphical language of that first paper is used more as an aid to intuition than as a logical tool in its own right: It is in Ross Duncan and Simon Perdrix's chapter of "Mathematical Theory and Computational Practice" [DP09] that the calculus appears as an object of individual study.

As use of the calculus grew its presentation became more refined. Short-hand symbols such as the diamond and the star were introduced, and the phase-group notation

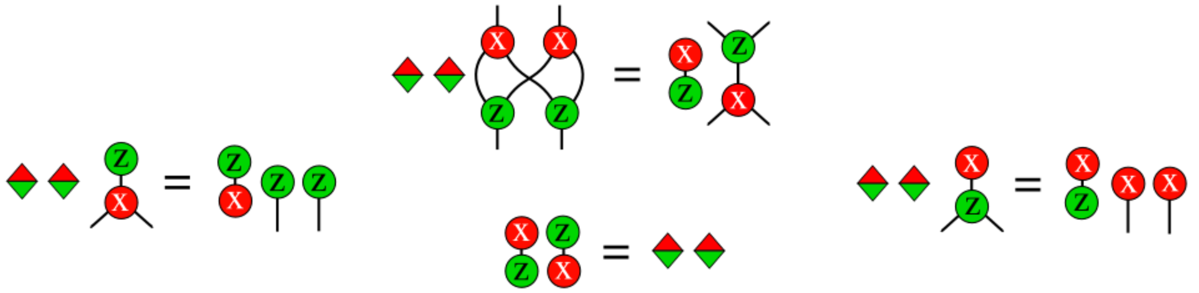


Figure 1: (A prototype of the) ZX-calculus as it first appeared in [CD08]

changed to that of additive groups. Papers separated the diagrams, their rules and their interpretations into distinct entities.

Initially the focus was on stabilizer quantum mechanics; for our purposes that means the phase group was restricted to multiples of $\pi/2$. The calculus was already being used to explore phenomena such as quantum teleportation, but proof of logical grounding is as important as the expressive power. It was in [Bac13] that the calculus was shown to be complete for stabilizer quantum mechanics, securing the logical grounding needed. From here the task moved on to the Clifford+T phase group (multiples of $\pi/4$), a segment of quantum mechanics expressive enough to approximate universal quantum computation, as well as other areas such as the use of qutrits rather than qubits, diagrams constrained to a fixed number of rails and so on. Most recently there have been results on the universal segment (the phase group can take any value in $[0, 2\pi)$), thanks to work done on the related ZW calculus.

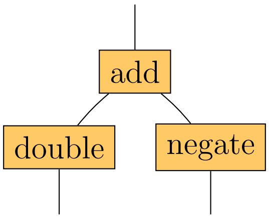
The visual depiction of the quantum teleportation protocol is the most often used example of ZX-calculus, but it is by no means the only concept that can be explored this way. The Oxford computer science master's courses that relate to quantum computing are now entirely taught using graphical formalisms, the contents of which are contained inside the book "Picturing Quantum Processes" [CK17]. The aim remains to express only those things that are necessary to the calculation, while being certain that any information ignored is justifiably ignored. Work has therefore sprung up that tries to express the rules of ZX-calculus in their simplest possible form, the results on completeness having now been established.

Since we are now up to date with the history of the ZX-calculus we will present it in its current form, in the context of the more general notion of process diagrams.

1.3 Process Diagrams

To paraphrase [CK17, Chapter 3]: A **process** is anything with zero or more inputs, and zero or more outputs. **Wires** join these inputs and outputs together, with wires often referred to as having a **type**, i.e. a description of what is being carried along that wire. The result of joining processes together by such wires is still a process; any open-ended wires are viewed as inputs and outputs of your entire system.

We shall use the convention that diagrams are to be read **bottom-to-top** in this paper, as that is the convention used in many of the papers we cite. Inputs will therefore always be drawn at the bottom of a process, and outputs at the top. As a simple example we present the following, where all wires have the real numbers, $\mathbb{R}$, as their type:



Which describes a process that takes two inputs (at the bottom) and produces an output (at the top.) The function from $\mathbb{R}^2$ to $\mathbb{R}$ that this describes is usually written as $P(a,b) = 2a - b$, but the usefulness of process diagrams becomes apparent when describing far more complicated structures.

Definition. [CK17, Definition 3.1] A **process theory** consists of:

1. A collection T of *system-types* represented by wires
2. A collection P of *processes* represented by boxes, where each input and output of a process is assigned a type from T
3. A means of *wiring processes together*. That is, an operation that interprets a diagram of processes in P as a process in P.

Example. Some example process theories:

- Functions, with sets as types
- Relations, with sets as types
- Linear maps, with vector spaces as types

There is a fundamental rule when dealing with process diagrams:

Only connectivity matters!

Which we see in figure 2, where diagrams that represent the same processes are drawn in different ways. There are some subtleties here: In some process theories you may not be allowed to bend wires in such a way that you would create cups and caps, which would then allow you to take the output of one process and plug it into the output of another box, (c.f. section §2.2.1.) In some process theories you may only be allowed to cross certain wires in certain ways. Thankfully we are subject to no such constraints in the ZX-calculus.

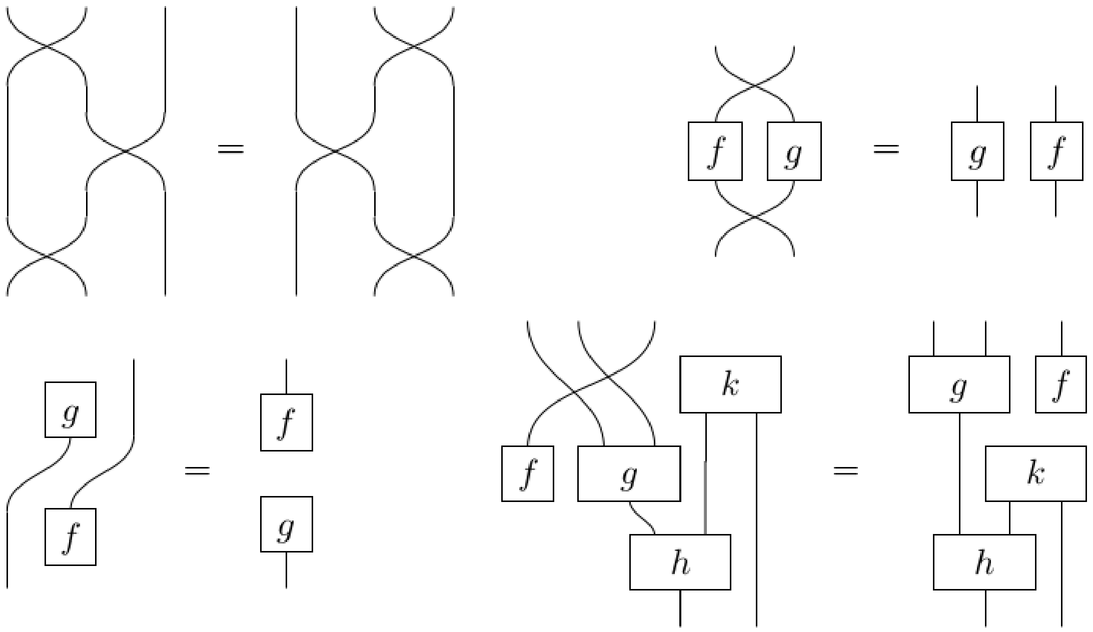


Figure 2: Examples of pairs of diagrams that represent the same processes, from [CK17]

1.4 Qubits

A lone qubit is represented as a vector in $\mathbb{C}^2$. Operations that act on a single qubit and result in a single qubit are linear maps $\mathbb{C}^2 \to \mathbb{C}^2$. We may choose bases for $\mathbb{C}^2$, traditionally we use the "computational basis"; an orthonormal basis with basis elements represented as $|\,0\,\rangle$ and $|\,1\,\rangle$.

TODO: Work out just how low level to make this

1.5 Fragments, and presentation of results

The versatility of the ZX calculus means that we can apply it to various situations, but this does, of course, mean that we need to be precise about which situation we are in. These situations are better known as **fragments**, a term evoking the idea that some situations form smaller versions of a more complicated whole. We will assume that we are dealing with qubits unless otherwise specified.

The first property of a fragment is its expressive power. All of our processes will be of the form $(\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes m}$, but, rather like $SL_n(\mathbb{C})$ embedding in $GL_n(\mathbb{C})$, we may restrict ourselves to certain subsets of those processes. We deal with the effects on the ZX-calculus in this section, and the relation between these terms and quantum computing in section §1.7. The universal ZX has a phase group of $[0, 2\pi)$ with addition modulo 2π. Fragments of the calculus concern themselves with different subgroups of this universal phase group. The only other thing taken into consideration is the notion of equivalence being used; for some purposes two states are considered equivalent if one is a scaled version of the other. Here are the most common fragments considered:

- Phase group, taken additively modulo 2π:

Stabilizer	multiples of $\pi/2$
Clifford+T	multiples of $\pi/4$
Universal	elements of $[0, 2\pi)$

And their expressive power:

Stabilizer	Efficiently simulatable on a classical computer (§1.7)
Clifford+T	Can arbitrarily approximate any complex linear map
Universal	Can specify any complex linear map

- Notions of equivalence for states x:

Equality	$x \sim x$	
Equivalence up to global phase	$x \sim e^{i\phi}x$	$\phi \in \mathbb{R}$
Equivalence up to a non-zero complex scalar	$x \sim \lambda x$	$\lambda \in \mathbb{C}^{\times}$

These are referred to using the following names:

"With scalars"	$x \sim x$	
"Up to global phase"	$x \sim e^{i\phi}x$	$\phi \in \mathbb{R}$
"Without scalars"	$x \sim \lambda x$	$\lambda \in \mathbb{C}^{\times}$

Note that the symbol used refers to what is *quotiented out* and not to what is still present.

For each combination of the above there are certain properties we are interested in. Foremost of these are:

- Completeness §5
- Universality §6
- Normal Forms §7

1.6 Quantum Circuits

The ZX calculus is at its most useful when we allow any number of qubit channels simultaneously. This is, sadly, not in keeping with the current paradigm of quantum computing. Quantum computers act in the following manner:

1. At all times the number of qubits available is constant
2. Any *gate* (action on the system by the computer) is unitary

Some work has been done on ZX circuits directly, see §1.6, and it shall be made clear when we are restricting ourselves to this case. In the rest of this survey we focus on "general" ZX diagrams as distinct from ZX circuits. Such a circuit is often rendered in ZX as a series of parallel horizontal lines (the qubit register) with wires inside gates drawn vertically. These diagrams are to be read left-to-right in keeping with standard circuit diagram notation. Any ZX circuit is automatically a ZX diagram, but which diagrams can be deformed into circuits is a focus in current research.

1.7 Pauli, Stabilizer, Clifford, and T

For a full treatment of these terms see [Bac16] and [HV12], but we provide a quick summary here. The notational abundance comes from the way that quantum computing lies on the intersection between several fields, each with their own history and conventions.

Definition. The *Pauli operators* are:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \text{and} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

and the *Pauli Group* P_1 is the closure of $\{\pm I, \pm iI, X, Y, Z\}$ under composition. P_n is formed by taking n-fold tensor products of P_1.

P is used as shorthand for P_1.

The Pauli operators appear frequently in quantum mechanics because they form an additive basis for unitary operations on a single qubit, but note that the ZX-calculus focuses on composition, not addition, of operations.

Definition. The *Clifford Group on n qubits*, denoted here as $\mathcal{C}(P_n)$ but appearing in other papers as $\mathcal{C}_n$, is the group of operators that normalise the Pauli Group P_n.

$$\mathcal{C}(P_n) := \left\{ U \,\middle|\, \forall g \in P_n : UgU^\dagger \in P_n \right\}.$$

The Clifford Group $\mathcal{C} := \mathcal{C}(P)$ has a concise normal form in ZX, which is shown in section 7.3. While P_n is formed of tensor products of elements of P_1, the same is not true of $\mathcal{C}(P_n)$ as products of elements of $\mathcal{C}(P_1)$. As such we distinguish between $\mathcal{C}(P_n)$ and $\mathcal{C}(P_1)^{\otimes n} = \mathcal{C}^{\otimes n}$.

Definition. The *local Clifford group on n qubits*, denoted $\mathcal{C}^{\otimes n}$ is the group of n-fold tensor products of elements of $\mathcal{C}$.

Note that this is different again to Gottesman and Chuang's Clifford Hierarchy, which also appears in papers as $\mathcal{C}_n$. This is defined as:

Definition.

$$\mathcal{C}_{k+1} := \left\{ U \,\middle|\, \forall g \in P_1 : UgU^\dagger \in \mathcal{C}_k \right\}.$$

As such there are two extant uses of the notation $\mathcal{C}_n$. When referring to the Clifford hierarchy it is the n-th level of normalisers of the Pauli group. Otherwise it is the group of normalisers of the n-th Pauli group. We trust that readers will make it clear in their own papers which version they refer to. It is thankfully the case that *The Clifford Group*, $\mathcal{C}$, consistently refers to the normaliser of P_1. Sadly P_1 is sometimes referred to as $\mathcal{G}$.

Definition. *Stabilizer quantum mechanics* is pure state qubit quantum mechanics with the following restrictions:

- The inputs to your system are all $|\,0\,\rangle$, and all measurements are in the computational basis.

- Unitary operations are required to be in the Clifford group.

We refer the reader to [Bac16] for a detailed look at stabilizer ZX-calculus, noting that while it is not precisely the same as "limit the phase group to multiples of $\pi/2$", it is close enough for our purposes. The reason for dividing our focus between the stabilizer segment and the other segments is best captured by the following result:

Result. [Got98, Theorem 1, "Gottesman-Knill"] Any quantum computer performing only:

1. Clifford group gates,
2. measurements of Pauli group operators, and
3. Clifford group operations conditioned on classical bits, which may be the results of earlier measurements,

can be perfectly simulated in polynomial time on a probabilistic classical computer.

The paper [Got98] contains excellent worked examples. In order to escape the realm of what is efficiently simulatable on classical computers we therefore need to go beyond the Clifford group gates.

Definition. The qubit *T gate* is a $\pi/4$ Z-phase rotation of the Bloch sphere (which is a $\pi/4$ green spider). Unhelpfully it is also known as the $\pi/8$ gate, since it can also be written as:

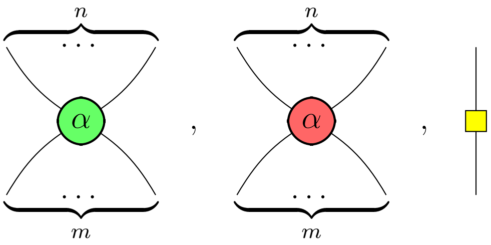
$$e^{i\pi/8}\begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}$$

It can also be used to refer to $\pi/4$ rotations in the X and Y directions, but we stick to the Z rotation version in this paper. Its primary interest is that once we have $\pi/4$ rotations (along with Hadamard and CNOT gates) we can start to approximate universal quantum computation. It is this property that has resulted (for qudit quantum computing) in the term *T gate* meaning any gate that extends your current gate set from qudit stabilizer quantum mechanics to approximate universal qudit quantum mechanics. For an exploration of this idea see [HV12].

2 The ZX-calculus

2.1 Diagrams

A qubit ZX-calculus diagram is a process diagram generated by the following processes:



Where m and n are any natural numbers (in particular, one or both could be zero), and the α belong to your *phase group*. The green spider is the Z spider, the red spider is the X spider, and the yellow box is the *Hadamard* box.

All wires connecting processes represent either the 2-dimensional Hilbert Space $\mathbb{H}$ (which is $\mathbb{C}^2$ with an inner product) or its dual space $\mathbb{H}^*$ (which are canonically isomorphic.) These wires can be bent (to create *cups* and *caps*) and can cross over each other (creating *swaps*.) This is all explained in detail in section 2.2.1. The choice of phase group defines the expressive power available in your interpretation. Choices of phase group (taken additively modulo 2π) discussed in this paper are:

Stabilizer	multiples of $\pi/2$
Clifford+T	multiples of $\pi/4$
Universal	any real phase $[0, 2\pi)$, but often written as $(-\pi, \pi]$

The equation below shows that the Hadamard box is not a necessary inclusion in the calculus, but since it is used so frequently in quantum computing it merits its own notational symbol. It has the disadvantage of being of fixed arity; always one wire in and one wire out, and so recent papers have taken to representing a wire with a Hadamard box on as a different colour of wire, rather than drawing the box.

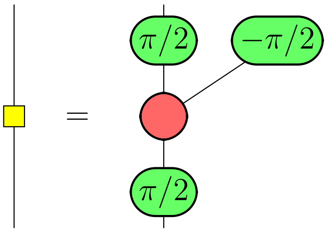


Figure 3: The Hadamard map

When dealing with qu**d**its instead of qu**b**its our wires become the d-dimensional Hilbert space (instead of 2-dimensional), but everything still follows in much the same way. In this survey's current iteration we shall restrict ourselves to the qubit case.

Any Abelian group could be used as a phase group, Spekkens' toy model ([BD16]) being a good example, but the groups that interest us most are those suited for describing quantum computation. Historically, one assumed the existence of certain additional symbols with given properties, and then derived equations such as the one for the Hadamard above. Other symbols, such as ★ and ⧫, have been used in such a way but will not be used in this paper.

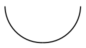
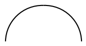
2.2 Rewrite Rules

We present here what we deem to be the standard ruleset for the ZX-calculus. There are other rulesets, and much work has been put into showing that these other rulesets are equivalent to the standard one. Section 8 is on minimality and simplicity and covers the search for other rulesets. The reasons for wanting to use different rulesets are practical in nature and usually to do with analysis on the rules themselves, or for use when working with computers on particular problems.

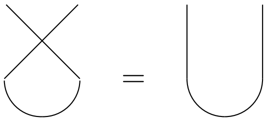
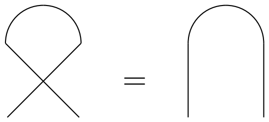
2.2.1 Wire-bending rules

The following sentence is true but does not need to be understood to use the ZX calculus: *ZX lives in a compact closed symmetric monoidal category.* This means that for any wire:

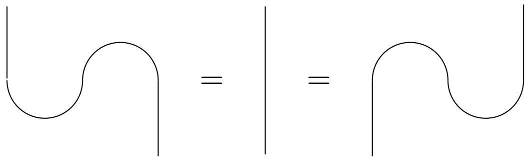
- Cups and caps exist:

 and 

- Cups and caps are commutative:

 and 

- Cups and caps obey the snake equations:



Readers that are familiar with compact closed categories may be more familiar with these equations expressed in terms of:

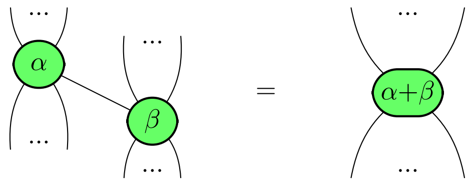
$$\epsilon_A : A \otimes A^* \to I \quad \text{and} \quad \eta_A : I \to A^* \otimes A$$

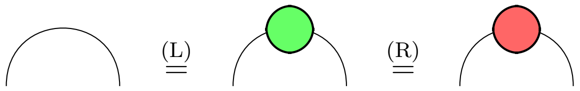
Note that here we use a cup to represent η and a cap to represent ϵ. This would normally require us to label our wires as either A or A^* accordingly, but since they are isomorphic in all situations considered in this paper we leave the labels off the wires and leave the isomorphism implicit.

2.2.2 Stabilizer Rewrite rules

All of the following rules rules hold when flipped upside-down, or with the colours red and green swapped. We present them in section 4.2 in their latest incarnation, which is as they appear in [Jea+17]. The name for each rule is given in brackets next to the rule. Note that these names reflect the history of the calculus, for example we have (S3′) but no longer an (S2). The Stabilizer rules and their names are explained in further detail below, and the right-hand side of (IV′) denotes the empty diagram.

- **Spider Laws** (S1) and (S3′):



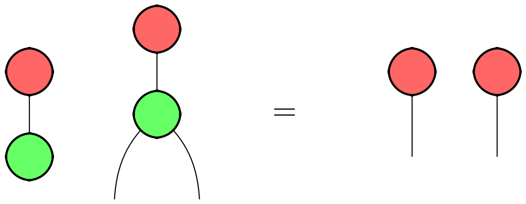


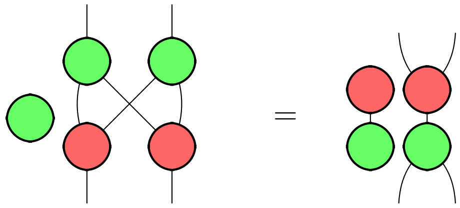
  The spider laws are the fundamental reason for choosing this notation. A connected diagram formed from spiders of only one colour is dependent on:

  - The number of inputs
  - The number of outputs
  - The sum of the phases

  A notion encapsulated by the spider laws.

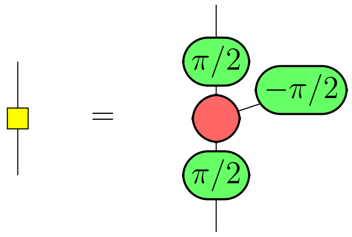
- **Bialgebra Laws** (B1) and (B2)

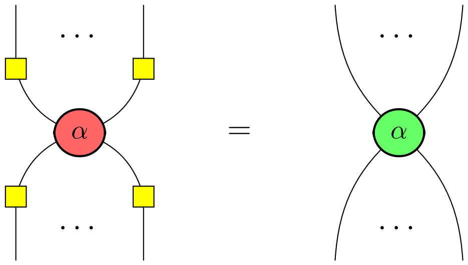




  Commonly (B2) is referred to as *the* bialgebra law, and (B1) as the *copy law*. Together with their colour-swapped counterparts they show that our spiders form a bialgebra.

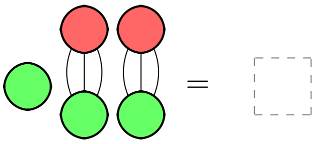
- **Euler Decomposition** (EU’) and **Hadamard Action** (H)





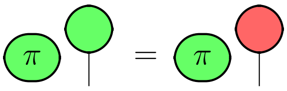
Whether the Euler Decomposition is a law or a definition depends on whether you include the yellow box as a formal part of the language, or as a shorthand for the collection of gates shown in (EU). Either way the action of the Hadamard map is to swap the colour of a spider.

- **Inverse Law** (IV')



This rule captures the multiplication of the three scalar diagrams: $2 \times \frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$, totalling the scalar 1, which is represented as the empty diagram.

- **The Zero Rule** (ZO′)



The legless *pi* spider represents the scalar 0, and as such any two diagrams of the same arity containing this spider must be equal. Using (ZO′), (S3′) and (B1) one can break apart any wire in your diagram and so show that is equal to a totally disconnected one.

2.3 Interpretation

Definition. An **interpretation** $[\![-]\!]$ of a diagrammatic calculus $\mathcal{D}$ is a monoidal functor that assigns to each diagram of $\mathcal{D}$ and element of $\mathcal{C}$.

$$[\![-]\!] : \mathcal{D} \to \mathcal{C}$$

In this survey we restrict ourselves to the case where $\mathcal{C}$ is the category $\mathrm{Mat}_{\mathbb{C}}$ of complex-valued matrices. We will explain all the relevant properties of such a monoidal functor below.

Interpretations of diagrams are generally denoted by $[\![-]\!]$, with subscripts used to identify different interpretations. While there are occasions to use multiple interpretations of the same diagram (most notably in the old proofs of incompleteness,) there is a standard interpretation of a ZX-calculus diagram and we show it in section 4.1.

To make certain that there is no ambiguity or unfamiliarity with the notation: We are working in the 2-dimensional Hilbert Space $\mathbb{H}$, which is the same as $\mathbb{C}^2$ with the Euclidean inner product. (Recall that we are assuming we are working in the qubit case.) A "ket"

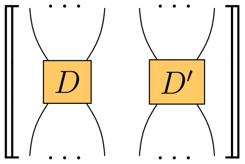
$|\,x\,\rangle$ denotes a vector in $\mathbb{C}^2$ labelled by x, with the "bra" $\langle\,x\,|$ denoting the conjugate transpose of $|\,x\,\rangle$ (which inhabits the dual space). The matrix application of $|\,x\,\rangle$ to $\langle\,y\,|$ is written $\langle\,y\,|\,x\,\rangle$ and results in the value $y.x$. We will use the basis notation inherited from quantum computation:

$$\begin{aligned}
|\,0\,\rangle &:= \begin{pmatrix}1\\0\end{pmatrix} & \\
|\,1\,\rangle &:= \begin{pmatrix}0\\1\end{pmatrix} & \\
|\,+\,\rangle &:= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} & &= \frac{1}{\sqrt{2}}(|\,0\,\rangle + |\,1\,\rangle) \\
|\,-\,\rangle &:= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} & &= \frac{1}{\sqrt{2}}(|\,0\,\rangle - |\,1\,\rangle)
\end{aligned}$$

Figure 4: Bra-ket notation.

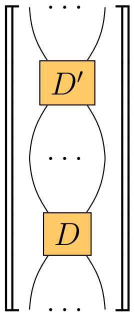
Joining diagrams together via tensor product or composition follows from $[\![\,-\,]\!]$ being a monoidal functor:

- Tensor product:



$$= [\![\,D\,]\!] \otimes [\![\,D'\,]\!]$$

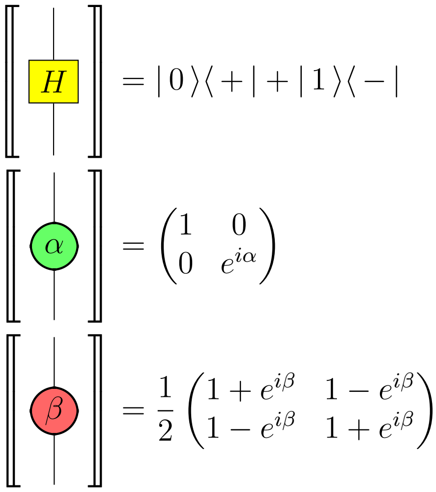
- Composition:



$$= [\![\,D'\,]\!] \circ [\![\,D\,]\!]$$

Note that for composition D necessarily has as many outputs as D' has inputs, however tensoring in copies of identity wires allows us to add inputs and outputs as required.

This forces the interpretation of the Hadamard as shown in figure 5, in which we also include the Z- and X- phase twists as examples.



$$= |\,0\,\rangle\langle\,+\,| + |\,1\,\rangle\langle\,-\,|$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$$

$$= \frac{1}{2}\begin{pmatrix} 1+e^{i\beta} & 1-e^{i\beta} \\ 1-e^{i\beta} & 1+e^{i\beta} \end{pmatrix}$$

Figure 5: The Hadamard gate, and the Z- and X- phase twists in the standard interpretation.

3 The ZW calculus

3.1 Background

The ZW calculus was primarily developed by Amar Hadzihasanovic, and the primary source we use in this paper is the thesis [Had17]. In the words of that thesis: “The ZW calculus is a complete diagrammatic axiomatisation of the theory of qubits.” This does, however, feel like an understatement. The ZW calculus can be seen as a parameterised family of calculi and for every commutative unital ring R there is the language ZW_R. Elements of this ring are referred to as r or s in the axioms, and as *labels* in text. Theorem 5.27 of [Had17] reads as follows:

> (Completeness of the ZW calculus) The map $zw_R : ZW_R \to R\textbf{bit}$ is an isomorphism of PROPs.

By substituting R for $\mathbb{C}$ we recover the statement $ZW_\mathbb{C} \simeq \textbf{Qubit}$ (*Qubit* being the usual term for $\mathbb{C} - bit$.) i.e. that $ZW_\mathbb{C}$ is complete as a diagrammatic language for qubit calculus. Not only did this give us a universal, complete diagrammatic language for quantum computation, but it provides us with a language for many more exotic models as well.

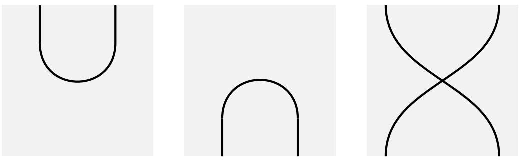
There is a minor point of disambiguation needed: “Vanilla ZW calculus” is the same as “the undecorated ZW calculus” which is *equivalent* to $ZW_\mathbb{Z}$. Undecorated ZW refers to the language where no labels are used, so there is only a single w spider of any given arity, rather than a spider for every element of R and every arity. The last two axioms of section 4.6 are not used in the undecorated calculus, but they do indicate how the equivalence between vanilla ZW and $ZW_\mathbb{Z}$ is found.

Often the ring R in question can be inferred from context, but we encourage authors to specify R at least once, and to use the phrase *vanilla* or *undecorated* ZW when appropriate. Plain “ZW” can then be used to refer to either the family of languages, or the particular ZW_R in context. In the rest of this text we will asume we are working with $ZW_\mathbb{C}$. Likewise the grey background used for ZW diagrams makes the white z spider clearer to the reader, but is not necessary.

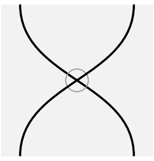
For the purposes of the ZX-calculus the completeness proofs of the Clifford+T and Universal segments both rely on the completeness of the ZW calculus. See 5 for further details.

3.2 Generators

For a quick reference of the generators and their standard interpretations, see section 4.5. The language itself comprises the usual compact categorical structure of cups, caps and swaps:

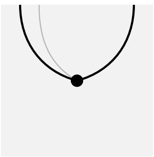


But introduces the more exotic "crossing":

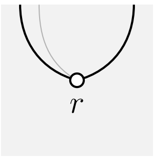


This is *almost* a braiding, but fails to satisfy some of the naturality conditions for the situations we are considering; notably the crossing does not act naturally with the z spider. The crossing does follow the Reidemeister rules (labelled rei in the rules summary,) and it obeys naturality with the w spider (nat^w_x.) It is possible to separate the space we are looking at as a $\mathbb{Z}_2$-graded module, and in that situation the crossing makes a braiding of this graded module, but for now we shall leave things as they are.

The w spider is undecorated, and subtly different to the spiders of the ZX calculus. Spider merging (the (S1) rule from ZX) no longer applies in ZW, but the related cut_w rule, commutativity and some other similarities still hold. It is represented as a black spider:



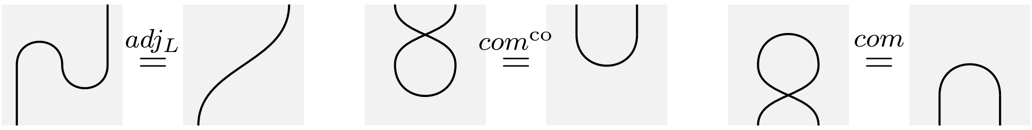
The z spider, by contrast, is decorated with elements of R. (Unless you are using the undecorated calculus, which is the same as only using the label "1")



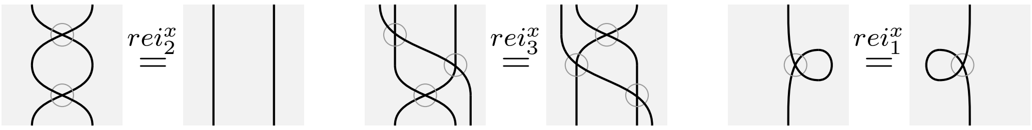
3.3 Rules

The reference list of ZW_R rules is found in section 4.6. Here is a quick run-down of the rules broken into section:

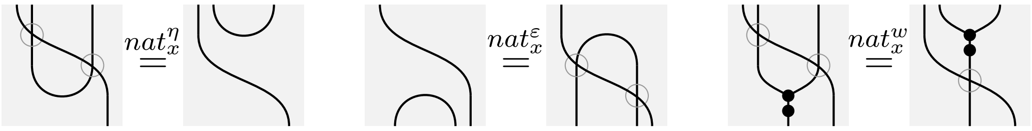
Wire-bending rules



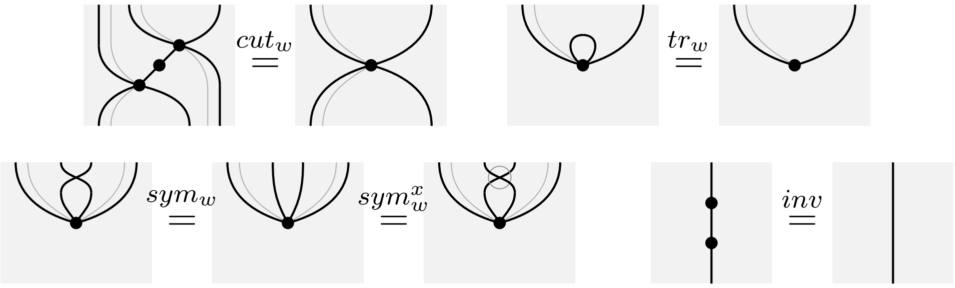
Reidemeister rules for the crossing



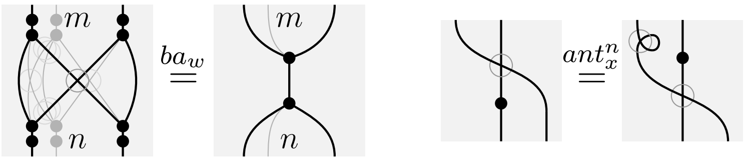
Naturality of the crossing



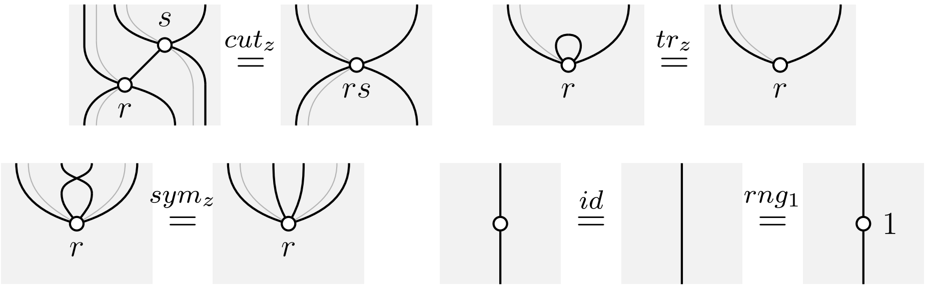
w-spider rules



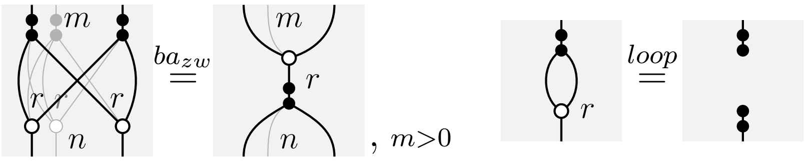
w-spider bialgebra and antipode



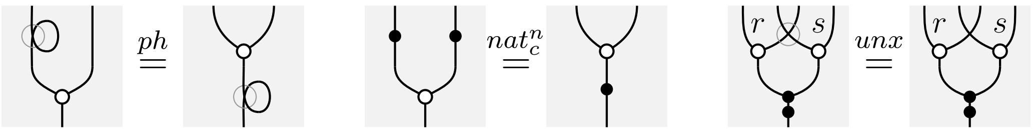
z-spider rules



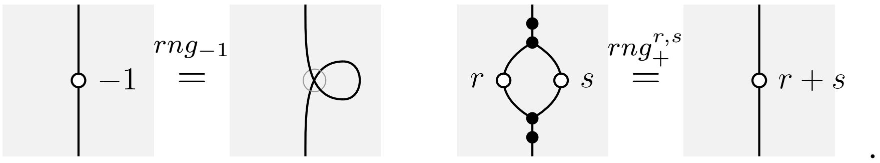
w and z bialgebra interaction and loop



z-spider naturality and uncrossing



Addidtion and additive inverse in R



4 Rules reference

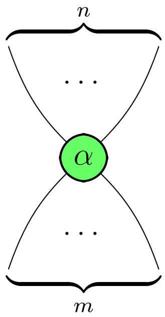
This section is to serve as a quick reference for the rules of various related graphical languages.

4.1 ZX Qubit Notation

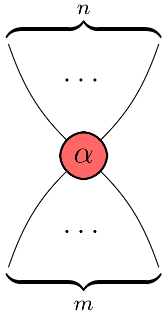
Phase groups

$\alpha \in < \pi/2 >$	Stabiliser
$\alpha \in < \pi/4 >$	Clifford+T
$\alpha \in [0, 2\pi)$	Universal

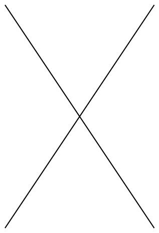
Standard interpretation



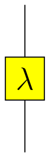
$|\,0\,\rangle^{\otimes n}\langle\,0\,|^{\otimes m} + e^{i\alpha}|\,1\,\rangle^{\otimes n}\langle\,1\,|^{\otimes m}$



$|\,+\,\rangle^{\otimes n}\langle\,+\,|^{\otimes m} + e^{i\alpha}|\,-\,\rangle^{\otimes n}\langle\,-\,|^{\otimes m}$

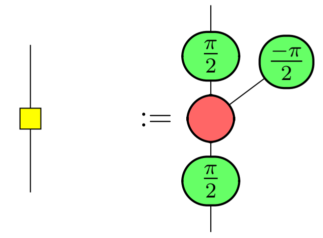


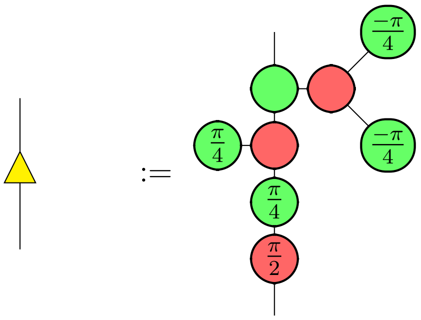
$|\,00\,\rangle\langle\,00\,| + |\,01\,\rangle\langle\,10\,| + |\,10\,\rangle\langle\,01\,| + |\,11\,\rangle\langle\,11\,|$



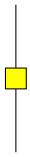
$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & \lambda \end{pmatrix}$

Notational shorthand

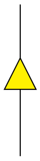




Standard interpretation of these shorthands



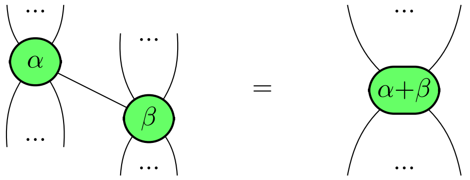
$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}$$



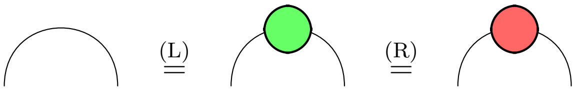
$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

Table 2: Standard interpretation and some common notational shorthands for qubit ZX-calculus. Diagrams in this papers are to be read from bottom to top.

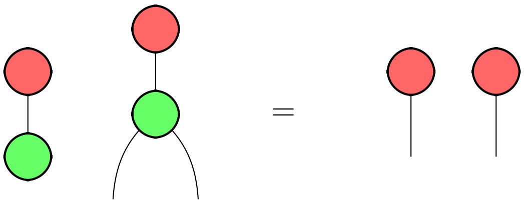
4.2 ZX Qubit Stabilizer Rules, September 2017



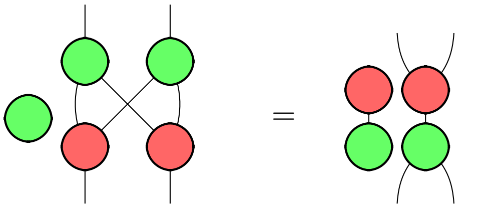
(S1)



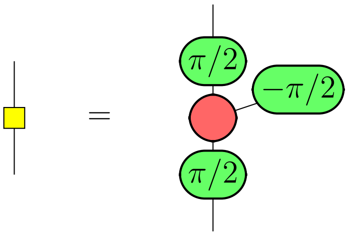
(S3′)



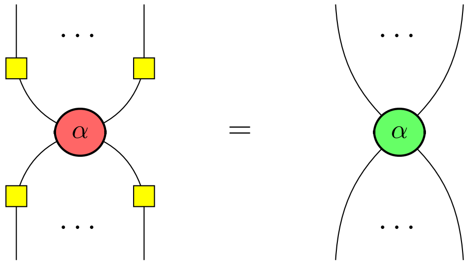
(B1)



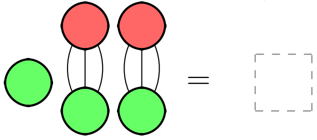
(B2′)



(EU′)

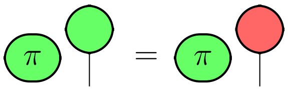


(H)



(IV′)

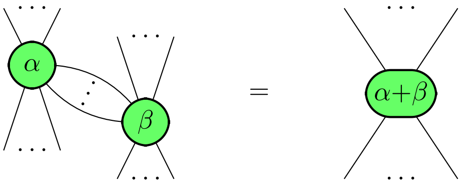
ZX Qubit Stabiliser Rules



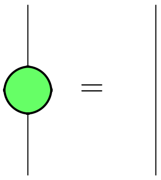
(ZO′)

Table 3: Simplified rules for the stabilizer ZX-calculus, using the conventions that the right-hand side of (IV′) is an empty diagram and that ellipses denote zero or more wires. Source: [BPW17]

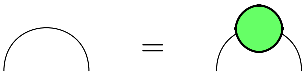
4.3 ZX Qubit Clifford+T Rules, May 2017



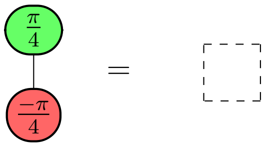
(S1)



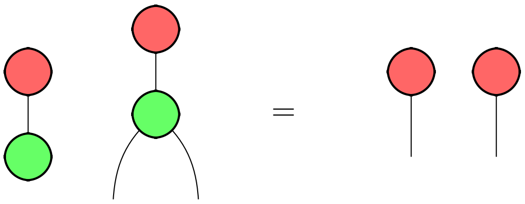
(S2)



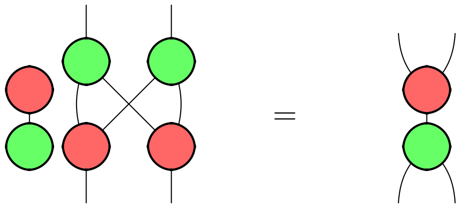
(S3)



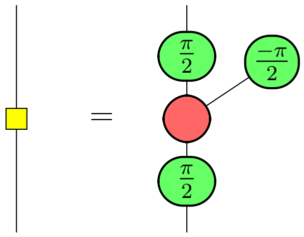
(E)



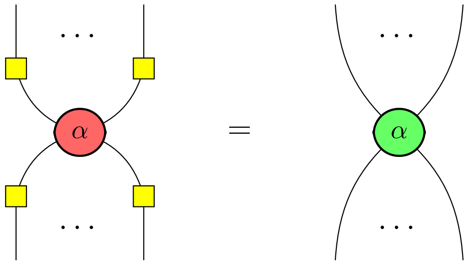
(B1)



(B2)

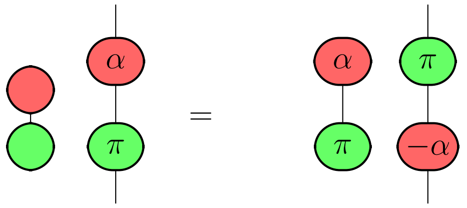


(EU)

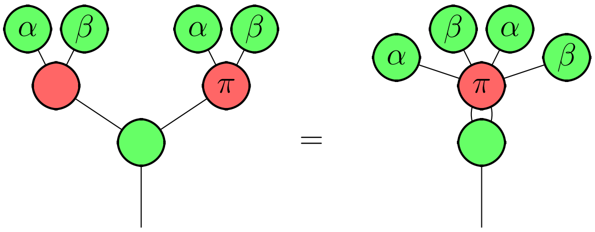


(H)

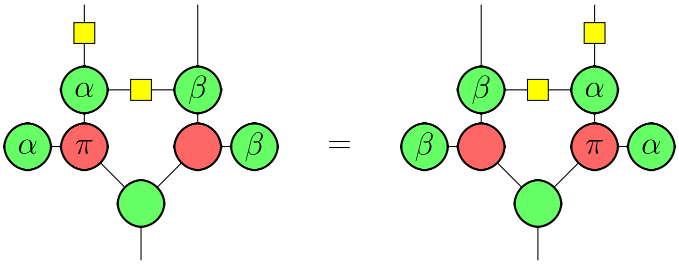
ZX Qubit Clifford+T Rules



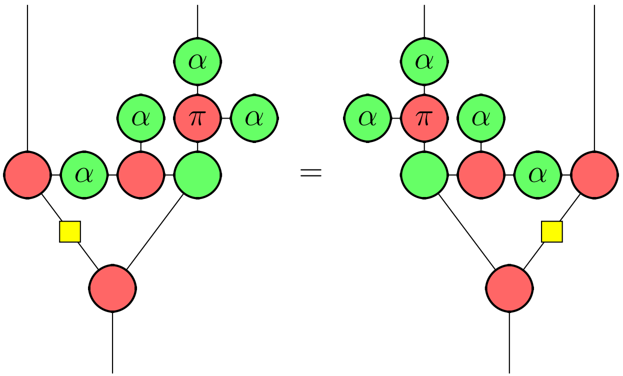
(K2)



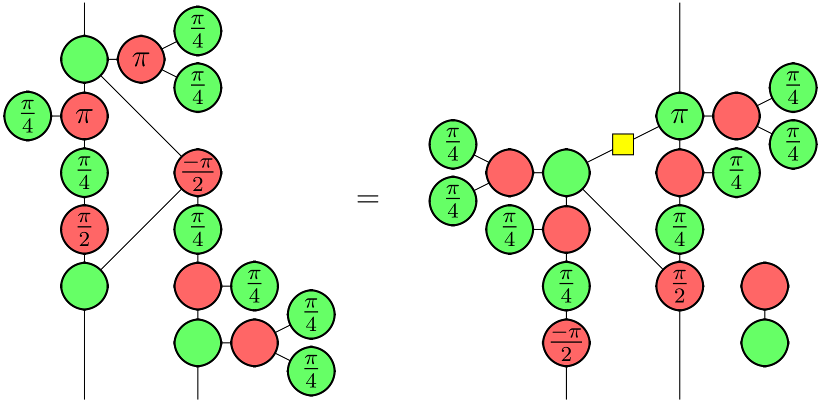
(SUP)



(C1)



(C2)

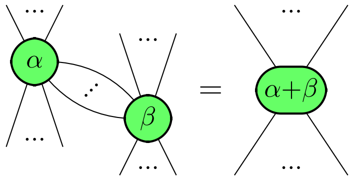


(C3)

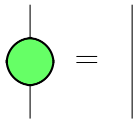
Table 4: The set of rules for Clifford+T ZX-calculus with scalars. All of these rules also hold when flipped upside-down, or with the colours red and green swapped. The right-hand side of (E) is an empty diagram. (...) denote zero or more wires, while ($\iddots$) denote one or more wires. Source: [JPV17]

4.4 ZX Qubit Universal Rules, June 2017

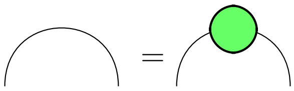
Original ZX-calculus rules, where $\alpha, \beta \in [0,\ 2\pi)$



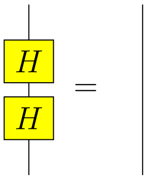
(S1)



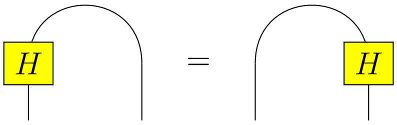
(S2)



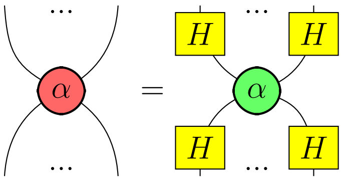
(S3)



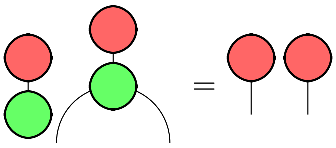
(H2)



(H3)

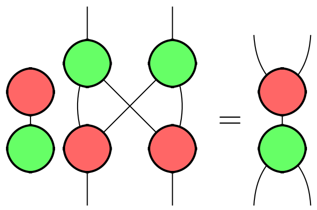


(H)

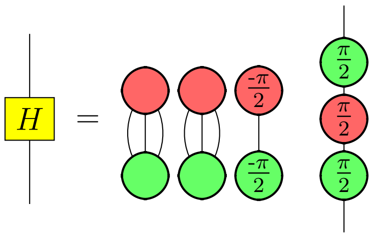


(B1)

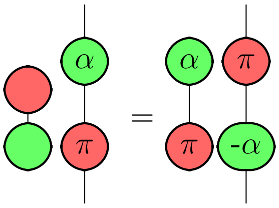
ZX Qubit Universal Rules



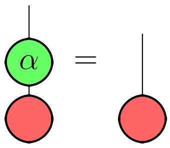
(B2)



(EU)

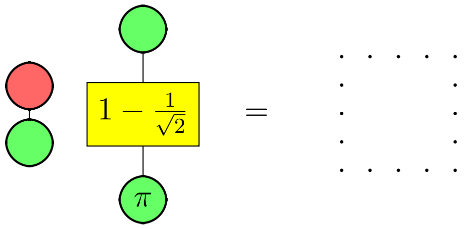


(K2)

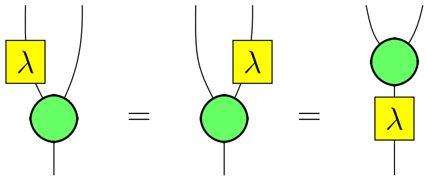


(S4)

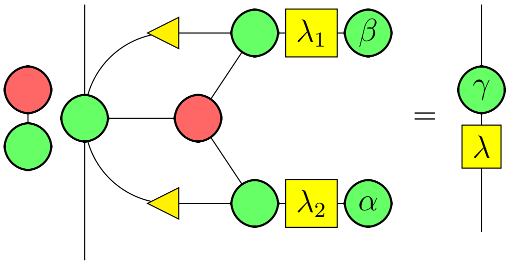
Extended ZX-calculus rules for λ and addition, where $\lambda, \lambda_1, \lambda_2 \geq 0, \alpha, \beta, \gamma \in [0,\ 2\pi)$; in (AD), $\lambda e^{i\gamma} = \lambda_1 e^{i\beta} + \lambda_2 e^{i\alpha}$.



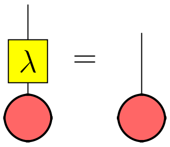
(IV)



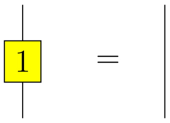
(L1)



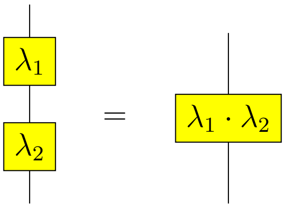
(AD)



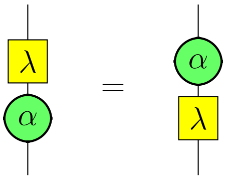
(L2)



(L3)

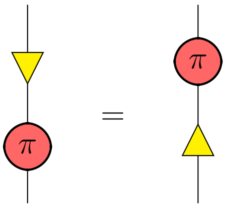


(L4)



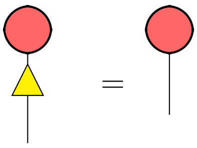
(L5)

Extended ZX-calculus rules for the triangle, where $\lambda \geq 0, \alpha \in [0,\ 2\pi)$.

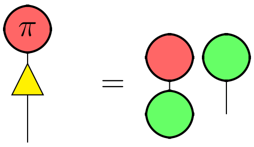


(TR1)

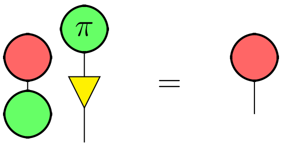
ZX Qubit Universal Rules



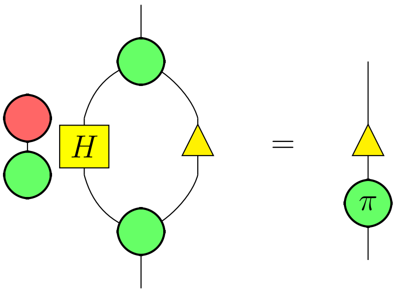
(TR2)



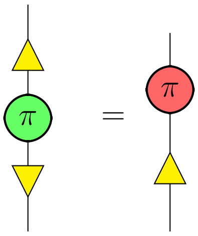
(TR3)



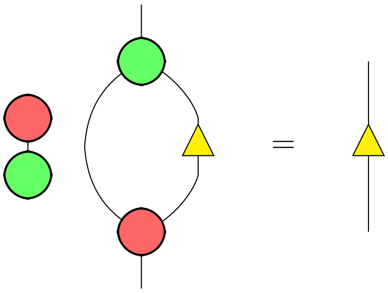
(TR4)



(TR5)

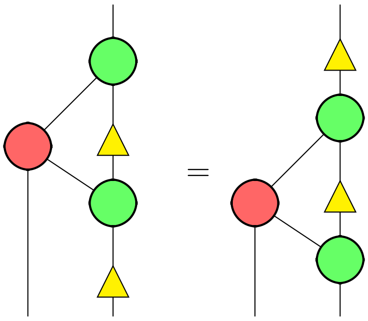


(TR6)

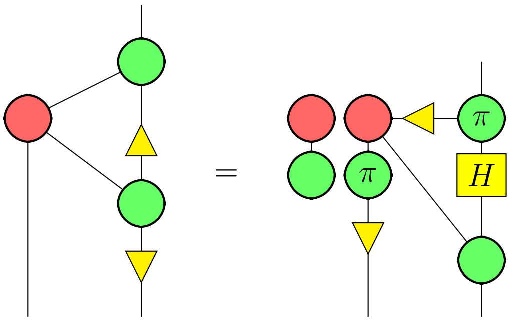


(TR7)

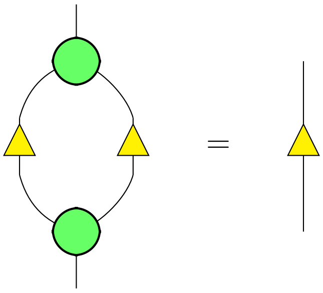
ZX Qubit Universal Rules



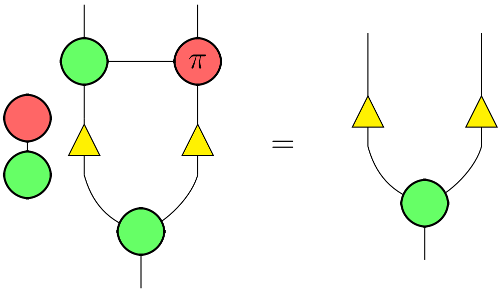
(TR8)



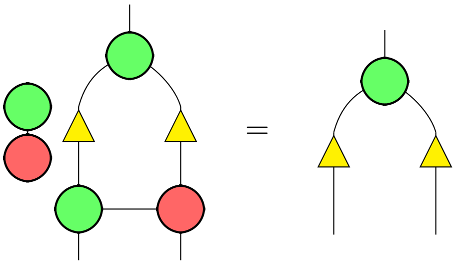
(TR9)



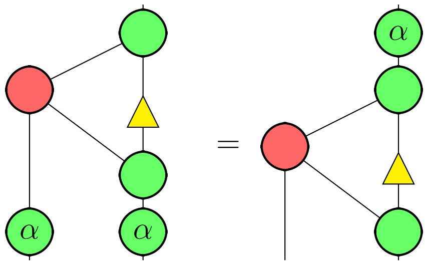
(TR10)



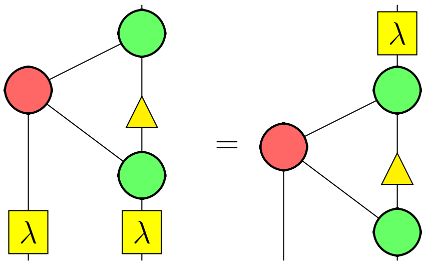
(TR11)



(TR12)



(TR13)



(TR14)

Table 5: The set of rules for Qubit Universal ZX-calculus with scalars. All of these rules also hold when flipped upside-down, or with the colours red and green swapped. The right-hand side of (IV) is an empty diagram. (...) denote zero or more wires, while (.·˙) denote one or more wires. Source: [NW17]

4.5 ZW Notation

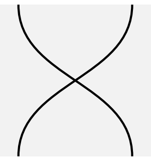
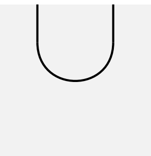
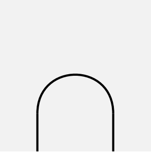
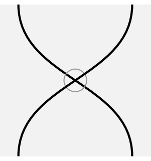
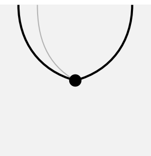
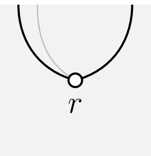
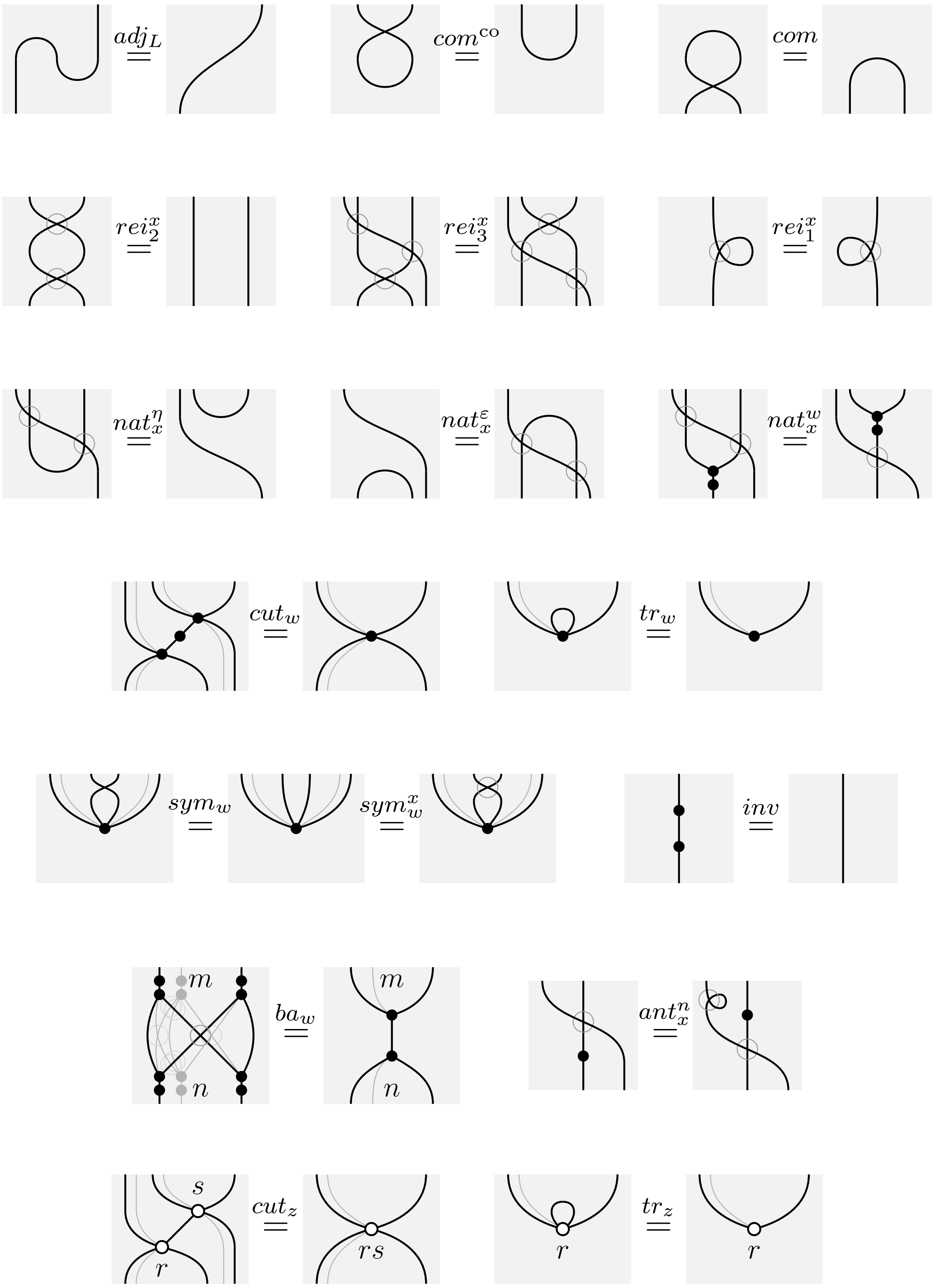
		Standard interpretation
	$s : [2] \to [2]$	$\mapsto\ \lvert 00\rangle\langle 00\rvert + \lvert 01\rangle\langle 10\rvert + \lvert 10\rangle\langle 01\rvert + \lvert 11\rangle\langle 11\rvert$
	$\eta : [0] \to [2]$	$\mapsto\ \lvert 00\rangle + \lvert 11\rangle$
	$\varepsilon : [2] \to [0]$	$\mapsto\ \langle 00\rvert + \langle 11\rvert$
	$x : [2] \to [2]$	$\mapsto\ \lvert 00\rangle\langle 00\rvert + \lvert 01\rangle\langle 10\rvert + \lvert 10\rangle\langle 01\rvert - \lvert 11\rangle\langle 11\rvert$
	$w_n : [0] \to [n]$	$\mapsto\ \sum_{k=1}^{n} \lvert \underbrace{0\dots0}_{k-1}1\underbrace{0\dots0}_{n-k}\rangle$
	$z_n^r : [0] \to [n]$	$\mapsto\ \lvert \underbrace{0\dots0}_{n}\rangle + r\lvert \underbrace{1\dots1}_{n}\rangle$

Table 6: Standard interpretation for the ZW-calculus.

4.6 ZW Calculus Rules, September 2017



ZW Rules

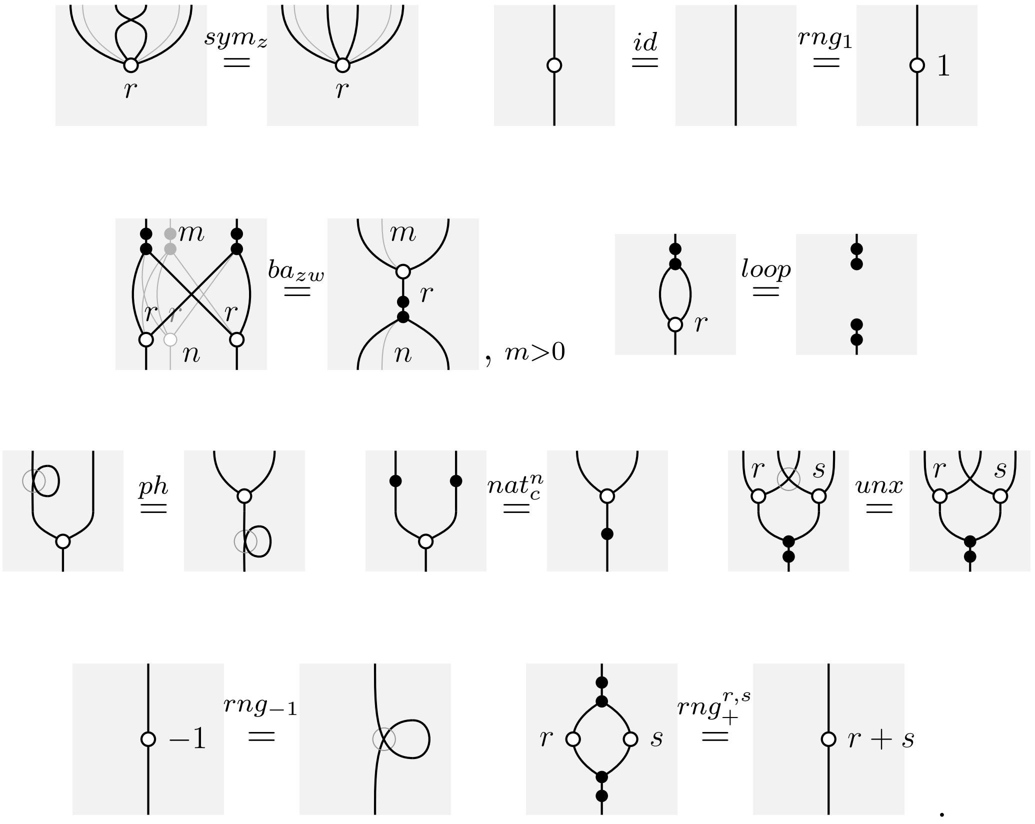


Table 7: The full list of axioms for ZW_R, where R is any commutative ring with 1. Restricting to $r, s = 1$ and removing the last two axioms gives an axiomatisation of undecorated ZW. The axioms of PROPs are implied. Source: [Had17]

5 Completeness

5.1 Definition and statement of results

TODO: The section was last up-to-date in late 2017. A lot has happened since then.

Definition. [Bac16] A graphical rewrite system is **complete** for an interpretation $[\![-]\!]$ if for any diagrams D_1 and D_2:

$$[\![D_1]\!] = [\![D_2]\!] \implies D_1 = D_2$$

Note that the first equality is equality in the interpretation, assumed to be $\mathrm{Mat}_{\mathbb{C}}$, and the second equality is equality as diagrams i.e. an equality that can be shown using your rules.

Definition. A graphical rewrite system is therefore **incomplete** for an interpretation $[\![-]\!]$ if there exist diagrams with the same interpretation that we cannot show to be equal from inside the rewrite system.

Table 8 presents the known results on completeness. The expression of scalar products as tensor products allows us to consider completeness in two passes; completeness up to complex scalar equivalence first ($x \sim \lambda x$, $\lambda \in \mathbb{C}^{\times}$), and then completeness of scalars separately. Results written in brackets are easy corollaries of published results.

	$< \pi/2 >$	$< \pi/4 >$	$[0, 2\pi)$
1	$\top$	$\top$	$\top$
$e^{i\phi}$	$(\top)$	$(\top)$	$(\top)$
λ	$\top$	$(\top)$	$(\top)$

Table 8: Completeness of ZX-calculus

Thanks to a concerted effort in the Summer of 2017 we now have completeness results for all major segments of the qubit ZX calculus.

5.2 Stabilizer Segment

Result. Stabilizer ZX-calculus is complete for exact, positive-real and complex notions of equivalence.

Proof. The proof is based on the argument in [Bac13], and then further built on in [Bac16].

The completeness result relies on the existence of something that is almost a normal form; more accurately the existence of a small collection of diagrams in each equivalence class that allow for (relatively) easy comparison of diagrams. The general principle for proving completeness goes as follows:

- Show completeness for scalar-free stabilizer diagrams via rGS-LC normal form (covered in section 7.2.)

- Show completeness for zero stabilizer diagrams via the disconnected normal form in figure 6.

- Show completeness for non-zero scalar stabilizer diagrams, via a normal form built from red effects and green states (covered in section 7.4.) Notation: Backens uses the star diagram ★ to represent the diagram with value 1/2.

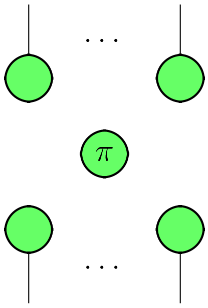


Figure 6: The disconnected normal form, interpreted as 0

With the number of inputs and outputs dependent on the arity of the process considered. □

The current ruleset is given in section 4.2

5.3 Clifford+T Segment

Result. Clifford+T ZX-calculus is complete for exact, positive-real and complex notions of equivalence.

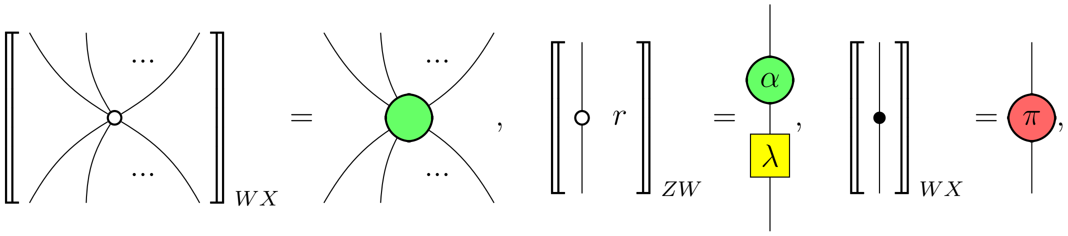
Proof. The result is due to Jeandel, Perdrix and Vilmart, published in [JPV17] and makes use of an equivalence with the ZW calculus. □

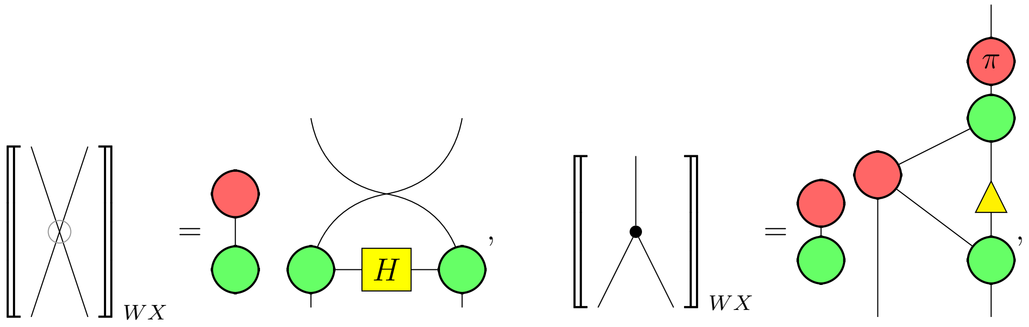
A current ruleset in given in section 4.3.

5.4 Universal Segment

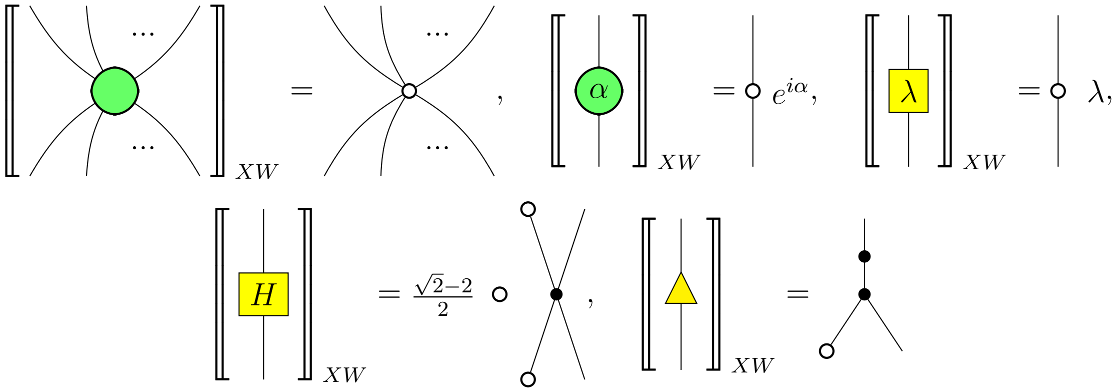
Result. Exact, universal ZX-calculus is complete

Proof. The result is due to Ng and Wang, published in [NW17] and makes use of an equivalence with the ZX calculus, albeit a slightly different equivalence to the one in section 5.3. Cups, caps and swaps are unaffected by the equivalence, leaving the following translation $[\![\cdot]\!]_{WX}$ from ZW to ZX:





Where $r = \lambda e^{i\alpha}$, $\lambda \in \mathbb{R}_{>0}$. The translation $[\![\cdot]\!]_{XW}$ from ZX to ZW is:



where $\alpha \in [0,\ 2\pi)$, $\lambda \in \mathbb{R}_{>0}$.

□

A current ruleset is given in section 4.4

6 Universality

6.1 Definition and statement of results

Definition. [Bac16] A graphical language is **universal** under the interpretation $[\![-]\!]$ if for any process P in the underlying theory there exists a diagram D such that:

$$[\![D]\!] = P.$$

The universality results continue to use the standard interpretation of the ZX-calculus. The choice of underlying theory, however, depends on the phase group; we are implicitly expanding the underlying theory we try to describe when we move from the $< \pi/2 >$ phase group to the larger phase groups.

Table 9 expresses the same information in two different ways. The first is in the form of phase groups, underlying theories and universality laws. The second describes the method of proof; linking the underlying theory to rotations of the Bloch sphere, where statements of universality have become statements of density. When we say that the phase group "generates" a group G (e.g. $SU(2)$) we mean that the Z-phases and X-phases in conjunction with the CNOT gate and their adjoints generates G.

In all of the following we are showing the results up to strict equality (i.e. "with scalars".) The results for weaker notions of equivalence follow immediately, and the results are show in table 10.

Phase Group	Underlying theory	Universality
$<\pi/2>$	Pure state qubit quant. mech.	Universal
$<\pi/4>$	Qubit quantum computation	Approx. universal
$[0, 2\pi)$	Qubit quantum computation	Universal

Phase Group	Target Bloch-Sphere Rotation group	Generates
$<\pi/2>$	The Pauli subgroup of $SU(2)$	Yes
$<\pi/4>$	$SU(2)$	Dense
$[0, 2\pi)$	$SU(2)$	Yes

Table 9: Universality results, expressed in two different forms

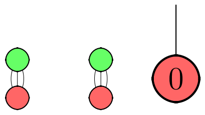
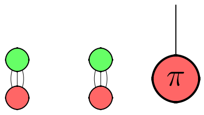
	$<\pi/2>$	$<\pi/4>$	$[0, 2\pi)$
1	$\top$	$\top$	$\top$
$e^{i\phi}$	$(\top)$	$(\top)$	$(\top)$
λ	$(\top)$	$(\top)$	$(\top)$

Table 10: Universality of ZX-calculus.

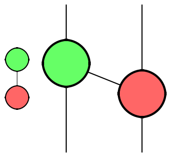
6.2 Stabilizer Segment

Result. [Bac16, Lemma 3.1.7] The ZX-calculus with the standard interpretation $[\![-]\!]$ is universal for pure state qubit quantum mechanics.

Proof. Construct the inputs:

$| 0 \rangle =$  , $| 1 \rangle =$ 

Construct the CNOT gate:

$C_X =$ 

Then use the normal form for scalars as in section 7.4, and the normal form for single-qubit Clifford operators as in section 7.3.

6.3 Clifford+T Segment

The core of the approximate universality result for Clifford+T quantum computing lies in earlier work by Boykin et al. in [Boy+99]. The results we need are as follows, the last one following easily from inspection of the standard interpretation.

Result. [NC10] CNOT and $SU(2)$ are sufficient for universal quantum computation.

Result. [Boy+99] The Clifford+T segment of quantum mechanics is dense in $SU(2)$.

Result. The ZX-calculus for Clifford+T is approximately universal for quantum computation.

6.4 Universal Segment

Result. [CD11] The ZX-calculus with the phase group $[0, 2\pi)$ is universal for quantum computation.

The argument is the same as for universality of the stabilizer segment (section 6.2); except that now we obtain single-qubit unitary operators via the Euler Decomposition (section 7.9) and obtain scalars via the normal form in section 7.10. As such the entirety of $SU(2)$ can be represented exactly, in contrast to the approximate representation via the Clifford+T phase group. This, in combination with the results in section 6.3, gives the result.

7 Normal Forms

7.1 Definition and statement of results

Definition. A **normal form** is an unique choice of representative from each equivalence class.

We shall consider both multi-qubit and single-qubit versions of the calculus. The single-qubit version of the calculus does not allow arbitrary spiders or CNOT gates, just phase gates applied sequentially to a single qubit. (The Hadamard gate can be formulated using just $\pi/2$ phase gates on a single qubit, up to a scalar, and so also appears in the single-qubit version.) The multi-qubit version is simply the version we have been using in the rest of this survey.

A side effect of this single-qubit consideration is that we lose tensor products, and consequently scalar products as well. Single- and multi- qubit scalars will therefore have to be treated separately, and although we lose one of major features of the calculus the expressive power of the single-qubit system is still of sufficient interest to include the results here.

	$< \pi/2 >$	$< \pi/4 >$	$[0, 2\pi)$
1	$\top$	?	$\top$
$e^{i\phi}$	$(\top)$	?	?
λ	$\top$	$\top$!	$\top$!

Table 11: Existence of normal forms. "!" denotes single-qubit version only.

7.2 Stabilizer: Multi qubit

The rGS-LC ("reduced Graph State - Local Clifford") form provides a means of comparing two diagrams describing the Stabilizer segment of ZX-calculus. Since it does not provide an unique element of each equivalence class it is not a normal form; rather it provides a small subset of each equivalence class, in a way that allows easy comparison of two GS-LC forms. The relevant definitions and theorems are given below, and the general method for a given pair of diagrams goes as follows:

1. Put each diagram into GS-LC form.

2. Reduce both diagrams to form an rGS-LC pair.
3. Simplify the rGS-LC pair
4. Compare the two elements on the pair; they are identical if and only if the physical interpretations of the two original diagrams were identical.

A full description of this and the results that follow is available in [Bac13]

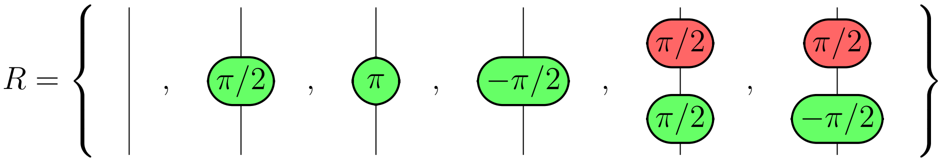
Definition. Given an n-vertex simple graph $G = (V, E)$, the corresponding **graph state** is the n-qubit state prepared as follows:

- For each vertex a qubit prepared in the state $|+\rangle$
- For each edge a controlled-Z operator applied to the appropriate qubits.

Definition. A **GS-LC** diagram consists of a graph state diagram with an arbitrary single-qubit Clifford operator applied to each output, together with an arbitrary scalar. Such Clifford operators are referred to as **vertex operators**.

Definition. An **rGS-LC** diagram is a diagram in GS-LC form satisfying the following additional constraints:

1. All vertex operators belong to the set:



2. Two adjacent vertices must not both have vertex operators that include red nodes.

Definition. A pair of rGS-LC diagrams on the same number of qubits is called **simplified** if there are no pairs of qubits p, q such that:

- p has a red node in its vertex operator in the first diagram but not the second
- q has a red node in its vertex operator in the second diagram but not the first, and
- p and q are adjacent in at least one of the diagrams.

Result. [Bac13] Any pair of rGS-LC diagrams on the same number of qubits is equal to a simplified pair.

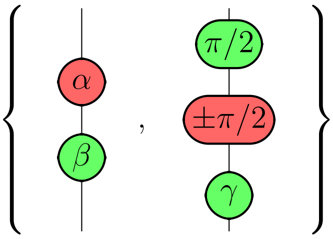
Result. [Bac13] Any stabilizer state diagram is equal to some rGS-LC diagram within the ZX-calculus.

Result. [Bac13] The two diagrams making up a simplified pair of rGS-LC diagrams have equal standard interpretations if and only if they are identical as diagrams.

7.3 Stabilizer: Single qubit

The consideration of only a single qubit inside the stabilizer segment of quantum mechanics reduces to the consideration of Clifford operators. There are 24 operators in the single-qubit Clifford group, and these can be expressed using no more than three phase gates each.

Result. [Bac14] The following set describes an unique representation for each Clifford operator:

$C :=$ 

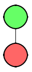
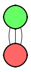
Where $\alpha, \beta, \gamma \in \{0, \pi/2, \pi, 3\pi/2\}$.

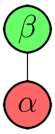
7.4 Stabilizer: Scalars

Result. [Bac16] Let D be a non-zero stabilizer scalar diagram. Then:

$$[\![D]\!] \in \{\sqrt{2}^r e^{is\pi/4} | r, s \in \mathbb{Z}\}$$

and therefore we can construct a normal form for stabilizer scalars as the combination of a diagram representing $\sqrt{2}^r$ and another representing $e^{is\pi/4}$.

- Positive real part:
  - For $r > 0$, r copies of 
  - For $r = 0$ the empty diagram, and
  - For $r < 0$, $|r|$ copies of 
- Complex phase part:

Recall that: 

$= \frac{1}{\sqrt{2}}(1 + e^{i\alpha} + e^{i\beta} - e^{i(\alpha+\beta)})$

Figure 7 uses this to build a list of representatives of $e^{si\pi/4}$ with $s = 0, \ldots, 7$.

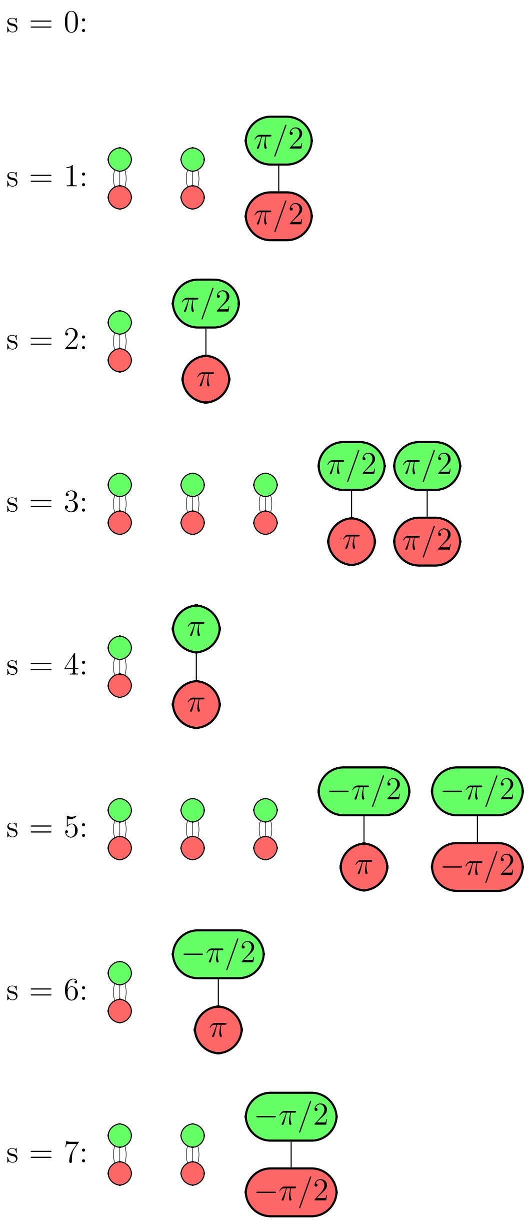


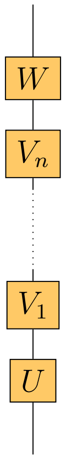
Figure 7: Representatives for $e^{is\pi/4}$.

7.5 Clifford+T: Multi qubit

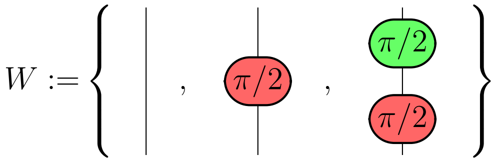
There is not currently a known normal form for ZX Clifford+T diagrams.

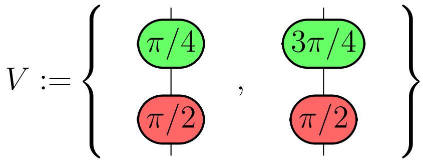
7.6 Clifford+T: Single qubit

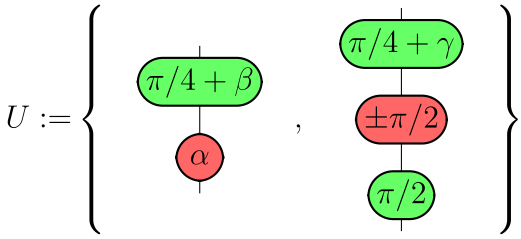
Result. [Bac14] Any single-qubit operator consisting of phase twists that are multiples of $\pi/4$ is either a single-qubit Clifford operator (see section 7.3) or it can be written in the following normal form:



Where







and $\alpha, \beta, \gamma \in \{0, \pi/2, \pi, 3\pi/2\}$, $n \in \mathbb{N}$.

7.7 Clifford+T: Scalars

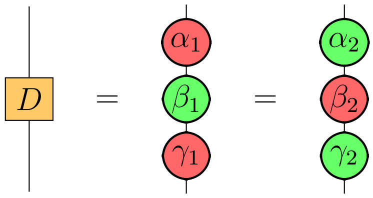
The question of a normal form for Clifford+T scalars is still open.

7.8 Universal: Multi qubit

The normal form from ZX (see section 7.11 does have a direct translation into ZX, via the functor given in [NW17], however as yet there is no neat nor illuminating version of that normal form in ZX.

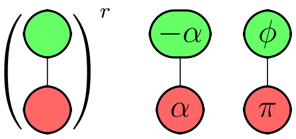
7.9 Universal: Single qubit

There exists for any unitary diagram D with one input and one output a decomposition as three phase gates (up to a modulus one complex scalar):



7.10 Universal: Scalars

Result. Any complex scalar can be represented uniquely in the following form:



Where $r \in \mathbb{N}$, $\alpha \in [0, \pi/4)$ and $\phi \in [0, 2\pi)$.

Proof. Given a complex number λ factorise it into modulus and argument:

$$\lambda = R \times (\sqrt{2}e^{i\phi})$$

Factorise the modulus again:

$$R = \sqrt{2^r} \times s$$

Where

$$\begin{aligned} &s \in [0, \sqrt{2}) \\ &r \in \mathbb{N}\,\text{minimal} \\ &\alpha := \arccos{(s/\sqrt{2})} \end{aligned}$$

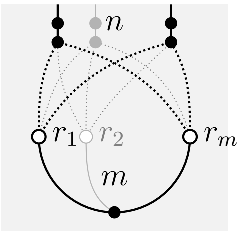
Giving us the r, α and ϕ used in the diagram. □

7.11 ZW

The ZW_R calculus exhibits the following normal form. If trying to represent:

$$\sum_{i=1}^{m} r_i |\, b_{i1} \dots b_{in} \,\rangle,$$

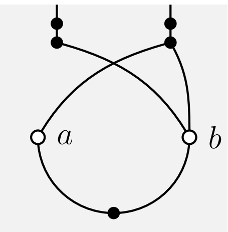
where $r_i \neq 0$, and no pair of n-tuples $(b_{i1}, \dots, b_{in})$ is equal. Use the diagram:



,

where, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, the dotted wire connecting the i-th white node to the j-th output is present if and only if $b_{ij} = 1$.

For example if trying to represent $a|\,01\,\rangle + b|\,11\,\rangle$ use:



8 Minimality and Simplicity

Efforts to reduce and simplify the set of rules we need to express the ZX-calculus are necessarily intricate in nature. The introduction and later removal of various symbols used in the literature are testament to how the rules and building-blocks of the language change over time.

The latest work on finding a minimal ruleset for Stabilizer ZX-calculus can be found in [BPW17], however the variety of approaches used in proving the necessity of each rule is too varied for it to be worth summarising here. The reader is referred to [BPW17], and to [Jea+17] for some examples of the most recent work in this area. Section 4 contains the most up-to-date versions of the rules for the ZX and ZW calculi, although research is on-going to discover which collections of rules describe which fragments of the calculi.

References

[Bac13] Miriam Backens. "The ZX-calculus is complete for stabilizer quantum mechanics". In: (2013). DOI: 10.1088/1367-2630/16/9/093021. eprint: arXiv:1307.7025.

[Bac14] Miriam Backens. "The ZX-calculus is complete for the single-qubit Clifford+T group". In: (2014). DOI: 10.4204/EPTCS.172.21. eprint: arXiv:1412.8553.

[Bac16] Miriam Backens. *Completeness and the ZX-calculus*. 2016. eprint: arXiv:1602.08954.

[BD16] M. Backens and A. N. Duman. "A Complete Graphical Calculus for Spekkens' Toy Bit Theory". In: *Foundations of Physics* 46 (Jan. 2016), pp. 70–103. DOI: 10.1007/s10701-015-9957-7. arXiv: 1411.1618 [quant-ph].

[Boy+99] P. Oscar Boykin et al. "On universal and fault-tolerant quantum computing". In: (1999). arXiv: quant-ph/9906054 [quant-ph].

[BPW17] M. Backens, S. Perdrix, and Q. Wang. "Towards a Minimal Stabilizer ZX-calculus". In: *ArXiv e-prints* (Sept. 2017). arXiv: 1709.08903 [quant-ph].

[CD08] Bob Coecke and Ross Duncan. "Automata, Languages and Programming: 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II". In: ed. by Luca Aceto et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. Chap. Interacting Quantum Observables, pp. 298–310. ISBN: 978-3-540-70583-3. DOI: 10.1007/978-3-540-70583-3_25. URL: http://dx.doi.org/10.1007/978-3-540-70583-3_25.

[CD11] Bob Coecke and Ross Duncan. "Interacting quantum observables: categorical algebra and diagrammatics". In: *New Journal of Physics* 13.4 (2011), p. 043016. URL: http://stacks.iop.org/1367-2630/13/i=4/a=043016.

[CK17] B. Coecke and A. Kissinger. *Picturing Quantum Processes*. Cambridge University Press, 2017. ISBN: 9781107104228.

[DP09] Ross Duncan and Simon Perdrix. "Graph States and the Necessity of Euler Decomposition". In: *Mathematical Theory and Computational Practice*. Ed. by Klaus Ambos-Spies, Benedikt Löwe, and Wolfgang Merkle. Vol. 5635. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 167–177. ISBN: 978-3-642-03072-7. DOI: 10.1007/978-3-642-03073-4_18.

[Got98] D. Gottesman. "The Heisenberg Representation of Quantum Computers". In: *eprint arXiv:quant-ph/9807006* (July 1998). eprint: quant-ph/9807006.

[Had17] A. Hadzihasanovic. "The algebra of entanglement and the geometry of composition". In: *ArXiv e-prints* (Sept. 2017). arXiv: 1709.08086 [math.CT].

[HV12] M. Howard and J. Vala. "Qudit versions of the qubit $\pi/8$ gate". In: 86.2, 022316 (Aug. 2012), p. 022316. DOI: 10.1103/PhysRevA.86.022316. arXiv: 1206.1598 [quant-ph].

[Jea+17] E. Jeandel et al. "Generalised Supplementarity and new rule for Empty Diagrams to Make the ZX-Calculus More Expressive". In: *ArXiv e-prints* (Feb. 2017). arXiv: 1702.01945 [quant-ph].

[JPV17] E. Jeandel, S. Perdrix, and R. Vilmart. “A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics”. In: *ArXiv e-prints* (May 2017). arXiv: `1705.11151 [quant-ph]`.

[NC10] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010. DOI: `10.1017/CBO9780511976667`.

[NW17] K. F. Ng and Q. Wang. “A universal completion of the ZX-calculus”. In: *ArXiv e-prints* (June 2017). arXiv: `1706.09877 [quant-ph]`.