Categorical Quantum Logic and the ZX-Calculus
CSC 747, Spring 2024

Bryce Goldman

Contents

1 Introduction

As quantum CPUs become more advanced, supporting a greater number of qubits and faster quantum gates, the possibility of quantum programs being deployed at the level of industry multiplies proportionately. Due to the nature of quantum circuits, applications built on qubits and quantums gates are inherently complex, with contemporary error-correcting techniques compounding the sophistication of these circuits. As such, it is increasingly important that we develop a theoretical infrastructure to verify the correctness of quantum algorithms.

Furthermore, any practical application of quantum computation will necessitate a substantial number of qubits, and consequently also an immense number of quantum gates. One of the factors currently limiting the efficacy of quantum computing is the speed of these gates – a quantum circuit with a large (serialized) gate count can run no faster than it takes for a physical qubit to pass through the hardware for those gates. Reducing the number of gates required to implement a particular quantum algorithm could potentially improve the runtime of that algorithm by orders of magnitude.

It may come as a surprise that there exists a theoretical tool which handles both of these concerns simultaneously. The *ZX-calculus* is a formal language for studying maps of qubits (i.e. quantum programs), which is equipped with a handful of *rewrite rules* to simplify and transform circuits without changing their underlying semantics. Using the ZX-calculus, we can identify when two circuits implement the same algorithm, discover relationships between different quantum gates, and reduce the complexity of quantum programs, often by hand.

In this paper, we aim to introduce the ZX-calculus from first-principles. We assume the reader is familiar with the basics of quantum computation (qubits, quantum gates, and simple circuits), as well as some linear algebra. In Section 2.1, we will cover *ZX-diagrams*, the atomic building blocks of the ZX-calculus. The focus of Section 2.2 is the calculus itself: the rewrite rules of ZX-diagrams, and their soundness. Finally, Section 2.3 is devoted to illustrating how the ZX-calculus can be applied to simplify circuits and formally verify quantum algorithms.

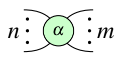
2 The ZX-Calculus

In this section we formally introduce the ZX-calculus. In Section 2.1 we define ZX-diagrams and illuminate the correspondence between ZX-diagrams and linear maps on qubits; in Section 2.2, we present the axioms for manipulating these diagrams, and some additional rewrite rules which can be derived from them; finally, in Section 2.3 we explore a handful of basic applications of the ZX-calculus to simplify quantum circuits.

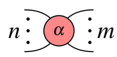
2.1 ZX-Diagrams

We begin by establishing what a ZX-diagram is, how they can be formed and combined, and how they can be interpreted as linear maps (and, for our purposes, as quantum circuits):

Definition 2.1.1. The most basic units of a ZX-diagram are *wires* and *spiders* – wires carry individual qubits throughout the diagram, while the spiders act on their input qubits to produce some number of output qubits. These spiders come in two flavors, *Z-spiders*, and *X-spiders*, depicted in green and red, respectively:

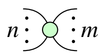


An n-input, m-output Z-spider with phase α.



An n-input, m-output X-spider with phase α.

Each spider receives a number of input wires n, coming from the left, and produces a number of output wires, extending to the right; the spider itself is also labeled with a real number[1] α in $[0,2\pi]$ (viewed as an angle, and hence subject to arithmetic modulo 2π). This label is called the *phase* of the spider. We will typically drop this label when the phase is 0, depicting the spider with an empty node instead:



We denote the Z-spider with n inputs, m outputs, and phase α by $Z_n^m(\alpha)$. Similarly, we write $X_n^m(\alpha)$ or $X(\alpha) : n \to m$ for the X-spider with the same characteristics. When $\alpha = 0$, we will typically abbreviate these spiders by Z_n^m and X_n^m (respectively), and may also drop the input/output indices when they are unimportant or clear from context. We write I to denote a single wire (with one input and output).

Note that spiders are read left to right. This convention is inherited by all ZX-diagrams, which we now define:

Definition 2.1.2. *ZX-diagrams* are generated inductively by combining spiders and wires through *vertical composition* and *horizontal composition*. The vertical composite of a pair of ZX-diagrams is given by simply stacking one on top of the other. The horizontal composite of diagrams is given by joining the output wires of one diagram to the input wires of the second. The Z-spiders and X-spiders constitute the base case for this inductive procedure.

For formal reasons[2], given ZX-diagrams D_1 and D_2, we denote their vertical composite by $D_1 \otimes D_2$, and their vertical composite, if it exists, by $D_2 \circ D_1$. Thus, all ZX-diagrams D are of one of the following forms:

(1) $D = I$, an isolated wire.

(2) $D = Z_n^m(\alpha)$ for some phase α and nonnegative integers m and n.

(3) $D = X_n^m(\alpha)$ for some phase α and nonnegative integers m and n.

(4) $D = D_1 \otimes D_2$ for some pair of ZX-diagrams D_1 and D_2.

[1] More precisely, an element of the additive quotient group $\mathbb{R}/2\pi\mathbb{R}$.

[2] One can recast ZX-diagrams in the language of category theory – there is a category **ZX**, whose objects are the nonnegative integers, such that the morphisms $n \to m$ are all ZX-diagrams with n inputs and n outputs. Composition in this category is then our horizontal composition, while vertical composition defines a symmetric monoidal structure [5] on **ZX**.

(5) $D = D_2 \circ D_1$ for some pair of ZX-diagrams D_1 and D_2, where the number of output wires of D_1 is equal to the number of input wires of D_2.

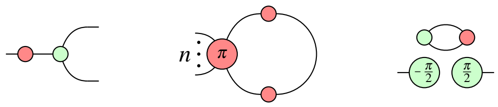
We note that although the vertical composite of two diagrams is always defined (we can always stack one on top of the other), the horizontal composite $D_2 \circ D_1$ is only defined when D_1 has a number of output wires equal to the number of input wires of D_2. Of course, if one diagram does not have sufficient wires to facilitate horizontal composition, one could simply vertically compose with the appropriate number of trivial wires first.

We also observe that vertical composition is symmetric (i.e. that $D_1 \otimes D_2 = D_2 \otimes D_1$), since it does not matter how the two disjoint components of the diagram are stacked. On the other hand, horizontal composition is not typically symmetric. Both kinds of composition are associative:

$$D_3 \square (D_2 \square D_1) = (D_3 \square D_2) \square D_1$$

where $\square$ can be either $\otimes$ or $\circ$, for all ZX-diagrams D_1, D_2, D_3 (such that their composite is well-defined).

Although they are built from only two types of generators through simple operations, ZX-diagrams can be quite sophisticated. For instance, the following are all examples of valid ZX-diagrams:



(1)

Before we delve into the relationship between ZX-diagrams and quantum computing, we will take a moment to inspect these examples, and take inventory of some of their visual features. First, we note the number of inputs and outputs in each diagram in (1): the leftmost has a single input and two outputs; the middle takes n inputs, and has no output (i.e. it is an *effect*); and the rightmost has one input and one output.

We further observe the differences in the architecture of these diagrams. Both X-spiders and Z-spiders can coexist within the same diagram, and any kind of spider can wire with any other. Note also that the number of inputs and outputs need not be the same across diagrams, nor must they be equal within a diagram. We further observe that a diagram need not be connected, as illustrated by the rightmost example in (1), and that spiders can accept zero inputs and/or outputs.

These diagrams also demonstrate both vertical and horizontal composition. The leftmost diagram in (1) is merely the horizontal composite of an X-spider with a Z-spider, both with trivial phases. The middle diagram is slightly more sophisticated: it is the horizontal composite of an X-spider with phase π with the subdiagram formed by vertically composing a pair of X-spiders of trivial phase. Finally, the rightmost diagram shows that the sequence of compositions that defines a particular diagram is non-unique: it can be viewed as the vertical composite of two subdiagrams (the two connected components of the graph, each given by horizontal composition); or as the horizontal composite of two vertical composites (the left diagram consisting of two Z-spiders, and the right consisting of an X-spider atop a Z-spider). This is symptomatic of a more general phenomenon [1], whereby horizontal composition "distributes" over vertical composition, that is, that

$$(D_1 \otimes D_2) \circ (E_1 \otimes E_2) = (D_1 \otimes E_1) \circ (D_2 \otimes E_2)$$

for all diagrams D_i and E_j for which the composites above exist.

Having (loosely) defined ZX-diagrams, and analyzed some of their basic characteristics, we now turn our attention to their connection with quantum computing. These diagrams, while not appearing especially deep or sophisticated at a glance, are quite expressive. In fact, as we will prove in Theorem 2.1.15, ZX-diagrams can model any quantum circuit, or more generally, any linear map $\mathbb{C}^{2^n} \to \mathbb{C}^{2^m}$. We define the translation between diagrams and linear transformations inductively, beginning with the generators (i.e. the spiders):

Definition 2.1.3. The *linear interpretation* of $Z_n^m(\alpha)$, denoted $[\![Z_n^m(\alpha)]\!]$ is the linear map $\mathbb{C}^{2^n} \to \mathbb{C}^{2^m}$ given by

$$\begin{aligned}[\![Z_n^m(\alpha)]\!] &:= |0\rangle^{\otimes m} \langle 0|^{\otimes n} + e^{i\alpha} |1\rangle^{\otimes m} \langle 1|^{\otimes n} \\ &= |0\cdots 0\rangle \langle 0\cdots 0| + e^{i\alpha} |1\cdots 1\rangle \langle 1\cdots 1|.\end{aligned}$$

Similarly, the linear interpretation of $X_n^m(\alpha)$ is the linear map $\mathbb{C}^{2^n} \to \mathbb{C}^{2^m}$ given by

$$\begin{aligned}\llbracket X_n^m(\alpha) \rrbracket &:= \left|+\right\rangle^{\otimes m} \left\langle+\right|^{\otimes n} + e^{i\alpha} \left|-\right\rangle^{\otimes m} \left\langle-\right|^{\otimes n} \\ &= \left|+\cdots+\right\rangle \left\langle+\cdots+\right| + e^{i\alpha} \left|-\cdots-\right\rangle \left\langle-\cdots-\right|.\end{aligned}$$

The linear interpretation of I is simply the identity map $\mathbb{C}^2 \to \mathbb{C}^2$.

Now, given a ZX-diagram D, if $D = D_1 \otimes D_2$ for ZX-diagrams D_1 and D_2 then we define the linear interpretation of D by

$$\llbracket D \rrbracket := \llbracket D_1 \rrbracket \otimes \llbracket D_2 \rrbracket,$$

where $A \otimes B$ denotes the tensor product of linear maps. Note that if D_i has n_i inputs and m_i outputs, then $\llbracket D \rrbracket : \mathbb{C}^{2^{n_1+n_2}} \to \mathbb{C}^{2^{m_1+m_2}}$.

Finally, if $D = D_2 \circ D_1$ for diagrams D_1 and D_2 (whose horizontal composite is well-defined), we define the linear interpretation of D by

$$\llbracket D \rrbracket := \llbracket D_2 \rrbracket \circ \llbracket D_1 \rrbracket,$$

where $B \circ A$ denotes the usual composition of linear maps. Since $n_2 = m_1$ by hypothesis this is well-defined, and $\llbracket D \rrbracket : \mathbb{C}^{2^{n_1}} \to \mathbb{C}^{2^{m_2}}$.

Since every ZX-diagram D is either a spider or formed from smaller ZX-diagrams via vertical and horizontal composition, Definition 2.1.3 is comprehensive.

Remark 2.1.4. Note that, although we can interpret every ZX-diagram as some linear map between qubits, this map will not necessarily be unitary. This is manifestly true from the fact that some ZX-diagrams have different numbers of inputs and outputs. In fact, ZX-diagrams are far more expressive; as we'll observe in Example 2.1.9, not only can we represent quantum circuits with ZX-diagrams, we can also use them to represent quantum states.

Now, let us examine the matrices corresponding to some basic ZX-diagrams:

Example 2.1.5. The Z-spider Z_1^1 (with trivial phase) has interpretation

$$\llbracket Z_1^1 \rrbracket = \left|0\right\rangle \left\langle 0\right| + \left|1\right\rangle \left\langle 1\right| = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Although slightly less obvious, the X-spider X_1^1 likewise has interpretation

$$\llbracket X_1^1 \rrbracket = \left|+\right\rangle \left\langle +\right| + \left|-\right\rangle \left\langle -\right| = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

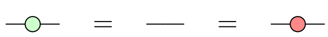
since

$$\left|+\right\rangle \left\langle +\right| = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

and

$$\left|-\right\rangle \left\langle -\right| = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

Thus, both Z_1^1 and X_1^1 are no-ops (i.e. are equivalent as linear maps to an isolated wire I):



We will soon see in Section 2.2 that this is in fact one of the rules used to simplify ZX-diagrams.

Example 2.1.6. As we've observed, spiders can have zero inputs. For instance, we have that

$$\llbracket Z_0^1 \rrbracket = \left|0\right\rangle + \left|1\right\rangle,$$

which is $\left|+\right\rangle$ up to a factor of $\frac{1}{\sqrt{2}}$. Similarly,

$$\llbracket Z_0^1(\pi) \rrbracket = \left|0\right\rangle - \left|1\right\rangle,$$

which is $|-\rangle$ multiplied by $\sqrt{2}$. On the other hand,

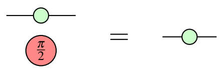
$$[\![X_0^1]\!] = |+\rangle + |-\rangle\,,$$

and

$$[\![X_0^1(\pi)]\!] = |+\rangle - |-\rangle\,,$$

which are $|0\rangle$ and $|1\rangle$, respectively, up to a (nonzero) scalar coefficient.

Definition 2.1.7. We will subsequently drop nonzero coefficients when considering ZX-diagrams, and consider two diagrams to be *equivalent* if their linear interpretations differ by some nonzero scalar factor. Within a ZX-diagram, a scalar is represented by a disjoint (sub)diagram with no input nor output (like the top subdiagram in the leftmost diagram of (1)). Thus, our choice to disregard nonzero scalar factors can be interpreted graphically as deleting all such subdiagrams, allowing us to write, for instance,

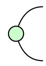


This reduction modulo scalars is grounded in our interpretation of diagrams as quantum circuits: by convention we ignore global phase, which is equivalent to identifying states which differ by a nonzero coefficient (since, in the quantum setting, every state vector is unital).

An important generalization of Example 2.1.6 allows us to capture the notion of entangled states. First, we must make formal what a “state” is:

Definition 2.1.8. A diagram with only output wires is a *state*.

Example 2.1.9. Consider the two qubit state B with a single, phaseless Z-spider with no inputs and two outputs:

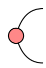


Examining the matrix of the linear interpretation of B, we find that

$$[\![B]\!] = |00\rangle + |11\rangle\,,$$

which we recognize as the *Bell state* (up to a scalar factor, as usual).

Similarly, let B' be the state obtained from a single, phaseless X-spider with the same number of inputs and outputs:



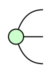
Then, ignoring nonzero scalar coefficients, we have

$$\begin{aligned}[\![B']\!] &= |{+}{+}\rangle + |{-}{-}\rangle \\ &= (|0\rangle + |1\rangle)^{\otimes 2} + (|0\rangle - |1\rangle)^{\otimes 2} \\ &= (|00\rangle + |10\rangle + |01\rangle + |11\rangle) + (|00\rangle - |10\rangle - |01\rangle + |11\rangle) \\ &= |00\rangle + |11\rangle\,,\end{aligned}$$

so we find that $[\![B]\!] = [\![B']\!]$. Consequently, we can comfortably drop the spider from both diagrams altogether, writing

It is no coincidence that this equivalence resembles that of Example 2.1.5 – these are two manifestations of the same rewrite rule, which allows phaseless spiders with a single input and output to be removed from a diagram (see Axiom 2.2.2).

Example 2.1.10. By increasing the number of output wires, we can entangle more qubits. For instance, the *GHZ state* can be described by the diagram



which clearly corresponds to the matrix

$$|000\rangle + |111\rangle\,,$$

as desired. However, unlike with the Bell state, we can no longer drop the Z-spider from the diagram. To see why, observe that, via routine computation, the interpretation of the phaseless, three qubit state X-spider is

$$|{+}{+}{+}\rangle + |{-}{-}{-}\rangle = |000\rangle + |011\rangle + |110\rangle + |101\rangle\,.$$

Thus the relationship observed in Example 2.1.5 between phaseless spiders does not generalize beyond those with exactly two wires.

So far we've seen how to translate several simple "low-dimensional" diagrams into the context of quantum computing. We now turn our attention to interpreting some of the more commonly encountered quantum gates as ZX-diagrams.

The astute reader may have already noted the suggestive classification of the spiders with the characters "Z" and "X" together with a phase. As one may have guessed, spiders indeed correspond to rotations (by their phase) about the corresponding axis of the Bloch sphere:

Example 2.1.11. Fix an angle $\alpha \in [0, 2\pi]$. Then by definition we have

$$[\![Z_1^1(\alpha)]\!] = |0\rangle\langle 0| + e^{i\alpha}|1\rangle\langle 1| = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha}, \end{pmatrix}$$

which corresponds to a rotation of the Bloch sphere about the z-axis by an angle of α. Similarly, by routine calculation, we have

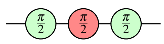
$$[\![X_1^1(\alpha)]\!] = |+\rangle\langle +| + e^{i\alpha}|-\rangle\langle -| = \begin{pmatrix} \cos\frac{\alpha}{2} & -i\sin\frac{\alpha}{2} \\ -i\sin\frac{\alpha}{2} & \cos\frac{\alpha}{2} \end{pmatrix}$$

(up to some nonzero global phase), which corresponds to a rotation of the Bloch sphere about the x-axis by angle of α.

For instance, the Pauli Z-gate is equivalent to the ZX-diagram $Z_1^1(\pi)$, while the Pauli X-gate is equivalent[3] to the diagram $X_1^1(\pi)$. Of course, it follows immediately that the Pauli Y-gate is given by the ZX-diagram $Z_1^1 \circ X_1^1$.

It would be impossible to claim the ZX-calculus holds any relevance to the world of quantum computing without also addressing superposition. Fortunately, we can also express Hadamard gates as ZX-diagrams:

Example 2.1.12. Consider the ZX-diagram H, depicted below:

$H :=$ 

Computing its linear interpretation using Example 2.1.11, we find that

$$[\![H]\!] = [\![Z(\tfrac{\pi}{2})X(\tfrac{\pi}{2})Z(\tfrac{\pi}{2})]\!] = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}\begin{pmatrix} \sqrt{2} & -i\sqrt{2} \\ -i\sqrt{2} & \sqrt{2} \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

But this is precisely the matrix of the Hadamard gate, so we can encode superposition within ZX-diagrams. Because of its prevalence in quantum computing, the[4] Hadamard diagram has a special place in the ZX-calculus, and we will subsequently denote this diagram with a unique symbol:

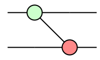
$H :=$

Finally, no discussion of quantum computing would be complete without mention of the CNOT (controlled not) gate, which, when coupled with Hadamard gates, is responsible for most of the utility of quantum programs by enabling a circuit to entangle its qubits. As the reader may have guessed, these too can be formulated as a ZX-diagram:

[3]To be precise, we've omitted a global phase of $e^{i\pi/2}$ in this identification.

[4]The use of the definite article is slightly inaccurate. There are many different, equivalent diagrams which represent the Hadamard gate, but by the soundness result in Theorem 2.2.11 later in this section we are free to choose a representative diagram from among them as we please.

Example 2.1.13. Let C be the ZX-diagram

$C :=$ 

or written syntactically,

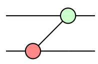
$$C := (X_2^1 \otimes I)(I \otimes Z_1^2).$$

Computing the linear interpretation of C, we obtain

$$[\![C]\!] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

(up to some nonzero global phase), which is indeed the matrix for the CNOT gate (with the top wire, least significant qubit, as the control).

Now, let C' be the ZX-diagram

$C' :=$ 

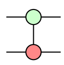
that is,

$$C' := (Z_2^1 \otimes I)(I \otimes X_1^2).$$

Through some tedious but straightforward calculation, we find that

$$[\![C']\!] = [\![C]\!],$$

so in fact both C and C' are representatives of the CNOT gate. Because of this equality, we can denote the ZX-diagram for the CNOT gate by



without ambiguity.

This gives us our first taste of the *topological symmetry* of the ZX-calculus: if we slide the nodes in one around, without changing the way they connect to each other, then the linear interpretation remains constant. The semantics of a ZX-diagram are said to be *topologically invariant*. We'll see shortly in Section 2.2 that this is true in much greater generality – any time one diagram can be "deformed" into another while preserving how each spider connects, they will have the same linear interpretation.

Remark 2.1.14. The astute reader may, at this point, have concerns. We have seen above that the quantum gates we are familiar with may require ZX-diagrams be combined in particular ways. That is, we may not be able to translate any arbitrary ZX-diagram (with a unitary linear interpretation) back into the world of quantum computation. If our objective is to mutate the diagram representing some circuit into some other, simpler diagram, how can we be sure the final product will still be expressible as a quantum circuit?

In practice this is not an obstacle, at least for our purposes. When manipulating ZX-diagrams using the ZX-calculus (see Section 2.2 and Section 2.3), although we may end up with some intermediate diagrams which cannot be translated into quantum gates, we will often be able to recover a valid quantum circuit from the diagrams lying at either end of some sequence of transformations. The purpose of using the ZX-calculus in this way is only to establish the equivalence of a pair of quantum programs through its formal rewrite rules, so it is not particularly important that its rewrite rules stay entirely on the ground (so long as they end at the same "elevation" at which they started).

We conclude this section with a formal proof of the *universality* of ZX-diagrams: that is, that every quantum circuit (and in fact, every linear map $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$, as we will show) is represented by a ZX-diagram. This property is desirable (necessary, even, depending on one's religion) for the ZX-calculus to be useful. After all, if we cannot express every circuit as a ZX-diagram, we have no hope of using it to study arbitrarily complex quantum programs.

Theorem 2.1.15 (Universality of ZX-Diagrams)**.** Every linear map $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ is equal to[5] the linear interpretation of some ZX-diagram.

[5]This is true even when taking scalar coefficients into account, which will be evident from the proof.

To prove this result, we build all linear maps from the ground up, through a series of lemmata. We begin with the simpler task of proving that every complex number can be expressed as a ZX-diagram. For this, we will require the following auxiliary result:

Lemma 2.1.16. Let $\mathscr{S} \subseteq \mathbb{C}$ be the multiplicatively closed set generated by the complex numbers

$$\frac{1}{\sqrt{2}}, \sqrt{2}e^{i\alpha}, 1+e^{i\alpha}$$

as α ranges over $[0, 2\pi]$. Then $\mathscr{S} = \mathbb{C}$. That is, every complex number can be expressed as the product of numbers of the form above.

Proof. First, we will show that the subset of real numbers $[0,1]$ is contained in $\mathscr{S}$. We note that, given $\alpha \in [0, 2\pi]$, we have

$$\begin{aligned} |1+e^{i\alpha}| &= |1+\cos\alpha + i\sin\alpha| \\ &= \sqrt{1+2\cos\alpha+\cos^2\alpha+\sin^2\alpha} \\ &= \sqrt{2}\sqrt{1+\cos\alpha}, \end{aligned}$$

and since $\cos(0) = 1$ and $\cos(\pi) = -1$, by the continuous value theorem the map $\alpha \mapsto |1+e^{i\alpha}|$ takes all values in $[0,1]$.

Now, let $r \in [0,1]$, and let $\alpha_r \in [0,2\pi]$ be such that

$$|1+e^{i\alpha_r}| = \frac{r}{\sqrt{2}}.$$

Additionally, let $\theta_r = \mathrm{Arg}(1+e^{i\alpha_r})$. Then we have

$$\frac{r}{\sqrt{2}} = e^{-i\theta_r}(1+e^{i\alpha_r}),$$

so

$$r = \sqrt{2}e^{-i\theta_r} \cdot (1+e^{i\alpha_r}) \in \mathscr{S}.$$

Thus $[0,1] \subseteq \mathscr{S}$.

Now, suppose $x \in \mathbb{R}_{\geq 0}$ is any arbitrary nonnegative real number. Then there exists some sufficiently large integer n such that $\frac{x}{\sqrt{2}^n} \in [0,1]$, and hence we have $\frac{x}{\sqrt{2}^n} \in \mathscr{S}$ by the argument above. But $\sqrt{2} = \sqrt{2}e^{i\cdot 0} \in \mathscr{S}$, so since $\mathscr{S}$ is multiplicatively closed we have $x \in \mathscr{S}$.

Finally, let $z \in \mathbb{C}$ be a complex number. Then $|z|$ is a nonnegative real number, and we have

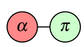
$$z = |z|e^{i\theta},$$

where $\theta := \mathrm{Arg}(z)$. But $e^{i\theta} = \frac{1}{\sqrt{2}} \cdot \sqrt{2}e^{i\theta}$ is contained in $\mathscr{S}$, as is $|z|$ as we've shown above, so $z \in \mathscr{S}$. Therefore $\mathscr{S} = \mathbb{C}$, as desired. □

Corollary 2.1.17. Every (complex) scalar is the linear interpretation of some ZX-diagram.

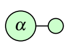
Proof. This follows immediately from Lemma 2.1.16 together with the verification that

(1) The linear interpretation of



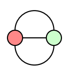
is $\sqrt{2}e^{i\alpha}$.

(2) The linear interpretation of



is $1+e^{i\alpha}$.

(3) The linear interpretation of



is $\frac{1}{\sqrt{2}}$.

□

Next, we will show that we can express every unitary map $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ (up to global phase) with ZX-diagrams:

Proposition 2.1.18. Every (unitary) linear map of qubits, and hence every quantum circuit, is the linear interpretation of some ZX-diagram, modulo some scalar factor.

Proof. First, we recall from Example 2.1.11 that every rotation of the Bloch sphere about the z-axis and x-axis can be represented by a ZX-diagram. Since every unitary transformation of the Bloch sphere is the composite of such rotations [4] (up to global phase), every unitary map on a single qubit is the linear interpretation of some ZX-diagram.

Additionally, as seen in Example 2.1.13, the CNOT gate is expressible by ZX-diagrams. But every unitary map on arbitrarily-many qubits is the composite of CNOT gates and single qubit unitary maps [6], so indeed every unitary map of qubits is represented by some ZX-diagram. □

Now that we can express all unitary maps between qubits with ZX-diagrams, we can also express all quantum states as well. The idea is that there is always some unitary transformation from the initial state $|0\cdots 0\rangle$ to an arbitrary quantum state. Since we also can find a ZX-diagram for every complex scalar, this will enable us to produce any complex vector lying in qubit space. We now make this formal:

Corollary 2.1.19. Given a positive integer n, every nonzero complex vector $v \in \mathbb{C}^{2^n}$ is the linear interpretation of some ZX-diagram.

Proof. First, let u be some quantum state vector (that is, a vector lying on the unit sphere in $\mathbb{C}^{2^n}$). Then u can be expressed as $u = U|0\cdots 0\rangle$ for some unitary map $U : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$. To see this, we observe that since both vectors are unitary, we can use the Graham-Schmidt process to find orthonormal bases $u_1, \ldots, u_{2^n}$ and $w_1, \ldots, w_{2^n}$ for $\mathbb{C}^{2^n}$, where $u_1 = u$ and $w_1 = |0\cdots 0\rangle$. Let $U(u_i) := w_i$ for $i = 1, \ldots, 2^n$. This fully specifies the linear map $U : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$, and since $\{u_i\}$ and $\{w_j\}$ are orthonormal bases U is automatically unitary.

Now, let $v \in \mathbb{C}^{2^n}$ be an arbitrary nonzero vector. Then $u := \frac{v}{||v||}$ is a unit vector (i.e. a quantum state vector), and by Corollary 2.1.17 we can express $||v||$ using ZX-diagrams. Moreover, by the argument above, together with Proposition 2.1.18, we can express U (and thus u) with ZX-diagrams as well. Taking the vertical composite of the diagrams for $||v||$ and u yields a ZX-diagram whose linear interpretation is v. □

Finally, we are equipped with the tools needed to prove the full universality of ZX-diagrams:

Proof of Theorem 2.1.15. Let $T : \mathbb{C}^{2^n} \to \mathbb{C}^{2^m}$ be a linear map. The Choi-Jamiołkowski isomorphism [2] defines a linear bijection between linear maps $\mathbb{C}^{2^n} \to \mathbb{C}^{2^m}$ and vectors of $\mathbb{C}^{2^n} \otimes \mathbb{C}^{2^m} \cong \mathbb{C}^{2^{m+n}}$. Hence, there exists a vector $J(T) \in \mathbb{C}^{2^{n+m}}$, which canonically corresponds to T. By Corollary 2.1.19, we can express $J(T)$ as the linear interpretation of some ZX-diagram D, and, by bending [10] the n output wires of the state D that correspond to the n input qubits of T we obtain a ZX-diagram D' whose linear interpretation is T. □

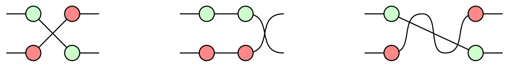
The upshot of what we have shown above is that, given any quantum circuit, we can find some ZX-diagram which represents it. In fact, we can find many such diagrams – the entire utility of the ZX-calculus is the ability to identify seemingly distinct diagrams which share the same underlying semantics. In the following section, we will begin to understand the axioms of the ZX-calculus, used to rewrite diagrams without altering their corresponding linear maps.

2.2 Axioms of the ZX-Calculus

In the previous section we introduced ZX-diagrams as a graphical language for representing quantum programs, and studied their expressive power. In this section, we will answer the question of how they are useful, and what benefit is conferred by the translation from circuit to ZX-diagram. The utility of this diagrammatic language comes from its *grammar*; that is, from the rules that allow us to rewrite diagrams in a way that preserves their semantics (i.e. their linear interpretations). This grammar is called the ZX-calculus.

The rewrite rules of the ZX-calculus enable their user to manipulate quantum circuits by hand, potentially simplifying a sophisticated diagram to a more compact form, verify the equivalence of two disparate programs, or probe qubits for entanglement (see Section 2.3). The majority of this section will be devoted purely to enumerating and justifying the rewrite of the ZX-calculus, followed by a brief discussion about the *soundness* and *completeness* of the calculus.

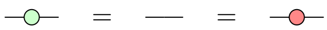
Before we begin reviewing the rewrite rules of the ZX-calculus in earnest, we make a brief note about topological invariance, previously mentioned in Example 2.1.13. The ZX-calculus is agnostic to the exact shape of diagrams, and is only concerned with how each spider is connected to its neighbors. That is, ZX-diagrams can be deformed, twisted, stretched, and contracted, as long as each spider retains the same input and output terminals for each of its wires. For instance, the following are all equivalent ZX-diagrams:



Remark 2.2.1. We refer to each rule as an "axiom" of the calculus, although this is a mild abuse of terminology. We offer a derivation (or at least some intuitive justification) for each rule, so a priori they are not axioms in the traditional sense. The intention with this choice of language is to suggest that the ZX-calculus could be defined without any foreknowledge of quantum circuits or the linear interpretation of a ZX-diagram. The rewrite rules exist independently, and could be defined before verifying their soundness.

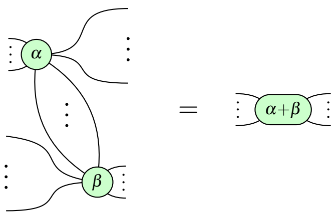
The first rewrite rule states that spiders with trivial phases are no-ops:

Axiom 2.2.2 (Identity)**.** We have equivalences

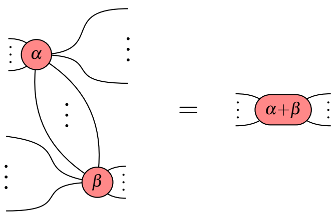


This follows from the discussion in Example 2.1.5. While the identity rewrite rule may not seem very useful on its own, its utility lies in the way it interacts with the other laws of the ZX-calculus. For instance, one often utilizes the identity rule alongside the next axiom, which tells us how spiders of the same color interact:

Axiom 2.2.3 (Spider Fusion)**.** For each pair of phases $\alpha, \beta \in [0, 2\pi]$, we have



and

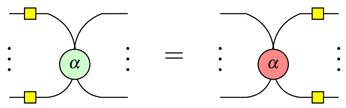


where $\alpha + \beta$ is taken modulo 2π.

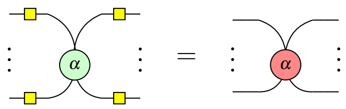
Informally, the spider fusion axiom of the ZX-calculus says that connected spiders of the same color can be merged together, with their phases added. In the one-dimensional case, for instance, this says that a rotation by an angle α around the z-axis (resp. x-axis) followed by another rotation by an angle β around the z-axis is the same as a single rotation around the z-axis (resp. x-axis) by the angle $\alpha + \beta$. A sufficiently motivated (or bored) reader can verify spider fusion by hand, by checking that the diagrams on either side of the equality have the same linear interpretation; for brevity, we opt to omit this tedious calculation and leave it as an exercise.

Axiom 2.2.4 (Hadamard Color Change)**.** The H-gate is an involution:

Additionally, for each phase $\alpha \in [0, 2\pi]$ we have



or equivalently (using the fact that Hadamard gates are involutions),

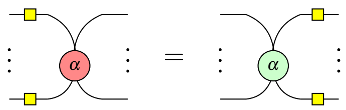


The first equivalence follows easily from Example 2.1.12, since it is easily verified that the linear map corresponding to the H-gate is self-inverse. The second half of the rewrite rule is essentially a corollary of the fact that $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$:

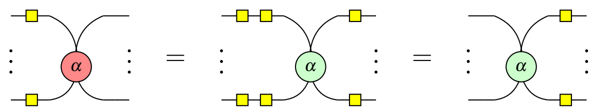
$$\begin{aligned} Z_n^m(\alpha) \circ H^{\otimes n} &= |0\rangle^{\otimes m} (\langle 0|H)^{\otimes n} + e^{i\alpha} |1\rangle^{\otimes m} (\langle 1|H)^{\otimes n} \\ &= |0\rangle^{\otimes m} \langle +|^{\otimes n} + e^{i\alpha} |1\rangle^{\otimes m} \langle -|^{\otimes n} \\ &= (H|+\rangle)^{\otimes m} \langle +|^{\otimes n} + e^{i\alpha} (H|-\rangle)^{\otimes m} \langle -|^{\otimes n} \\ &= H^{\otimes m} \circ X_n^m(\alpha). \end{aligned}$$

We can also easily derive from this axiom the dual rewrite rule where the H-gates lie on the left of an X-spider:

Proposition 2.2.5. For each phase $\alpha \in [0, 2\pi]$ we have



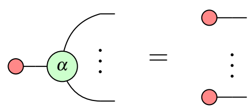
Proof. The result follows immediately from Axiom 2.2.4, from which we derive the following chain of equivalences:



□

The dual rewrite rule of Proposition 2.2.5 is a special case of a much more general consequence of Axiom 2.2.4. Namely, every rewrite rule has a dual rewrite rule, obtained by exchanging the color of every spider in each diagram. The idea is that, by flipping the color of every spider and placing H-gates along each input and output wire of that spider, we need only reduce the diagram to the same form of the rewrite rule we aim to dualize. But because every diagram is built from spiders, and since H-gates are involutions, each of the H-gates we inserted must cancel with some other H-gate we added. For brevity we will subsequently only state each rewrite rule in the ZX-calculus once, leaving its dual implicit, but the reader should keep in mind that one is free to flip the color of each spider without changing the semantics of the calculus.

Axiom 2.2.6 (Eigenstate Duplication)**.** For each phase $\alpha \in [0, 2\pi]$ we have



This remains true (up to a nonzero scalar coefficient) if all of the X-spiders above are given a phase of π instead of zero.

Axiom 2.2.6 follows from the following simple calculation:

$$(|0\rangle^{\otimes m}\langle 0| + e^{i\alpha}|1\rangle^{\otimes m}\langle 1|)\,|0\rangle = |0\rangle^{\otimes m}\langle 0|0\rangle + e^{i\alpha}|1\rangle^{\otimes m}\langle 1|0\rangle = |0\rangle^{\otimes m}.$$

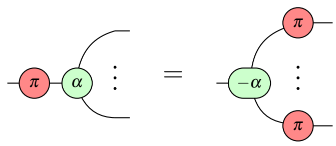
Replacing $|0\rangle$ with $|1\rangle$, we instead have

$$(|0\rangle^{\otimes m}\langle 0| + e^{i\alpha}|1\rangle^{\otimes m}\langle 1|)\,|1\rangle = e^{i\alpha}|1\rangle^{\otimes m}.$$

Note that Axiom 2.2.6 does not hold when the input state is not a basis state (i.e. if the X-spider on the left-hand side has any phase besides 0 and π). The geometric intuition behind this rewrite rule is that $|0\rangle$ and $|1\rangle$ are the fixed points for any rotation of the Bloch sphere about the z-axis, so these (and only these) state vectors should be preserved by such a rotation.

Although the eigenstate duplication rule only allows the basis states $|0\rangle$ and $|1\rangle$ (or $|+\rangle$ and $|-\rangle$, if we instead consider the dual rewrite rule with spider colors flipped) to be copied, we can also push Pauli gates through spiders of the opposite color. In particular, X-spiders (resp. Z-spiders) with a phase of π commute (or, more precisely, "super-commute," due to a sign change) with Z-spiders (resp. X-spiders) of any phase, as described by the next rewrite rule:

Axiom 2.2.7 (π-Commutativity)**.** For each phase $\alpha \in [0, 2\pi]$ we have



To verify that this rewrite rule is compatible with our expectations, we first check that the diagrams in Axiom 2.2.7 are equivalent when $\alpha = 0$. Recall from Example 2.1.11 that the X-gate (with linear map X) is represented by the ZX-diagram $X_1^1(\pi)$, so the linear interpretation of the left-hand diagram above is

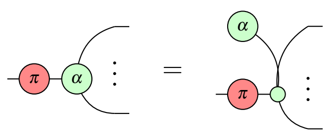
$$(|0\rangle^{\otimes m}\langle 0| + |1\rangle^{\otimes m}\langle 1|)X = |0\rangle^{\otimes m}\langle 1| + |1\rangle^{\otimes m}\langle 0|.$$

Similarly, the linear interpretation of the right-hand side is

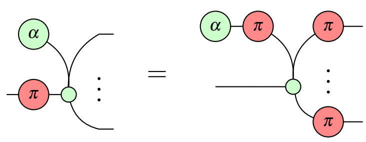
$$X^{\otimes m}(|0\rangle^{\otimes m}\langle 0| + |1\rangle^{\otimes m}\langle 1|) = (X|0\rangle)^{\otimes m}\langle 0| + (X|1\rangle)^{\otimes m}\langle 1| = |0\rangle^{\otimes m}\langle 1| + |1\rangle^{\otimes m}\langle 0|,$$

so indeed these ZX-diagrams are equivalent.

Now, for the general case, we observe that we can extract the phase from the Z-spider of the left-hand diagram in Axiom 2.2.7 using spider fusion in reverse:



Next, we apply the phaseless case of Axiom 2.2.7 already established above:

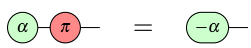


(2)

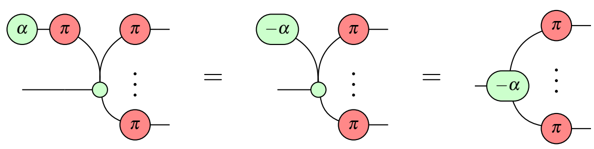
Moreover,

$$X(|0\rangle + e^{i\alpha}|1\rangle) = |1\rangle + e^{i\alpha}|0\rangle = e^{i\alpha}(|0\rangle + e^{i\alpha}|1\rangle),$$

so up to a nonzero scalar coefficient we have an equivalence



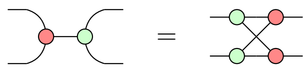
Performing this substitution in (2), and then applying spider fusion again, we obtain



as desired.

Now that we've started building a repertoire of rewrite rules for the ZX-calculus, we can see how their use allows us to translate tedious equational reasoning into relatively simple graphical transformations. To complete the ZX-calculus, we only need to introduce two more rewrite rules; although one could continue deriving additional equivalences between classes of ZX-diagrams, all others could be recovered from the handful we review in this section (in fact, as we will see shortly, the rewrite rule Axiom 2.2.9 is also a derivative of the other rewrite rules of this section).

Axiom 2.2.8 (Bialgebra Rule)**.** There is an equivalence of ZX-diagrams



This rewrite rule is so-named because of its relation to the concept of a *bialgebra* in mathematics. Informally, a bialgebra is a structure A equipped with both a monoidal structure $A \times A \to A$ and a comonoidal structure $A \to A \times A$, which are compatible with each other [9]. Part of this compatibility is closely related to the equivalence in Axiom 2.2.8.

To understand this comparison, consider the set $P = \{0,1\}$. There is a canonical bialgebra structure on P, whose monoidal "multiplication" is given by the XOR operation $\oplus$, and whose comonoidal "comultiplication" is expressed by the "copy" operation Δ, which maps $x \in P$ to the pair $(x,x) \in P \times P$. These operations make the following diagram commute:

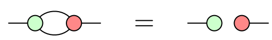
$$\begin{tikzcd}
P^2 \arrow[r, "\Delta\times\Delta"] \arrow[d, "\otimes"'] & P^4 \arrow[d, "\otimes\times\otimes"] \\
P \arrow[r, "\Delta"'] & P^2
\end{tikzcd}$$

That is, we can either XOR together a pair of bits and then copy them, or we can copy each bit and then XOR the results, and obtain the same output either way.

This is exactly the same relationship expressed in Axiom 2.2.8. The phaseless X-spider X_2^1 is behaves similarly to the XOR operation (with the basis state vectors $|0\rangle$ and $|1\rangle$ viewed as classical bits), the phaseless Z-spider Z_1^2 is analogous to the operation which copies its input bit across two parallel output wires, and the bialgebra rule simply makes this comparison precise. (We omit the formal verification of Axiom 2.2.8, because it is tedious and not particularly enlightening.)

Closely related to the bialgebra rule is the final rewrite rule we discuss: the *Hopf algebra rule*:

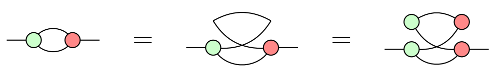
Axiom 2.2.9 (Hopf Algebra Rule)**.** There is an equivalence of ZX-diagrams



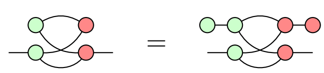
(3)

In mathematics, a *Hopf algebra* is a bialgebra equipped with a *unit* (an element of the bialgebra) and *counit* (informally, a kind of "evaluation map" on the bialgebra) which are coherent with each other [9]. For instance, our previous example of the bialgebra $(P,\oplus,\Delta)$ is a Hopf algebra, with unit $0 \in P$ and counit given by the constant map $P \to \{0\}$. In the language of this Hopf algebra, the aforementioned "coherence" states that copying a bit and then taking the XOR of the result will always result in the output bit 0 – that is, that composing comultiplication with multiplication is equal to applying the counit and then the unit. The analogy between this bialgebra and the phaseless ZX-spiders is precisely the equivalence depicted in Axiom 2.2.9.

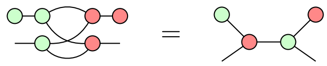
The Hopf algebra rule is not strictly necessary to express the semantics of the ZX-calculus, as it can be recovered from the other rewrite rules that precede it. To better acclimate the reader to manipulating diagrams using the ZX-calculus in preparation for Section 2.3, we outline this derivation explicitly. First, utilizing the topological invariance of ZX-diagrams to rewrite the left-hand side of (3) and then applying Axiom 2.2.2, we obtain the equivalence below:



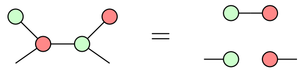
Next, we use Axiom 2.2.3 to obtain the equivalence



Applying Axiom 2.2.8, we find that



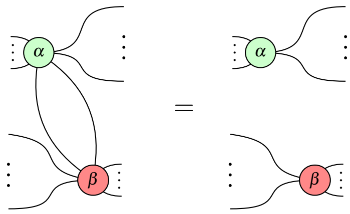
Finally, we use Axiom 2.2.6 to yield



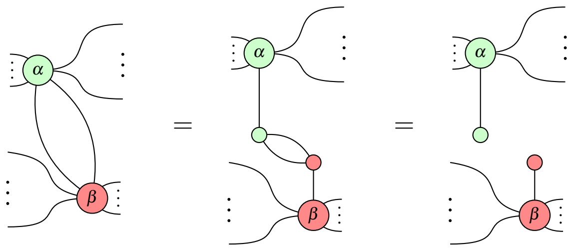
noting that the scalar coefficient can be dropped to yield the desired equivalence in (3).

From the Hopf rule we are able to derive the following useful auxiliary result, which allows us to disentangle spiders of different colors:

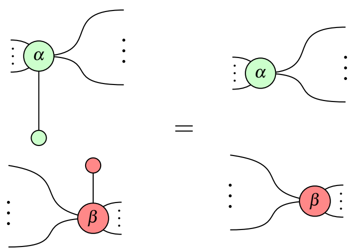
Lemma 2.2.10 (Spider Decoupling)**.** For each pair of phases $\alpha, \beta\ in[0, 2\pi]$, we have an equivalence of diagrams:



Proof. First, applying Axiom 2.2.3 and then Axiom 2.2.9, we have an equivalence



Finally, we can use Axiom 2.2.3 again to obtain



as desired. □

We conclude this section by discussing the formal properties of the language of the ZX-calculus. As mentioned in Section 2.1, the property of *universality* was desirable in order for ZX-diagrams (the *syntax* of our language) to be as expressive as possible. Similarly, we wish for the ZX-calculus, the *semantics* of our language, to capture all the information encoded in relationships between the linear maps they represent.

Formally, one captures how "faithful"[6] the logic of the ZX-calculus reflects the logic of quantum programs through two, dual concepts: *soundness* and *completeness*. Soundness is the property that, given a pair of ZX-diagrams D and D', if D can be transformed into D' using the rewrite rules of the ZX-calculus then $[\![D]\!] = [\![D']\!]$ (or, more precisely, that their linear interpretations only differ by some nonzero scalar coefficient). Conversely, completeness states that, if D and D' have the same linear interpretation, then one can be derived from the other using the rewrite rules.

[6]If one delves sufficiently deep into the underlying category-theoretic origins of the ZX-calculus, one will uncover that the linear interpretation functor being *faithful* in the technical sense is simply a reformulation of the soundness of the ZX-calculus.

The ZX-calculus, as we've introduced it, is sound but (unfortunately) not complete. However, one can easily extend the language of the ZX-calculus such that it satisfies both properties [11]. Although the lack of completeness of our bare-bones calculus is not ideal, this merely says that there may be some linear maps (i.e. quantum circuits) which implement the same program, such that the ZX-calculus would be unable to formally verify their equivalence. In this use-case, soundness is significantly more important: it states that whenever the ZX-calculus equates a pair of diagrams, their corresponding linear maps (i.e. quantum circuits) are indeed equal. Consequently, as long as a particular diagram is vulnerable to the rewriting rules of the ZX-calculus (and, as we will see in Section 2.3, such diagrams are in abundance), we are guaranteed that any derivations we perform are valid.

Theorem 2.2.11. The ZX-calculus is sound.

Proof. We aim to show that, given diagrams D and D' such that D' can be derived from D using rewrite rules, we have $[\![D]\!] = [\![D']\!]$ (up to a scalar factor). It suffices to show that, for each rewrite rule R which identifies diagram D with $R(D)$, we have $[\![R(D)]\!] = [\![R]\!]$, that is, that R preserves the linear interpretation of each diagram it is applied to. But this is precisely what we have shown throughout this section: that each rewrite rule of the ZX-calculus equates a pair of diagrams with the same underlying interpretation. The result then follows, since linear interpretation commutes with both vertical and horizontal composition; that is, if $[\![D]\!] = [\![D']\!]$ and E is some ZX-diagram, then we have

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

and

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

(and likewise for horizontal composition with E on the right). Thus, since each diagram is built inductively by combining smaller diagrams through vertical and horizontal composition, applying a rewrite rule to a subdiagram of D to form D' preserves linear interpretation, as desired. $\square$

2.3 Practical Derivations

Finally, we are equipped with the machinery required to study actual quantum programs using the ZX-calculus. In this section we will present a handful of applications of practical applications of the ZX-calculus to quantum computing.

One of the primary use-cases for the ZX-calculus is to verify that a quantum circuit implements the correct program. Classically, if one wants to check that the linear map on qubits produced by a particular circuit is as desired, the image of each basis state vector would need to be checked. In an n-qubit system, there are 2^n such state vectors, so this quickly becomes tiresome (and is not feasible, at least for a human, for sufficiently large circuits).

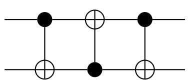
In keeping with the core philosophy of quantum computing, we would like some way to parallelize this verification procedure. The ZX-calculus provides a framework to do exactly that. By translating a circuit into a ZX-diagram, we are able to use the rewrite rules of Section 2.2 to attempt to massage that diagram into one whose linear interpretation is easier to compute, which can then be compared to the desired unitary transformation.

Example 2.3.1 (SWAP Implementation)**.** As a first example, we consider the SWAP gate, which takes a pair of qubits as input and exchanges them, that is,

$$\mathrm{SWAP}\,|q_1 q_0\rangle = |q_0 q_1\rangle$$

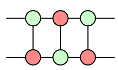
As a ZX-diagram, this operation would be represented by a pair of isolated wires that cross over each other:

To implement the SWAP gate, one typically would use the following quantum circuit, consisting of three serialized CNOT gates:

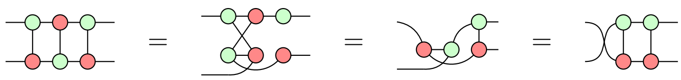


(4)

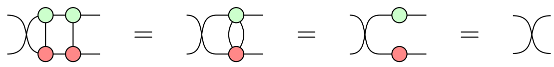
Using Example 2.1.13, this circuit is represented by the ZX-diagram



We will now show, using only the ZX-calculus, that this diagram is equivalent to that of the SWAP gate. First, we deform the diagram (while preserving its underlying topology), then apply Axiom 2.2.8, and finally rearrange the diagram again:



Using Axiom 2.2.3, Lemma 2.2.10, and Axiom 2.2.2, we then obtain

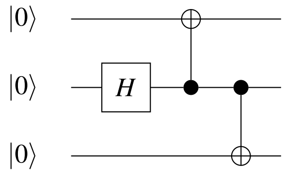


Thus (4) indeed implements the SWAP map, as claimed.

Note that nowhere in Example 2.3.1 did we need to actually compute the linear interpretation of any ZX-diagrams. All of that work has been front-loaded in Section 2.2, when checking each rewrite rule for soundness.

Because we can model quantum states, as well as circuits, as ZX-diagrams, we can also verify that the output of a particular circuit is correct.

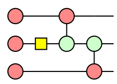
Example 2.3.2 (GHZ Preparation)**.** Consider the circuit



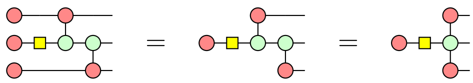
(5)

which, we claim, initializes the GHZ state $|000\rangle + |111\rangle$ when provided an input of $|000\rangle$. We will verify the correctness of this quantum circuit using the ZX-calculus.

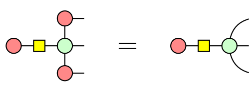
As usual, we start by translating (5) into a ZX-diagram. We recall from Example 2.1.9 that the state $|0\rangle$ is represented by $X_0^1(0)$, so using Example 2.1.13 and Example 2.1.12 this circuit (with its input state $|000\rangle$ initialized) is expressed by



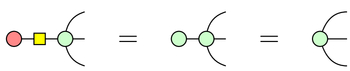
We begin by applying Axiom 2.2.3 twice, first to the X-spiders, and then the Z-spiders, to reduce the total number of gates:



Next using Axiom 2.2.2, we can eliminate the rightmost X-spiders:



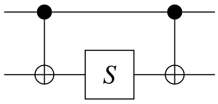
Finally, applying Axiom 2.2.4 to eliminate the H-gate, and then using Axiom 2.2.3, we conclude that



This ZX-diagram represents the GHZ state, as noted in Example 2.1.10, so indeed the circuit in (5) implements the desired program.

The ZX-calculus can also be useful for identifying general equivalences between common sub-circuits found in quantum programs. These are like the rewrite rules of Section 2.2, with the additional guarantee that the result on either end can be cast back to valid quantum circuits. For instance, a rule that allows, say, a Hadamard gate to be pushed through a particular circuit architecture could be useful in a diagram where that sub-circuit is found somewhere in the middle, if that Hadamard gate can later be cancelled with another H-gate (thus eliminating two gates from the circuit).

Example 2.3.3 (Pauli Pushing)**.** Consider the circuit



(6)

and suppose one wishes to precompose with the Pauli circuit $Z \otimes X$. A natural question to ask is whether this can be turned into post-composition with some other circuit. This could be useful if, for example, one has a larger circuit containing

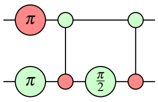
$$\mathrm{CNOT} \circ (S \otimes I) \circ \mathrm{CNOT} \circ (Z \otimes X) \circ Q, \tag{7}$$

where Q is some two-qubit circuit which terminates with a CNOT gate. Being able to rewrite (7) as

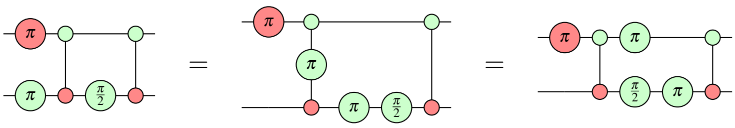
$$R \circ \mathrm{CNOT} \circ (S \otimes I) \circ \mathrm{CNOT} \circ Q$$

for some two-qubit circuit R would allow us to cancel adjacent CNOT gates, which, as we know, are very expensive. Provided the new circuit R is not more costly than $Z \otimes X$, on a large scale this could greatly improve the efficiency of a circuit.

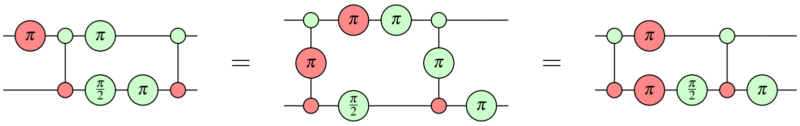
First, we translate the composite of $Z \otimes X$ with (6) into the following ZX-diagram using Example 2.1.11, Example 2.1.13, and noting that the S-gate, a rotation of $\frac{\pi}{2}$ about the Z-axis, is equivalent to $Z_1^1(\frac{\pi}{2})$, again by Example 2.1.11:



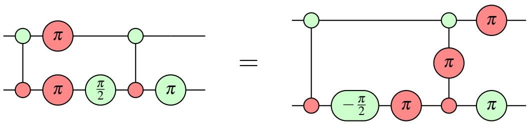
Our aim is to push the Z-gate and X-gate on the left-hand side to the right-hand side. We begin by using Axiom 2.2.7 and Axiom 2.2.3 to obtain



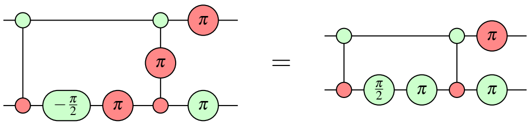
Next, we apply Axiom 2.2.7 to both the top-left X-gate and bottom-right Z-gate, followed again by Axiom 2.2.3:



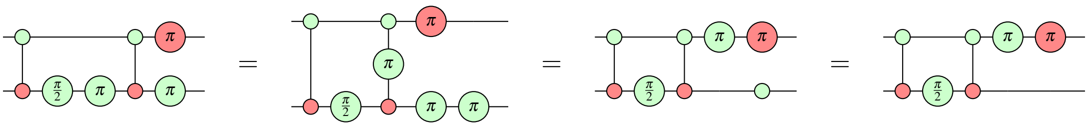
Using Axiom 2.2.7 on the remaining X-gates yields



so from Axiom 2.2.3 we deduce that



Finally, pushing the remaining middle Z-gate through the diagram using Axiom 2.2.7, Axiom 2.2.3, and Axiom 2.2.2, we have



Thus, $Z \otimes X$ indeed can be pushed through (6), and the resulting post-composed circuit R is no more sophisticated[7] than the initial Pauli circuit.

[7]If decoherence is a concern this circuit is marginally less stable, since now both Pauli gates are applied to a single qubit in sequence.

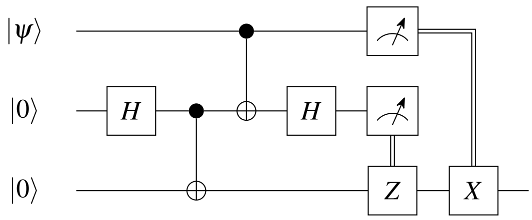
The ZX-calculus can also be useful when studying more complex quantum protocols:

Example 2.3.4 (Quantum Teleportation)**.** Using the ZX-calculus, we can unpack the quantum teleportation algorithm to show that the information the sender (Alice) transmits is passed to the receiver (Bob) unaltered.

Before proceeding, we briefly recall the teleportation protocol. Alice and Bob share a pair of entangled qubits, initialized in the Bell state

$$|\Phi\rangle := |00\rangle + |11\rangle \,.$$

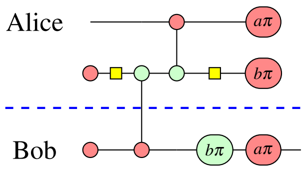
Half of $|\Phi\rangle$ is owned by Alice, while the other half is owned by Bob. Alice also has a single qubit quantum state $|\psi\rangle$, which she wishes to communicate to Bob. Alice has $|\psi\rangle$ act on her half $|\Phi_A\rangle$ of $|\Phi\rangle$, then measures $|\psi\rangle$ and $|\Phi_A\rangle$, and stores the results in classical bits a and b (respectively). The result of this measurement is transferred to Bob, who can then recover $|\psi\rangle$ from $|\Phi_B\rangle$, his half of $|\Phi\rangle$, by applying a sequence of Pauli gates to it, using the pair of classical bits (a,b) as controls. The quantum circuit for this routine is depicted below:



(8)

The bottom two qubits are entangled to prepare the Bell state $|\Phi\rangle$. Alice owns the top two qubits, while the bottom qubit belongs to Bob.

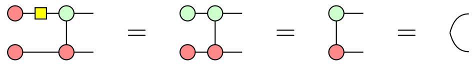
We can rewrite (8) as the ZX-diagram



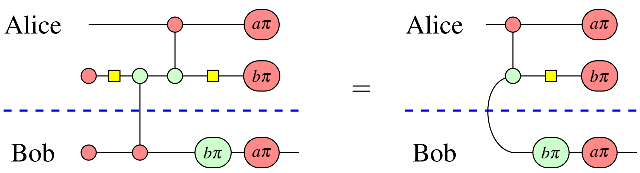
(9)

where the top input wire receives $|\psi\rangle$; here a is the measured state of $|\psi\rangle$, while $|\Phi_A\rangle$ and $|\Phi_B\rangle$ are the bottom two qubits, initialized to $|0\rangle$ and then entangled. Here a is the measured state of $|\psi\rangle$, while b is the measured state of $|\psi_A\rangle$; both are either 0 or 1.

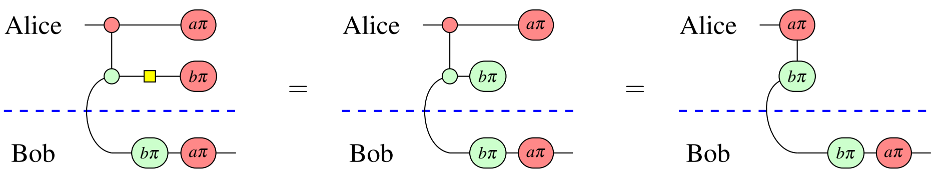
First, we can simplify the preparation of $|\Phi\rangle$ to recover the Bell state representation of Example 2.1.9. Using Axiom 2.2.4, then Axiom 2.2.3, and finally Axiom 2.2.2, we have



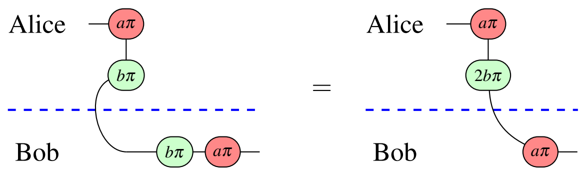
Substituting this equivalence back into (9), we have



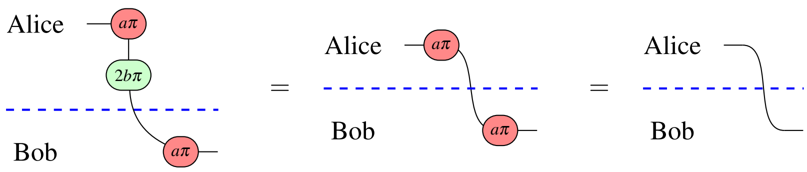
We now aim to simplify this diagram so that it consists of a single, isolated wire passing from Alice to Bob, as this is what it means for the qubit $|\psi\rangle$ to be transmitted without any change in information. Applying Axiom 2.2.4 and Axiom 2.2.3, we have



Using Axiom 2.2.3 again, and perturbing the shape of the diagram slightly, we obtain



But $b \in \{0,1\}$, so $2b\pi$ is a trivial phase, and hence by applying Axiom 2.2.2, Axiom 2.2.3, and then Axiom 2.2.2 again (as $2a\pi$ is also a trivial phase) we have



as desired. Hence the quantum teleportation protocol indeed transfers the data of Alice's qubit $|\psi\rangle$ to Bob.

Note that Alice no longer has access to the wire corresponding to $|\psi\rangle$, which is a symptom of the conversation of quantum information. Since Alice must measure $|\psi\rangle$ she destroys her access to that data, so in terms of the ZX-diagram there should be no output wire in Alice's domain (which is what we observe above).

3 Conclusion

In Section 2 we introduced a powerful tool for formally studying quantum circuits, in the form of the ZX-calculus.

We established the language for this tool in Section 2.1, wherein we presented ZX-diagrams, and explored their connections with quantum circuits. These diagrams were built from a simple alphabet, consisting of only two families of generators – Z-spiders and X-spiders – from which complex diagrams were built by stacking and concatenating (i.e. running in parallel and sequence, respectively) smaller subdiagrams. We saw that these diagrams could be translated into linear maps of qubits, and that we could realize many familiar quantum gates as manifestations of particular ZX-diagrams. We concluded Section 2.1 by formalizing the expressive power of this language, by proving that all linear maps between qubits could be formulated as ZX-diagrams.

Throughout the following section, Section 2.2, we established the grammar of the language of ZX-diagrams, in the form of rewrite rules. These rules specify how one ZX-diagram can be transformed into another in a way which preserves its underlying semantics (namely, its linear interpretation). For each axiom of the ZX-calculus, we proved (or at least outlined) its soundness, and discussed the basic properties of the calculus.

Finally, in Section 2.3, we worked through a handful of practical applications of the ZX-calculus, to illustrate how it can be used to perform derivations on actual quantum circuits. We used the ZX-calculus to formally verify quantum programs, such as the GHZ preparation circuit and quantum teleportation protocol, and to deduce interactions between Pauli gates with a simple circuit.

3.1 Further Reading

This paper only scratched the surface of the ZX-calculus and its many applications. As a next step, we encourage the interested reader to refer to the first five chapters of [10], which offers an introduction to the subject that is both thorough and beginner-friendly. Another good resource for beginners is this Pennylane blogpost, by Romain Moyard, which offers a less formal presentation of much of the same material covered in this paper, and provides some additional examples of applications of the ZX-calculus to quantum machine learning.

To the more mathematically mature reader, we recommend [1] and Chapter Seven of [10], which recasts the ZX-calculus in the language of category theory (see [5] or [7] for a rigorous treatment of the subject in general). Many of the definitions and results presented in this paper can be streamlined when reformulated in this context, and deeper connections with other fields of mathematics are more easily uncovered from the category-theoretic point of view.

Finally, the more implementation-oriented reader may benefit from reading [3], and attempting to utilize the corresponding Python package PyZX, whose documentation can be found here. PyZX (purportedly) provides an API for simplifying quantum circuits using the ZX-calculus. However, the author of this paper was unsuccessful at automating the optimization process, and was unable to recover the same derivations produced by hand using PyZX (see Section 3.3). We hope the brave reader who attempts this finds more success.

3.2 Acknowledgements

I would like to thank both the instructors of CSC 747, Prof. Wes Bethel and Prof. Daniel Huang, for their hard work throughout the Spring 2024 semester. This course was immensely rewarding, and I am very grateful for both professors' diligence and patience (particularly with my many questions).

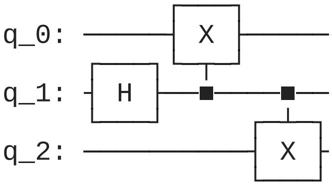
3.3 Appendix

Initially, part of the intention with this paper was to present a demonstration of the PyZX Python [3] package at the end of Section 2.3. Unfortunately, I was unable to produce the intended output with this software. Despite providing PyZX with quantum circuits I could simplify by hand with the ZX-calculus, the `simplify.full_reduce` and `optimize.full_optimize` methods appeared to increase total gate count and circuit complexity. Included below is a monument to my failed attempts to wrangle PyZX:

In [2]:
```
import pyzx as zx
import numpy as np

from qiskit import QuantumCircuit, visualization
from qiskit import qpy
```

In [3]:
```
# Attempt 1: automate the simplification of the GHZ preparation circuit.
qc1 = QuantumCircuit(3)
qc1.h(1)
qc1.cx(1, 0)
qc1.cx(1, 2)

qc1.draw()
```

Out[3]:



In [4]:
```
# Save circuit in QASM format
qc1.qasm(formatted = True, filename = "ghz.qasm")

# Load circuit with PyZX
circ1 = zx.Circuit.load("ghz.qasm")

# Note: H-gates in the ZX-diagram are depicted as blue edges
zx.draw(circ1)

# Note: this is indeed the correct ZX-diagram for the circuit qc1.
```

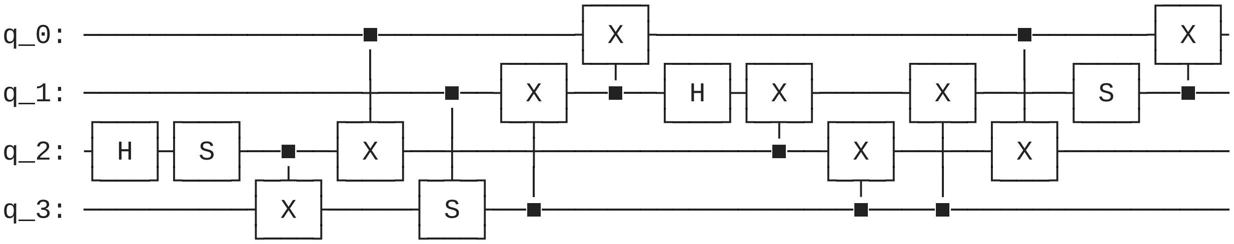
```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[3];
h q[1];
cx q[1],q[0];
cx q[1],q[2];
```

In [5]:
```
# Convert circuit to graph
g1 = circ1.to_graph()

# Simplify ZX-diagram
zx.simplify.full_reduce(g1)
g1.normalize()
zx.draw(g1)

# The expected output is the GHZ state. Instead we have converted every spider
# into a Z-spider by inserting a ton of H-gates?
```

In [7]:
```
# Attempt 2: reduce large circuit to simple entangled state
qc2 = QuantumCircuit(4)
qc2.h(2)
qc2.s(2)
qc2.cx(2, 3)
qc2.cx(0, 2)
qc2.cs(1, 3)
qc2.cx(3, 1)
qc2.cx(1, 0)
qc2.h(1)
```

```
qc2.cx(2, 1)
qc2.cx(3, 2)
qc2.cx(3, 1)
qc2.cx(0, 2)
qc2.s(1)
qc2.cx(1, 0)

qc2.draw()
```

Out[7]:



In [19]:

```
# Save circuit in QASM format
qc2.qasm(formatted = True, filename = "large.qasm")

# Load circuit with PyZX
circ2 = zx.Circuit.load("large.qasm")

zx.draw(circ2)

# Note: this is indeed the correct ZX-diagram for the circuit qc1.
```

```
OPENQASM 2.0;
include "qelib1.inc";
gate cs q0,q1 { p(pi/4) q0; cx q0,q1; p(-pi/4) q1; cx q0,q1; p(pi/4) q1; }
qreg q[4];
h q[2];
s q[2];
cx q[2],q[3];
cx q[0],q[2];
cs q[1],q[3];
cx q[3],q[1];
cx q[1],q[0];
h q[1];
cx q[2],q[1];
cx q[3],q[2];
cx q[3],q[1];
cx q[0],q[2];
s q[1];
cx q[1],q[0];
```

In [22]:

```
# Convert circuit to graph
g2 = circ2.to_graph()

# Simplify ZX-diagram
zx.simplify.full_reduce(g2)
g2.normalize()
zx.draw(g2)

# Convert ZX-diagram back to circuit
c_opt = zx.extract_circuit(g2.copy())
zx.draw(c_opt)

# The expected output is a GHZ state and some other single qubit tensor factor.
# In particular, this circuit can be reduced to a circuit containing only two
# Z-spiders, which is clearly not PyZX is producing.
```

Unfortunately, Jupyter notebook also could not manage to convert the PyZX diagrams into PDF graphics. The omitted diagrams can be found below:

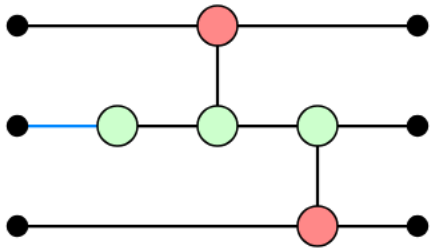


Figure 2: PyZX-generated diagram for `qc1`.

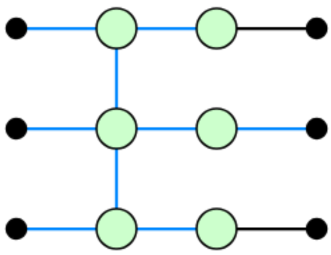


Figure 3: ZX-diagram for `qc1`, “simplified” by PyZX.

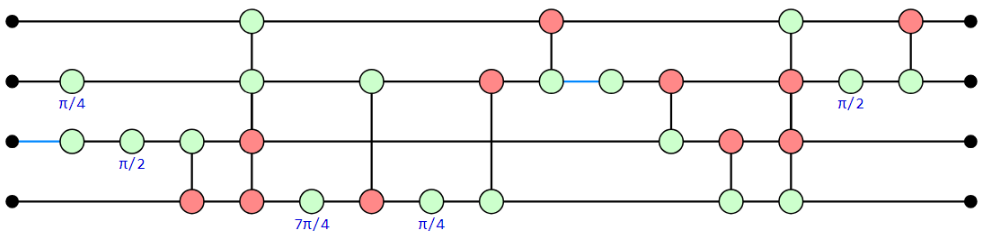


Figure 4: PyZX-generated diagram for `qc2`.

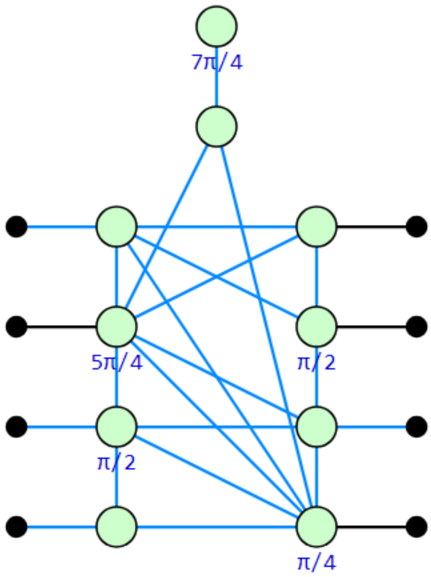


Figure 5: ZX-diagram for `qc2`, “simplified” by PyZX.

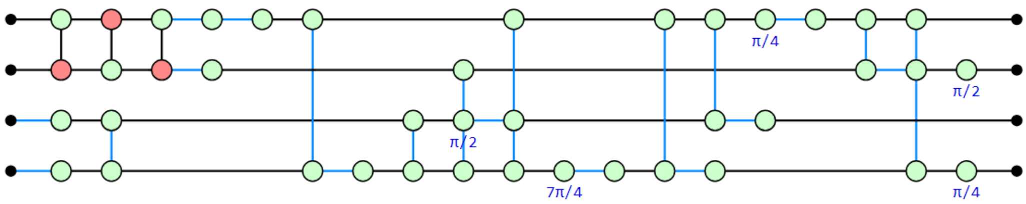


Figure 6: Circuit corresponding to the “reduced” ZX-diagram for `qc2`.

4 References

[1] D. Cicala. Categorifying the zx-calculus. *Electronic Proceedings in Theoretical Computer Science*, 266:294–314, Feb. 2018. doi: 10.4204/eptcs.266.19

[2] M. Frembs and E. G. Cavalcanti. Variations on the choi-jamiolkowski isomorphism, 2022.

[3] A. Kissinger and J. van de Wetering. Pyzx: Large scale automated diagrammatic reasoning. *Electronic Proceedings in Theoretical Computer Science*, 318:229–241, May 2020. doi: 10.4204/eptcs.318.14

[4] A. M. Krol, A. Sarkar, I. Ashraf, Z. Al-Ars, and K. Bertels. Efficient decomposition of unitary matrices in quantum circuit compilers. *Applied Sciences*, 12(2):759, Jan. 2022. doi: 10.3390/app12020759

[5] S. M. Lane. *Categories for the working mathematician*. Springer New York, 1971.

[6] M. Nakahara and T. Ohmi. *Quantum Computing: From Linear Algebra to Physical Realizations*. CRC Press, 2008.

[7] E. Riehl. *Category Theory in Context*. Courier Dover Publications, 2017.

[8] X. Sun and F. He. A first step to the categorical logic of quantum programs. *Entropy*, 22(2), 2020. doi: 10.3390/e22020144

[9] R. G. Underwood. *An Introduction to Hopf Algebras*. Springer New York, NY, 2011.

[10] J. van de Wetering. ZX-calculus for the working quantum computer scientist, 2020.

[11] Q. Wang. Completeness of the ZX-calculus, 2023.