Cambridge Quantum

Kindergarten Quantum Mechanics Graduates...
or how I learned to stop gluing Lego together and love the ZX-calculus

by Bob Coecke, Dominic Horsman, Aleks Kissinger, Quanlong Wang

Cambridge Quantum
BOB COECKE
bob.coecke@cambridgequantum.com
QUANLONG WANG
harny.wang@cambridgequantum.com

Université Grenoble Alpes, France
DOMINIC HORSMAN
dom.horsman@gmail.com

Oxford University, United Kingdom
ALEKS KISSINGER
aleks.kissinger@cs.ox.ac.uk

Kindergarden quantum mechanics graduates

...or how I learned to stop gluing LEGO together and love the ZX-calculus

Bob Coecke[†], Dominic Horsman[⋆], Aleks Kissinger[‡], Quanlong Wang[†]

[†] Cambridge Quantum Computing Ltd.
bob.coecke/harny.wang@cambridgequantum.com
[⋆] Université Grenoble Alpes. dom.horsman@gmail.com
[‡] Oxford University. aleks.kissinger@cs.ox.ac.uk

February 23, 2021

Abstract

This paper is a 'spiritual child' of the 2005 lecture notes *Kindergarten Quantum Mechanics* [23], which showed how a simple, pictorial extension of Dirac notation allowed several quantum features to be easily expressed and derived, using language even a kindergartner can understand. Central to that approach was the use of pictures and pictorial transformation rules to understand and derive features of quantum theory and computation. However, this approach left many wondering 'where's the beef?' In other words, was this new approach capable of producing new results, or was it simply an aesthetically pleasing way to restate stuff we already know?

The aim of this sequel paper is to say 'here's the beef!', and highlight some of the major results of the approach advocated in Kindergarten Quantum Mechanics, and how they are being applied to tackle practical problems on real quantum computers. Toward that end, we will focus mainly on what has become the Swiss army knife of the pictorial formalism: the *ZX-calculus*, a graphical tool for representing and manipulating complex linear maps on 2^N dimensional space. First we look at some of the ideas behind the ZX-calculus, comparing and contrasting it with the usual quantum circuit formalism. We then survey results from the past 2 years falling into three categories: (1) completeness of the rules of the ZX-calculus, (2) state-of-the-art quantum circuit optimisation results in commercial and open-source quantum compilers relying on ZX, and (3) the use of ZX in translating real-world stuff like natural language into quantum circuits that can be run on today's (very limited) quantum hardware.

We also take the title literally, and outline an ongoing experiment aiming to show that ZX-calculus enables children to do cutting-edge quantum computing stuff. If anything, this would truly confirm that 'kindergarten quantum mechanics' wasn't just a joke.

1 Introduction

A bit over 15 years ago, some people (including some of us) started using a nice trick. Take plain old Dirac 'bra-ket' notation, the typical go-to language for calculation in quantum computing, and write it in 2D, where matrix multiplication looks like 'plugging boxes together' and tensor

product looks like 'putting boxes side by side', for example:

$$(g(f \otimes \mathbb{1})(|\psi\rangle \otimes |\phi\rangle)) \otimes \mathbb{1} \qquad \rightsquigarrow$$
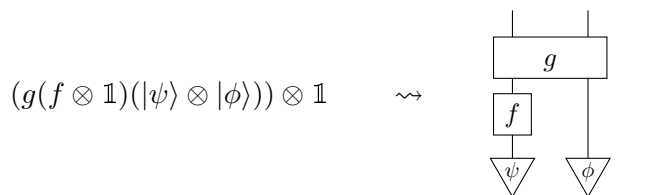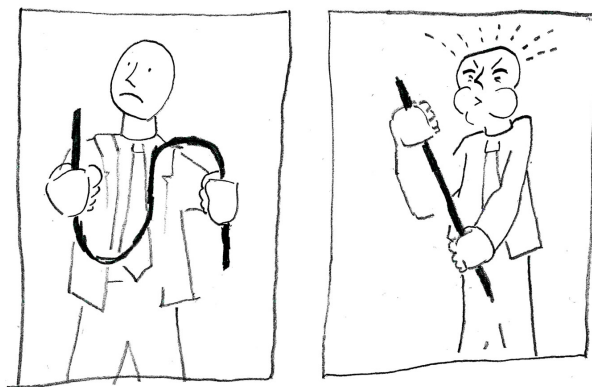

So far, things don't look so different from quantum circuits. However, the key trick was to write the maximally entangled state, and its adjoint, as bent pieces of wire:

$$\smile \quad := \quad |00\rangle + |11\rangle \qquad\qquad \frown \quad := \quad \langle 00| + \langle 11| \tag{1}$$

Then, the main idea behind quantum teleportation, which basically amounts to this equation:

$$(((\langle 00| + \langle 11|) \otimes \mathbb{1})(\mathbb{1} \otimes (|00\rangle + |11\rangle))) = \mathbb{1}$$

becomes something visually very intuitive:

Hence, kindergarten quantum mechanics became a thing. Now, these sort of tricks weren't entirely new, as a certain Nobel Prize winner named Roger Penrose got so fed up in the 1970's with staring at indices in the tensor notation of relativity, and for that purpose invented exactly the same kinds of pictures. So we were in pretty good company.

A good start, but, how much mileage can you get out of these sort of tricks? Well, as it turns out, a lot: one can teach an entire quantum computing and quantum foundations course in these terms.[1] How much is really new? That is, can drawing pictures of quantum processes allow us to do things we couldn't do before? Or is it just an art project?

This is where the *ZX-calculus* comes in. The ZX-calculus is a graphical language for expressing quantum computations, mainly over qubits. While it's been around since 2008, things have only really started booming around 2018, with the appearance of several major results:

(1) The ZX-calculus has been 'completed', which means all equations concerning quantum processes involving qubits that can be derived using linear algebra can also be obtained using a handful of graphical rules [68, 109]. This consolidates the promises made in the early days of kindergarten quantum mechanics, that graphical reasoning should not merely be seen as a helpful gadget, but as a genuine alternative to the Hilbert space formalism.

---

[1]This course has been running since 2012 at the University of Oxford, and this course formed the basis for 'dodo book' [39]. https://www.cs.ox.ac.uk/teaching/courses/2019-2020/quantum/

(2) For certain quantum circuit optimisation problems, ZX-based methods now outperform the state of the art, e.g. [44] showed T-counts that were up to 50% better than known techniques at the time of publication. These simplifications are important for making the problems fit on existing quantum computers, and has played an important role in the design of commercial quantum compilers such as Cambridge Quantum Computing's t|ket⟩ [103].

(3) ZX-calculus recently enabled a team to convert grammar-aware natural language processing [42] into variational quantum circuits [27] suitable for running on existing, small-scale quantum hardware, resulting in the first implementation of quantum natural language processing on a quantum computer [86].

This paper is not intended to be a tutorial, but is an easy-going introduction and a survey of some recent successes. If you are in need of a more detailed manual on how to use ZX-calculus, several other resources are already available. For example, the book [39] gives an extensive introduction to the broad subject of pictorial quantum reasoning, leading up to a detailed presentation of ZX-calculus. While this is a pretty hefty tome (850 pages), it's full of pictures and has been taught multiple times (at Oxford, Nijmegen and Peking) in about 20 hours of lecture time. A much shorter introductory ZX-tutorial is [31], and an extensive, up-to-date introduction with many practical worked examples is [107]. There is moreover a forthcoming secondary school book [37] that we discuss in Section 8.

2 ZX: LEGO for quantum computing

We will introduce ZX-calculus by comparing it to standard quantum circuit language, and in particular, by explaining the manner in which ZX-calculus (quite literally) stretches beyond how we can manipulate and reason with quantum circuits.

ZX-language. Typical primitives of quantum circuit language include the CNOT-gate and certain single qubit gates like Z-phase gates and the Hadamard gate. We denote these here as follows:

$$
 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \alpha := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \qquad := \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}
$$

While Z-phase gates are typically taken to be diagonal in the standard (or 'Z') basis, we can conjugate by the Hadamard gate to get X-phase gates, which are diagonal in the Hadamard (or 'X') basis:

These two kinds of phase gates can now be used to build other things, for example, the Hadamard gate itself now arises, up to a scalar factor (which we ignore), to it's *Euler decomposition* in terms of phase gates:
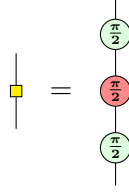

Rather than just using the standard phase gates as building blocks for other gates, ZX-calculus uses generalisations thereof, allowing one to vary the number of incoming and outgoing wires of these phase gates. More specifically, we can generalise the phase gates to 'spiders':

$$
\begin{aligned}
&\quad := \quad |0\ldots0\rangle\langle0\ldots0| + e^{i\alpha}|1\ldots1\rangle\langle1\ldots1| \\[2em]
&\quad := \quad |+\ldots+\rangle\langle+\ldots+| + e^{i\alpha}|-\ldots-\rangle\langle-\ldots-|
\end{aligned}
\tag{2}
$$

Without resorting to bra-ket notation, a Z-spider with m legs in and n legs out is a $2^n \times 2^m$ matrix with exactly 2 non-zero elements:

$$
\quad := \quad
\begin{pmatrix}
1 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & e^{i\alpha}
\end{pmatrix}
$$

and an X-spider can be made from a Z-spider much like we did with phase gates:
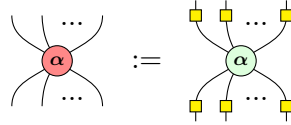
Putting no α means $\alpha = 0$, e.g.

$$
\quad := \quad
\begin{pmatrix}
1 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & 0 & 1
\end{pmatrix}
$$

It then follows that the cups and caps of (1), as well as many basic quantum states and effects, are special cases of spiders:

where $|+\rangle = 1/\sqrt{2}\big(|0\rangle + |1\rangle\big)$, $|-\rangle = 1/\sqrt{2}\big(|0\rangle - |1\rangle\big)$, and we have ignored some normalisation factors.

Spiders are all that the language of ZX-calculus consists of. Why can ZX-calculus get away with only these? Since we can now build the CNOT-gate from these spiders as follows:


(3)

That this is indeed the case is something that can be easily checked using matrices. So in particular, the CNOT-gate doesn't have to be treated as a primitive anymore, but breaks down in two smaller pieces. Once we have phase gates and the CNOT-gate, we know that we can reproduce any quantum circuit made up of any gates.

What is the upshot of doing this? More specifically, why is this better than using standard circuits? The true power of ZX-calculus arises from the fact that these smaller pieces in (3) are very easy to work with, in the sense that the rules that govern them are easy to figure out, remember, and do calculations with. Also, there aren't many of them. In contrast, coming up with all the rules that govern fixed sets of quantum gates is really hard, and little is known beyond the case of very limited gate sets [2] or small fixed numbers of qubits.

For example, it was shown in [100] that there *does* exist a set of quantum circuit equations rules that suffices to prove all true equations for 2-qubit circuits built from these gates:
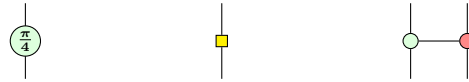
That is, the gates we introduced at the beginning of this section, but with Z-phases restricted to $\alpha = \pi/4$. However, some of the rules are huge and difficult to work with. They can be found in their entirety in [43], but to give a feel for their scale, here is the lefthand side of one of the rules, which is too big to fit on the page:

(4)
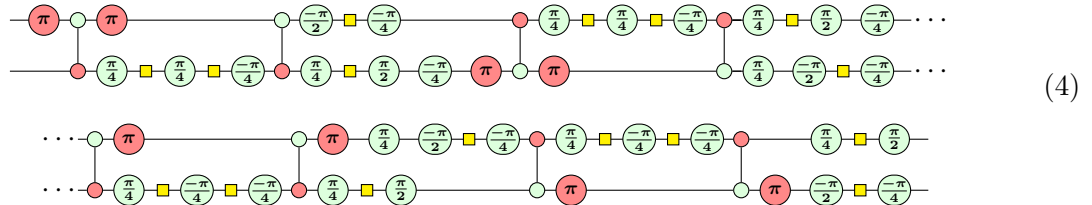
We expect this situation to become worse as we go to more qubits. For example, it is hard to imagine that a 3-qubit rule such as the following:

(5)
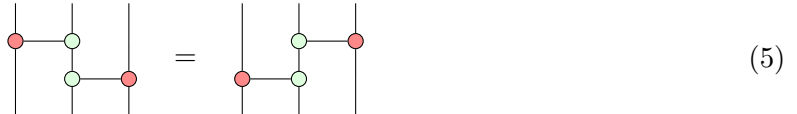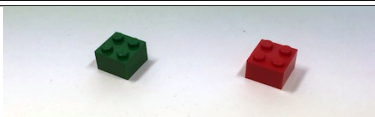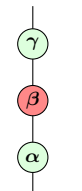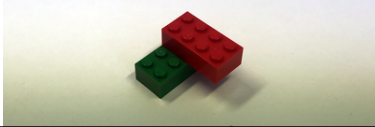
could ever be proven using just the 2-qubit rules from [43, 100], or any 2-qubit rules for that matter. Doing so seems to require decomposing at least one of the CNOT gates into single-qubit gates, which is impossible. Of course, the devil is in the details, so we'll leave the following as a conjecture for now:

Conjecture 2.1. No set of rules involving only two qubit circuits can be complete for circuits with more than 2 qubits.

On the other hand, we'll see in the completeness section 4 that it is possible to fit on one side of A4 all the ZX-rules needed to prove all the equations that are true for for all ZX-pictures, including circuits made from any gates with any number of qubits.

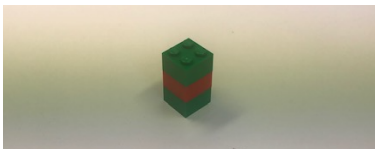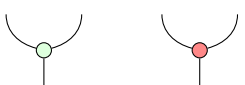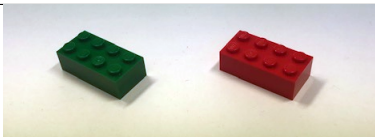These much simpler ZX-rules reflect the fact that the ZX-language is in some way or another more fundamental than circuits.

Consider an analogy using LEGO. The basic LEGO brick has been designed for it's versatility, but if you were crazy enough to glue all of your LEGO together into some fixed 'composite' blocks, that famous versatility goes away. Just for fun, let's take this a bit farther and suppose there were indeed LEGO analogues for ZX-pictures:

ZX-language	LEGO analogue

Standard LEGO allows for a wealth of creations:
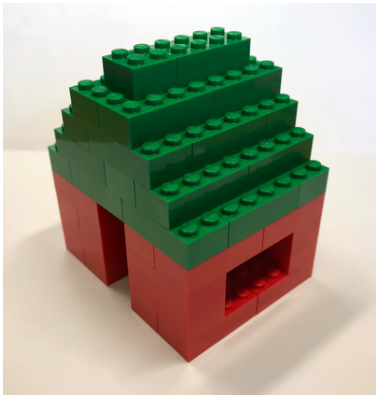
while the composite block only allows for a restricted spectrum of 'art':
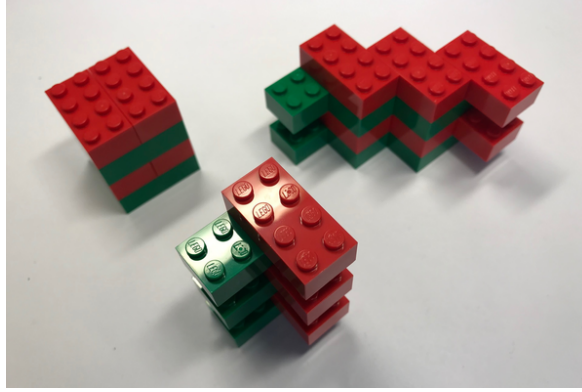
In particular, circuit gates have unitarity imposed upon them, while the ZX-components have been liberated from the unitary constraint.

If we want to actually run a computation on a quantum computer, it could be the case that we only really care about unitary quantum circuits in the end. In that case, it is natural to ask: is this extra freedom actually a *good* thing? We would contend that it is, and that we have a situation that is somewhat analogous to complex analysis. In the case of complex analysis, leaving real numbers behind (sometimes temporarily), gives us much more power and elegance, even when proving things about real numbers. We will see this same phenomenon happening for ZX-pictures in Section 5, where we discuss how to optimise quantum circuits by temporarily leaving the circuit world, then coming back.

It was explained in [104] that the algebraic structures underlying the ZX-calculus are not just normal LEGO, but 'magic LEGO', which are very bendy and enable all sorts of wild creations. This is thanks to the flexibility of the graphical language, which we'll discuss in the next section. By only considering 'glued-together' LEGO, i.e. quantum gates, we miss out on this whole story. So the moral is:

Stop gluing your LEGO together!

3   Basic ZX-rules

Spider fusion rules.   Concretely, there are three kinds of rules governing the ZX spiders (2). The first kind concerns how spiders of the same colour interact, and they are very simple: spiders of the same colour 'fuse' together and their phases add up:

$$\tag{6}$$

One way to think of spiders is as 'multi-wires', in that while ordinary wires have two ends, multi-wires can have multiple ends. The following multi-wires then happen to be ordinary wires:

Now, what characterises a wire is that it connects its two ends, and if you connect two wires together you again get a (now longer) wire. The same is true for multi-wires, and (6) just says that if you connect two multi-wires, then you get another multi-wire.

There also is no real difference between a spider-input-leg and a spider-output-leg, as spider-fusion allows these roles to be easily exchanged:
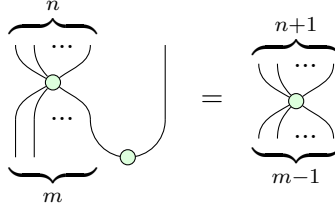
More generally, this implies that in ZX-calculus:

only connectivity matters

and that we can think of ZX-pictures as graphs, that is, something that is specified by nodes and edges connecting these. The loose legs then make it an 'open' graph [47]. This flexibility is something that makes no sense for ordinary circuits, where each gate must have well-defined inputs and outputs.

Strong complementarity rules. The second kind of rules concern the interaction between spiders of different colours. They can either be stated as these two rules:

 (7)
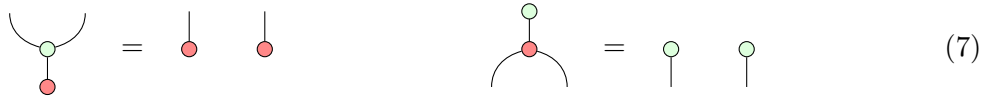
together with this third one:

 (8)

or, as this single rule:

 (9)
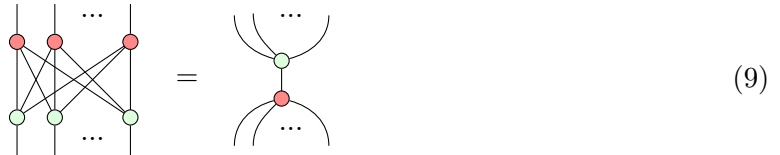
The rules (7) tell us that single leg spiders (a.k.a. states/effects), are copied by a spider of the opposite colour. The rule (8) is slightly harder to interpret, and let's not get us started about (8). But they all follow a clear pattern, namely, the distinct colours can move trough each other. Taking these rules, together with spider-fusion, one can derive this one [30]:

 (10)
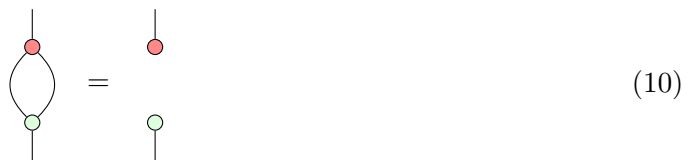
Let's stress again that it is essential to have spider-fusion to derive this rule. Without it (9) and (10) are independent. In fact, in mathematics, rule (9) defines a bialgebra, and having (10) makes it a Hopf algebra (with trivial antipode) [22]. We will say something more about the mathematical familiarity of these specific rules in Sec. 9.

Rule (10) has a very intuitive reading, namely, that two wires between spiders of opposite colour always vanish. In other words, a 2-cycle always vanishes:
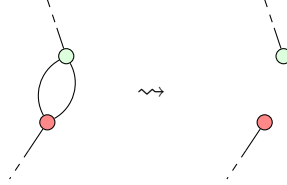
We can also give such an interpretation to (8), namely, that we can also eliminate all 4-cycles:
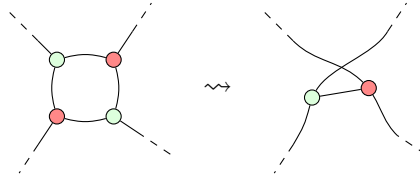
Rule (10) also has a very clear conceptual interpretation, namely, complementarity, or in modern terminology, unbiasedness. One can show that spiders, when defined as linear maps that obey spider-fusion are always uniquely fixed by a choice of orthonormal basis [41]. Then (10) tells us that these two ONBs must be mutually unbiased [30,39]. Mutually unbiased bases crop up all the time in quantum computing and quantum information theory. For example, a lot of quantum cryptography, including the famous BB84 quantum key distribution protocol [11], depends on mutually unbiased bases.

So the rule (10) defines pairs of mutually unbiased ONBs. Because, assuming spider-fusion, the rule (9) is stronger than (10), we call is 'strong complementarity'. A funny thing about this novel notion of strong complementarity is that we actually know more about it then about ordinary complementarity. We know that mutual strong complementarity is monogamous, so it can only come in pairs [39, Thm. 9.66], and all of these pairs have been fully classified for finite dimensional Hilbert spaces, in terms of the finite Abelian groups [32].

In terms of circuits, rule (10) tells us that CNOT-gates are unitary:
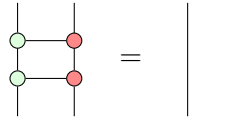
If instead of having the CNOT-gates acting on the same wire with the same colours, we do the opposite, we get a circuit interpretation for (8):
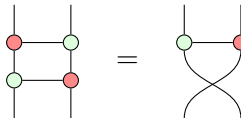
Together these two circuit equations yield:
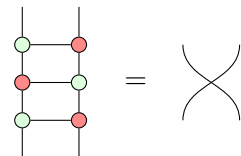
A more extensive discussion of strong complementarity is in [39]. For now we stop discussing rules, and do some stuff with the ones we have. We discuss rules further in the following section.
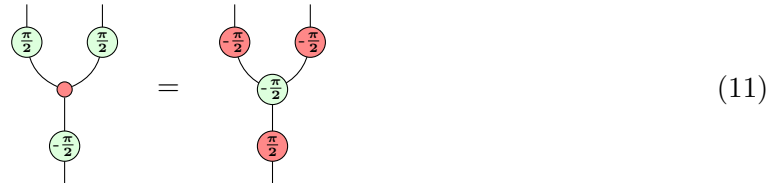
4 A complete calculus

Neither the rules (6) or (9) are specific to qubits, but make sense in all dimensions, and even beyond Hilbert space quantum theory. Indeed, they provide a canvas for studying theories more general than quantum theory, and they have for example enabled a crisp pictorial presentation of Spekkens' toy theory [7, 35, 36]. Notably, this kind of presentation enables one to pinpoint exactly where quantum theory and interesting 'quantum-like' theories depart. In this case, it has to do with the difference in the two finite groups $\mathbb{Z}_4$ and $\mathbb{Z}_2 \times \mathbb{Z}_2$. An extensive discussion of all of this is in [39], Chapter 11.
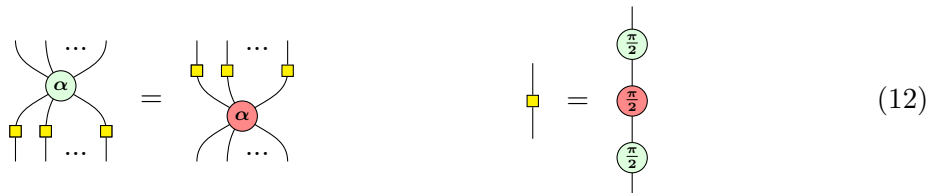
Other papers on generalised theories based on strong complementarity include [32, 33, 59, 60, 63, 64]. All of this is part of the 'process theories' approach to quantum foundations, where quantum-like theories are defined using a symmetric monoidal category, a.k.a. a process theory, and their features are studied abstractly (see e.g. [25, 26, 61, 62, 74, 82, 92, 93, 98, 99]).

However, if we come back down to earth, we can look at which rules actually *are* specific to quantum computation with qubits. As we will see, we don't need to go too far before we have enough rules to prove every true equation between pictures.

Qubit related rule(s). Turning our attention to Hilbert space again, and qubits specifically, another rule that was part of the ZX-calculus early on, although in a very different form, is the following one:

$$\tag{11}$$

The form in which it appeared initially was the 1st one of these rules [29]:

$$\tag{12}$$

which is a pretty one, with the 2nd one added a bit later [50], which is slightly less pretty. Together these two rules involving the yellow box are equivalent to (11). So what is (11) telling us?

We already told you about X spiders and Z spiders, but you might be wondering 'what happened to Y?' Did we put our brains in the oven and cook our Y's?

No! In fact, we didn't define Y-spiders, because they can already be defined in two different ways: in terms of an X-spider or in terms of a Z-spider. Equation (11) relates those two different ways.

This rule comes from the geometry of the *Bloch sphere*, a common way to visualise qubit operations as sphere rotations, in order to rotate X/Z into Y. Alternatively, you can slightly

modify this rule as follows:
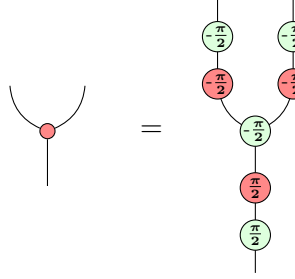
which really is:
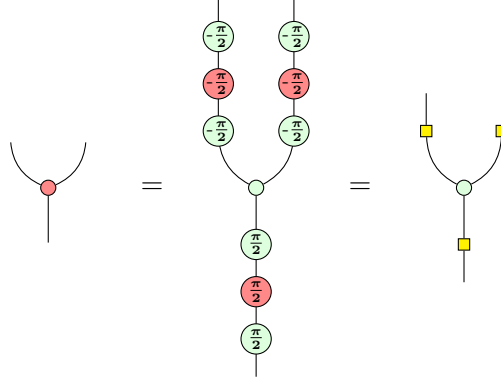
And hence-ish the equivalence with rules (12). See [39] for a proper proof, without the 'ish'. :)

A complete set of rules So what can we prove with the rules we now have? That is:

$$(13)$$

We already pointed out in Section 2 that with ZX-calculus we can go all the way and prove every equation that one can prove using linear algebra. It was shown in shown in [96] that these rules are not enough just yet.

However, Backens [3] showed that they do enable us to prove every equation that holds for *stabilizer quantum theory*, i.e. ZX-pictures with phases restricted to multiples of $\pi/2$.

This is surely not an unimportant fragment of quantum theory, as, for example, it suffices to prove that quantum theory is non-local [32]. On the other hand, stabiliser quantum circuits can be efficiently simulated classically [65].

In practice, even though the ZX-rules above cannot prove *all* equations involving circuits beyond stabiliser quantum theory, they seem pretty capable for many practical tasks such as circuit optimisation, as we'll see in the next section.

Of course, we do really want to understand which extra rules are needed in order to be able to prove all equations. These were established for the first time by Ng and Wang in [87], building

further on Hadzihasanovic's result on a graphical calculus related to the ZX-calculus [67, 68]. Along the way, a result by Jeandel, Perdrix and Vilmart established derivability of all equations for the 'Clifford+T' ZX-pictures, which generalise stabilisers by allowing multiples of $\pi/4$ rather than only $\pi/2$ [71].

Theorems like these are called *completeness* theorems, in the sense that the rules form a complete set with respect to derivability. There are now several different complete sets of rules for the full family of ZX-pictures [87, 109], as well as the various different special cases [71–73, 108]. The most succinct one currently around adds a single rule to the 4 rules above, which allows for exchanging the colours of the phases in triples [109]:

$$
\begin{array}{ccc}
\substack{\gamma \\ \beta \\ \alpha} & = & \substack{\tilde{\gamma} \\ \tilde{\beta} \\ \tilde{\alpha}}
\end{array}
\tag{14}
$$

where each of the phases $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$ are trigonometric functions of the phases on the left-hand side.

This rule was first introduced for the case of two-qubit circuits [43], with two of the authors of the present paper failing to realise that it would yield full-blown completeness as well. This seems to show us that the four basic rules (13) already capture all of the complex interactions of multiple qubits, up to some 'local' single qubit equations, which are all subsumed by (14).

So, if we have a complete set of rules for all ZX-pictures, we should be happy right? Wrong! Completeness should be seen as the beginning and not the end for the ZX-calculus, and there is much to be gained by finding better rules.

For example, the succinctness obtained from the introduction of the colour-exchanging rule (14) comes at the price of introducing complicated, trigonometric functions of phases whenever it is applied. In fact, these are ugly enough that we didn't even bother to write them here. If we are working with phases numerically on a computer, this isn't a big problem, but for symbolic manipulation this quickly becomes impractical.

One way around this problem is to shift to the *algebraic ZX-calculus*, which replaces the phases $\alpha \in [0, 2\pi)$ – which become $e^{i\alpha}$ in the definition of a spider (2) – with plain ol' complex numbers $a \in \mathbb{C}$:

$$
\substack{\cdots \\ \alpha \\ \cdots} \quad \rightsquigarrow \quad \substack{\cdots \\ a \\ \cdots}
$$

Our previous notion of spiders are still around, just by setting $a := e^{i\alpha}$, but the extra generality buys us several nice features such as a more direct encoding of complex-valued matrices as well as straightforward generalisations from 2D to all finite dimensions [111] and from complex numbers to any commutative semi-ring [110].

5 Automated circuit optimisation

If a circuit is given, can ZX-calculus help with simplifying it? Of course it can, and it seems to be better at it than anything else. Here's an example of how that works. Suppose we want to simplify the following circuit made up of multiple gates, and we need to measure the last two

quibits:

qubits prepared in $|0\rangle$
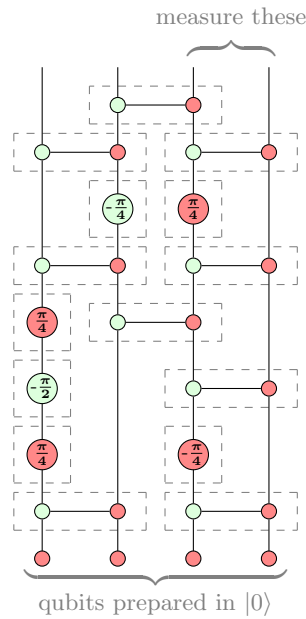
There are a lot of 4-cycles here, and we've just learned that ZX-calculus is good at getting rid of 4-cycles. The 4-cycles are here:
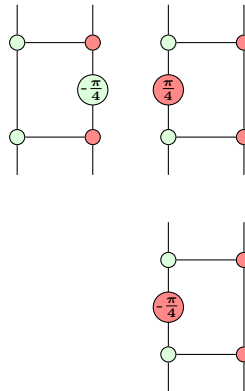
However, they are not 4-cycles because they happen to look like rectangles, as the 4-cycles we are looking for has alternating colours as corners. We can do some (un-)fusing:
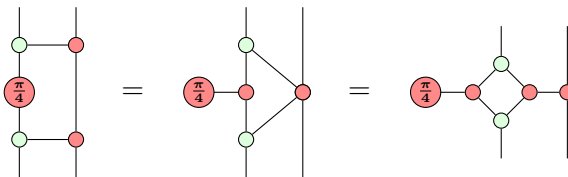
and now we can eliminate that square, and then re-arrange a bit:
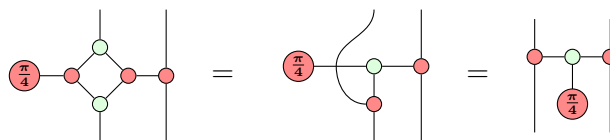
We can do the same for the other 4-cycles:
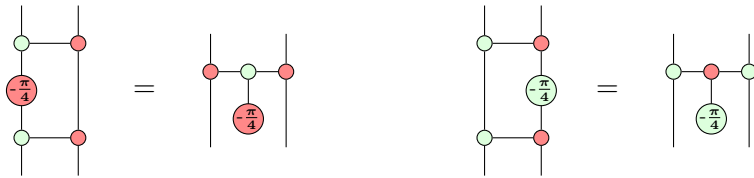
What we get has been called a 'phase gadget' [77]. By using the trick for eliminating 4-cycles again, one also finds that phase gadgets with opposite angles cancel out:
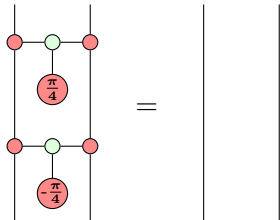
Hey ho let's go. We first bring in phase gadgets and then fuse:
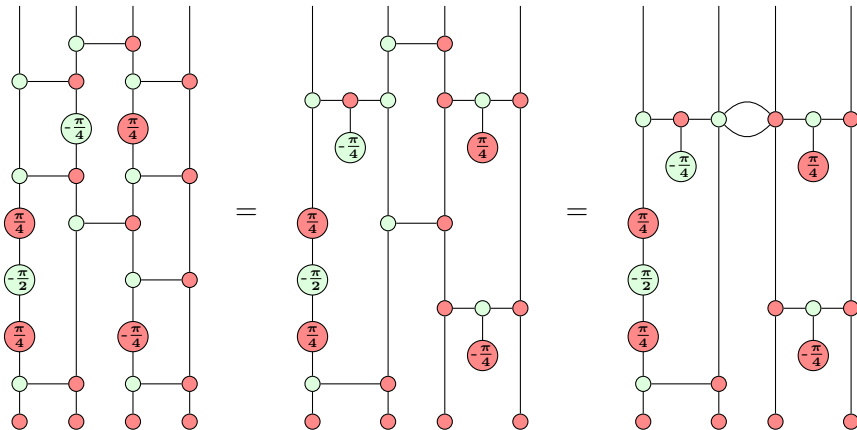
We get a 2-cycle which as we know vanishes, and then the two qubits on the left completely disentangle from those on the right, so we can forget about them:
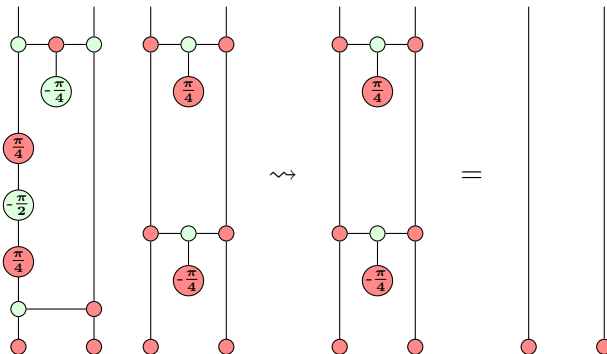
resulting in the fact that we end up with what we started with, despite the whole circuit looking pretty complicated when we started.

While it is easy to work on small circuits by hand, we would also like to apply these techniques to circuits with thousands or millions of quantum gates, so it is natural to consider

```
In [11]: zx.simplify.clifford_simp(g)
         g.normalise()
         zx.d3.draw(g)
```
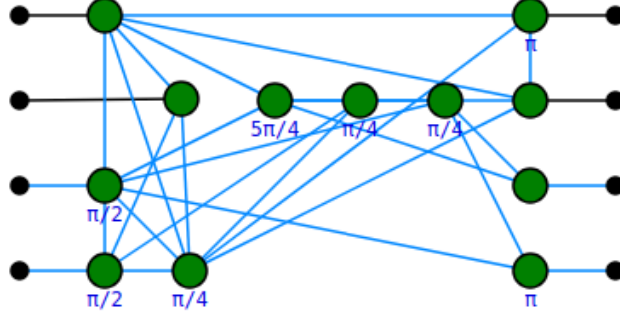

Figure 1: PyZX is a Python library and circuit optimsation tool using the ZX-calculus. See github.com/Quantomatic/pyzx.

how these kinds of simplifications can be automated. A standard method for this is to replace equations, which can be applied in either direction, with directed rewrite rules. For example:

$$
\vcenter{\hbox{}} = \vcenter{\hbox{}} \qquad\qquad \rightsquigarrow \qquad\qquad \vcenter{\hbox{}} \rightarrow \vcenter{\hbox{}} \tag{15}
$$

As long as the rules decrease some metric of the ZX-picture (e.g. the number of spiders), applying them blindly until they don't apply any more will always terminate. In rewrite theory lingo, this means we can automate simplification of ZX-pictures by using a *terminating rewrite system*, based on a subset of the ZX-calculus rules.

This rewriting can be formalised in such a way that ZX-pictures can be represented and transformed by software tools using a method called *double-pushout graph rewriting* [54]. The basic theory for representing ZX-pictures as graphs and rewriting them was presented in [47], and recently extended in [14]. This forms the basis of a diagrammatic 'proof assistant' called Quantomatic [78].

By 'breaking open' the gates in a quantum circuit, we can find simplifications in the ZX-calculus that would be hidden at the gate level. However, we may end up with something that doesn't look at all like a circuit any more. Hence, an important problem for ZX-based optimisation techniques is *circuit extraction*, that is efficiently recovering a gate-decomposition from a simplified ZX-picture. This simplify-and-extract technique for ZX was introduced in [49], generalised to a broader family of diagrams in [6], and forms the basis of the quantum circuit optimisation tool PyZX [76] (Fig. 1).

ZX-picture rewriting also forms the basis of a special-purpose circuit simplification tool STOMP [44], which reduces an important cost metric called the T-count of a quantum circuit using so-called 'spider-nest' identities.

6 Quantum Natural Language Processing

ZX-calculus grew out of a more general pictorial approach to quantum foundations and quantum computation, called categorical quantum mechanics (CQM) [1, 23]. In fact, what CQM does is propose an alternative formalism to Hilbert space, which puts the emphasis on how systems compose, rather than in which space systems are described. Thanks to the successes of ZX-calculus it is fair to say the this alternative has genuine practical advantages.

On the other hand, the graphical structures employed by CQM (and in many cases originating there) stretch well beyond quantum theory. For example they have been applied in computability theory [89], models of concurrency [105], control theory [9, 19], the study of electrical [10] and digital [57] circuits, game theory [56], broader cognitive features [13], natural language processing [42, 95], and even consciousness research [102, 106]!

As aspects of ZX-calculus are essential to some of these areas, one may argue that to some extent they are 'quantum-like'. While this may only be taken as a rough analogy in some cases, in the particular area of natural language processing (NLP), it seems to be useful to take this quantum connection seriously. In the approach to NLP put forward in [42], vector space models for word meaning were combined with grammatical structure to produce compositional models of sentence meaning. As this model makes crucial use of this tensor product of vector spaces, which gives exponential space requirements on a classical computer. On the other hand, forming tensor products on a quantum computer is cheap, as this is just what happens when you put two pieces of quantum data next to each other. This realisation led to the proposal of a quantum algorithm for natural language processing [113]. For various reasons, this first proposal was not very practical to run on quantum computers of today or the near future.

More recently, this proposal was adjusted and refined in order to fit on currently existing quantum hardware [27, 85], and implemented on IBM's quantum devices [86]. This provided an example of a 'quantum native' solution to a classical problem. That is, while the problem has nothing to do with quantum systems, it's structure still naturally lives on a quantum computer.

An important refinement from the original algorithm to the one recently implemented on a real quantum computer was the use of the ZX-calculus to turn a picture representing a natural language sentence into a runnable quantum circuit that computes something about the sentence's meaning within the NLP model. Here is an example of a sentence and it's interpretation as a picture:
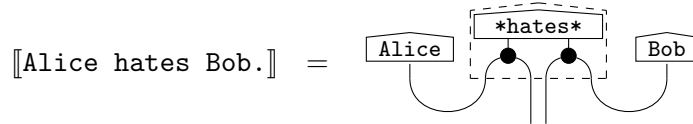
To 'run' this sentence on a quantum computer, we first interpret the black dot as the green ZX-spider. We can now use ZX-calculus to turn it into circuit-form:
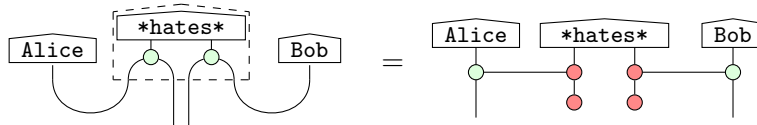
We may then use ZX-calculus rules to massage this diagram into a different shape (who's

meaning is equivalent):

and replace the word-meanings by some pieces of ZX-picture with free parameters, $\alpha, \beta, ...$:
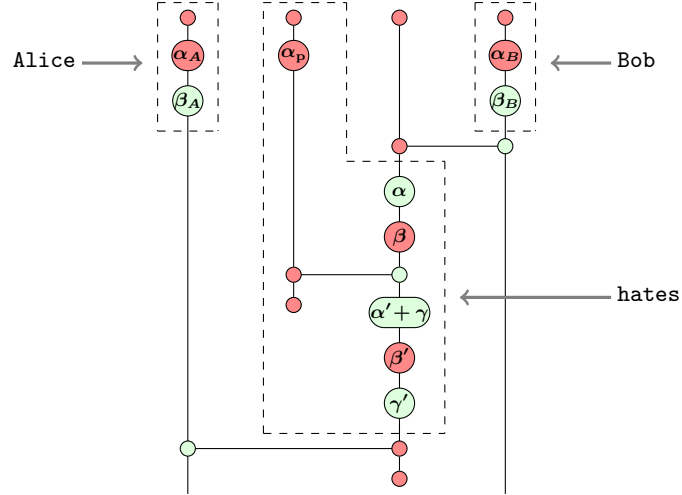
These parameters are 'trained' over the course of many runs of such circuits using machine learning techniques. The finished product is a quantum circuit capable in principle of comparing sentence meanings, answering questions, and doing many more linguistic tasks.

This very simple sentence only uses a dash of ZX-calculus, but it already becomes clear that the 'elasticity' of ZX is helpful for such tasks. There are many equivalent ways to compute the sentence meaning, and some fit better on a quantum computer than others, hence the 'massaging' in equation (16). This really starts to pay off when one starts to consider more complex sentences like this one:
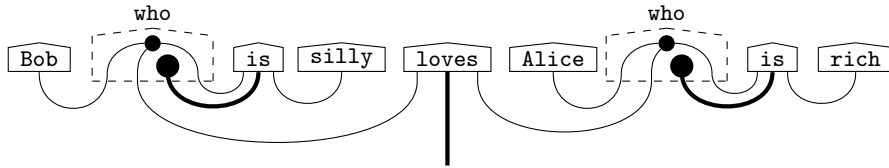
This can be seen as a compilation process, but one that doesn't take a program language as input, but natural language, and turns it into quantum machine code using the ZX-calculus to handle everything in between. The end result is a physics-first: the use of quantum systems to process natural language, with the help of the ZX-calculus.

Quantum machine learning plays a central role in quantum natural language processing. Recently, the ZX-calculus has started to play a role in enhancing our understanding of quantum machine learning itself: first in picturing quantum ansätze [112], and then in analysing important problems within the approach like the barren plateu phenomenon [114].

7 MBQC and Fault-tolerance

Measurement-based quantum computing (MBQC) is an alternative model of computation to the circuit model, where measurements, rather than quantum gates, are the main things driving the computation. The most well-studied MBQC setup is called the quantum *one-way model* [94] In this setup, many qubits are prepared in a certain fixed state, called a *graph state*, then single-qubit measurements are prepared in a particular order.

Notably, the choice of the kind of measurement performed can depend on past measurement outcomes, a principle referred to as *feed-forward*. Even though each individual measurement outcome is non-deterministic, a clever application of feed-forward can produce deterministic quantum computations.

For example, in the one-way model, measurements are defined by angles $\alpha \in [0, 2\pi)$. When they are performed, one of two things happens, non-deterministically:

$$ \textcircled{\alpha} \quad \text{or} \quad \boxed{\alpha+\pi}. $$

Suppose we actually wanted the first outcome for our computation, then the ZX-calculus tells us how to 'push' the unwanted π forward in time, changing future measurement angles:
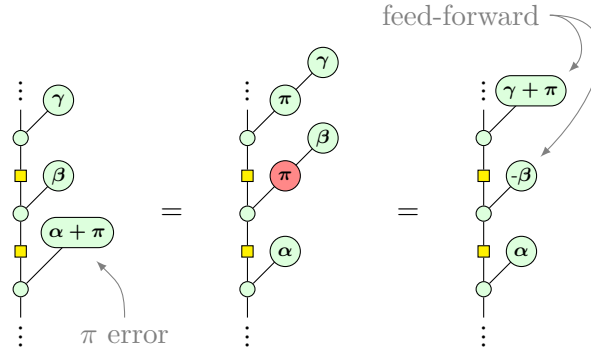
In fact, making these kinds of computations in the one-way model easier was one of the original motivations for the ZX-calculus. ZX was used, for example, to teach the one-way model in a fully-graphical way [39], give the first technique for translating MBQC computations into circuits that didn't require extra qubits or (non-physical) feedback loops [51], and produce an alternative model for MBQC based on Pauli-ZZ interactions [75], which are the native 2-qubit gate for most types of quantum hardware.

Another popular family of measurement-based models of quantum computation are various forms of fault-tolerant computations based on the *surface code*, a type of quantum error correcting code. Quantum error correction, and fault-tolerance is a huge subject, and way too huge to cover here. However, the basic idea is that many low-level 'physical' qubits correspond to a few 'logical' qubits. When doing computation in this way, it is useful to abstract away individual operations on the physical qubit to and certain high-level logical transformations. A particularly nice instance of this is *lattice surgery* [70], which was co-developed by one of the authors of this survey. In lattice surgery, the main logical operations are 'Z-split', 'X-split', 'Z-merge', and 'X-merge'. You might notice that I just said 'ZX' twice, so maybe this is a job for the ZX-calculus!

Indeed, in [46], the authors showed that ZX is a natural language for lattice surgery computations. For one thing, the basic operations are exactly what they sound like:

split! ← Z   ← Z   split! ← X   ← X

merge!   merge!
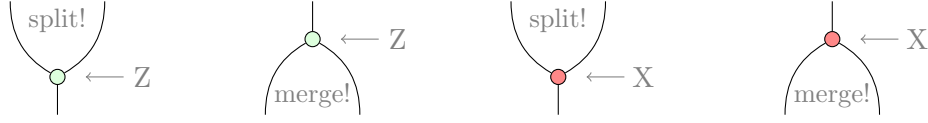
Since these are just spiders, we already know how to use lattice surgery operations to build, for example, a CNOT gate:

$$= \qquad\qquad (17)$$

While splits can be done deterministically, merges might introduce a π error. However, much like in the one-way model, these errors can often be fed-forward using ZX-rules and accounted for by later operations:

π error   feed-forward

$$\pi \qquad = \qquad \pi \qquad = \qquad \pi$$
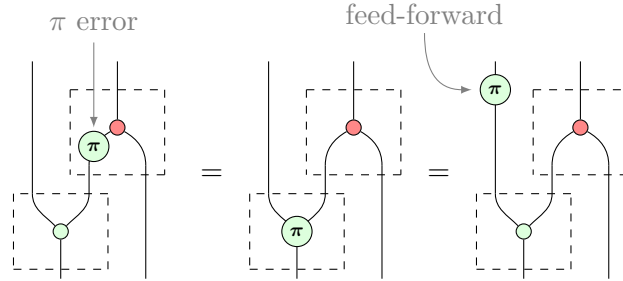
This ZX language for lattice surgery was given a formal foundation in [45] and variations have been used by groups at Google [58] and NII Toyko [69] for optimising various aspects of fault-tolerant computations.

While originally envisioned as a model based on the new primitives of split and merge, subsequent work has focused mainly on using lattice surgery as a tool for building CNOT gates as in equation (17) (with a few notable exceptions, e.g. [83]). Interestingly, in 2020 we saw the first experimental demonstration of logical qubit entanglement using lattice surgery [55, *Nature*], where the authors noted that it was much more efficient to use the primitive split and merge operations to prepare an entangled state. They did it like this:

$$= \qquad =$$
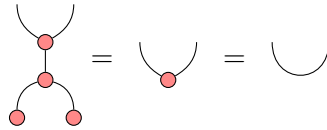
8   Kindergarten quantum mechanics: the experiment

In the abstract, we claimed that this paper is a spiritual child of the 2005 lecture notes *Kindergarten Quantum Mechanics* [23], but in fact, it is rather a spiritual grandchild. The middle generation was a paper called *Quantum Picturalism* [24], which contained among other things a vague proposal for testing the effectiveness of the pictorial formalism. It was claimed that, given the proper learning materials, high-school students could outperform their teachers in quantum theory, if the students used the pictorial formalism while the teachers used the Hilbert space formalism.

Now, ten years later, we have the materials in place for a far more ambitious goal: getting high-school students to do state-of-the-art quantum computing, on par with the abilities of Oxford post-graduate students. First of all, this required a book specifically targeted at high-schoolers, and a set of tasks to set both the high-schoolers and the postgrads, and some other interesting groups (like art students!). The book [37] and the tasks are written, but still under-wraps until the experiment is done. Without giving too much away, this should give some idea of the tone of the book:
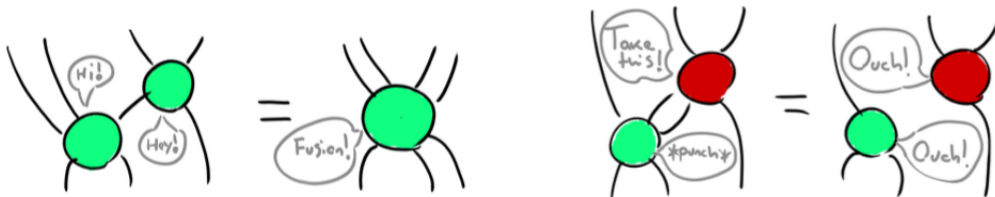
The experiments have already begun. Watch this space!

9 How we got here: a brief history of the ZX-calculus

Conception. ZX-calculus was 'born' in a rejected conference abstract [28] (QIP 2007), written in the mountains north of Tehran. The referee reports said things like:

'Looks cute, so what?'

The basic idea at the time was to expand categorical quantum mechanics to complementary quantum observables, with the now-stated aim to make it directly applicable to practical quantum computing, but the deeper goal was to do something the program of Birkhoff-von Neumann quantum logic [12] failed to do: produce from first principles a full-fledged alternative to Hilbert space quantum theory. Strong complementarity was reverse-engineered by looking at 'generalised flow' for MBQC [51], and phases just followed from general abstract nonsense, a.k.a. category theory.

ZX-calculus was 'officially' introduced in the accepted conference paper [29, ICALP, 2008]. A slightly unfortunate statement in [29] concerns the relative status of complementarity (10) and strong complementarity (9): it was shown (in Theorem 3) that under a 'mild assumption' these are equivalent. Later, in the 85 page corrected and substantially expanded journal version [30, NJP, 2011] that 'mild assumption' was in Thm. 9.24 shown to be essentially equivalent to strong complementarity. A proper treatment of the (huge!) difference between complementarity and strong complementarity appeared in [32], by establishing a connection with non-locality, and fully classifying strongly complementary bases. (The full classification of complementarity bases is still completely open, and has swallowed several careers whole.)

Early rule fuzz. One of the early goals of the ZX-calculus was to fully understand MBQC using pictures. In doing so, it quickly become clear that the Euler decomposition rule on the right of equation (12) was needed in addition to rules that were already established [50]. This than settled the core of ZX-calculus, as it still is now.

After that, we attempted to move ZX-calculus beyond bog-standard quantum gates and MBQC to describe W-states. In quantum entanglement theory, there is 'essentially' only one two-qubit entangled state, up to equivalence by so-called *stochastic local operations*, but for 3

qubits there are two [53]. One is called a GHZ-state, and is just a 3-legged spider, and the other is called a W-state.

At first, a lot of time and energy was spent trying to cram W-states into the ZX-calculus. Along the way we got a useful new ZX-rule (*supplementarity* [34]), but we didn't get much closer to being able to work with W-states. This early defeat made some of us consider an alternative to the ZX calculus which is now called the...

ZW-calculus. The completeness of ZX-calculus was initially proven using completeness of another calculus: the ZW-calculus, a.k.a. the GHZ/W calculus [38]. The key idea was to slightly vary the rules governing spiders as follows:
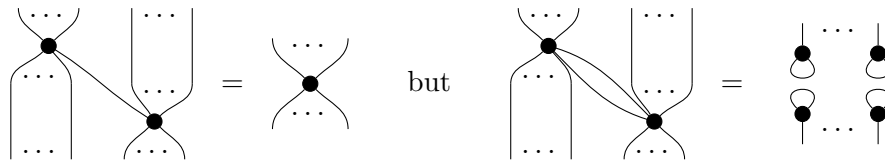
These spiders were called W-spiders, as the W-state was an instance of them. While this seemed like a relatively minor tweak to the notion of spiders, it turned out that, unlike the ZX-calculus, it was relatively straightforward to find a complete set of rules [66], owing in part to the fact that the ZW-calculus it more directly encodes the rules of arithmetic [40].

The first completeness theorems for the ZX-calculus were proven using a somewhat roundabout technique that encoded ZX-pictures as ZW-pictures and showed (painstakingly) that each of the ZW-rules was derivable in ZX. This proved an important step in the progress of ZX theory, but the original *raison d'etre* for ZW remains open:

Open problem. 9.1. Provide a classification of many-qubit entanglement (which is still poorly understood beyond three qubits) using the ZW-calculus.

A dead end: the 'XYZ-calculus'. An early variation on ZX-calculus was the trichromatic calculus of [81], where a third colour (i.e. the Y-observable) was added. As one cannot have the cups for all three observable coincide, for the sake of symmetry, none did. This resulted in a substantially more complex rule-set and the calculus was never really used. The reason it shouldn't be used, probably, is because of monogamy of strong complementarity [39]. That is, at most two colours of spiders can satisfy the strong complementarity rules described in section 4 with each other, so to accommodate more colours, you have to put some sort of 'awkward twist'.

(In)completeness and presentations. The rules of ZX-calculus as firstly introduced in [30] without much consideration for scalar factors. These tended to be ignored when it was convenient, which causes problems e.g. for computing probabilities of quantum measurement outcomes. Scalars were seriously considered in the rules for the stabilizer fragment of ZX-calculus [5]. Minimality (whether a rule is non-derivable from other rules) of ZX rules was initially considered in [8] for stabilizer ZX-calculus, then it was further investigated for Clifford+T ZX-calculus in [101].

As mentioned in section 4, the first breakthrough for completeness of ZX-calculus was made by Backens [3] for the stabilizer fragment. The completeness of the real stabilizer ZX-calculus then followed in [52]. Furthermore, Backens proved that the scalar version of stabilizer ZX-calculus and the single-qubit Clifford+T fragment of ZX are complete [4, 5]. At the same

time, Schröder de Witt and Zamdzhiev showed by a counter-example that ZX-calculus can't be universally complete if it is just equipped with stabiliser-style rules [97]. They also conjectured that completeness could be achieved by adding a rule of form (14). Later on, Perdrix and Wang proved that the stabiliser-style ZX even can't be complete for the multi-qubit Clifford+T fragment, and the supplementarity rule is necessary [91].

At some point, some people (including at least one of the authors of this paper) started to believe there would be *no* finite set of rules which would be complete for any substantial extension of stabiliser quantum theory.

Fortunately, the aforementioned author didn't put money on it, as 2017-18 saw a veritable frenzy of completeness results for ZX. First Jeandel, Perdrix, and Vilmart (a.k.a. 'team Nancy') proved the completeness of multi-qubit Clifford+T ZX-calculus with a translation from ZW-calculus [71]. Very soon after, Ng and Wang finished the first complete axiomatisation for universal qubit ZX-calculus using a similar approach to the Nancy team and by introducing some new generators to the theory [87]. They were also able to give a (different) complete axiomatisation for the multi-qubit Clifford+T fragment [68, 88]. Inspired by Ng and Wang's results, the Nancy team gave another complete axiomatisation for universal qubit ZX-calculus in terms of original ZX spiders [72]. Furthermore, they proposed a normal form for ZX diagrams based on which universal completeness was still obtained without any resort to translation from ZW-calculus [73]. At last, as we mentioned in section 4, Vilmart successfully proved Schröder de Witt and Zamdzhiev's conjecture with the explicit expression of (14) [109].

Precursors and successors. The kind of pictorial reasoning used in this paper was initiated by Penrose as a more intuitive alternative for ordinary tensor notation [90]. In fact, even though Penrose had reportedly been using the notation since he was an undergrad, he didn't think too highly of its prospects, mainly due to typesetting issues. In his 1984 text *Spinors and Spacetime*, he notes:

> The notation has been found very useful in practice as it greatly simplifies the appearance of complicated tensor or spinor equations, the various interrelations expressed being discernible at a glance. Unfortunately the notation seems to be of value mainly for private calculations because it cannot be printed in the normal way.

Of course a lot can change in 20 years. In 2004, this notation was adopted to the specific needs of (finite-dimensional) quantum theory in CQM [1], which started the compositional axiomatization of quantum theory.

Spiders, in their algebraic incarnation as certain Frobenius algebras, first appeared in the category-theory literature [21, 79]. Hopf algebras, which in ZX-calculus terms correspond to the strong complementarity rules in absence of the spider-rules, have been around in their current concrete form since 1956 [22], when Cartier generalised earlier definitions based on structure theorems on the cohomology of compact Lie groups by Hopf, Samelson, Borel and others in the 1940s. Hopf algebras and their representations are now studied extensively under of moniker of *quantum group theory* (see e.g. [84]).

The idea of depicted (classical boolean) circuits as pictures of more basic components, and the pictorial depiction of the Hopf algebra (a.k.a. strong complementarity) laws, goes back to Lafont [80]. However, to capture the full richness of quantum circuits, one needs not just a single Hopf algebra, but a pair of them which interact in a special way (namely via the Frobenius,

a.k.a. spider fusion laws). This structure was first made explicit, to the best of our knowledge, as part of the ZX-calculus.

Notably, this structure contains non-trivial algebraic parts (i.e. those with operations taking many inputs) and non-trivial *co-algebraic* parts (i.e. those with operations producing multiple outputs), which interact with each other. This novel mathematical structure is interesting in its own right, and has since been studied using category theory [16,18,20,48] and found a multitude of applications e.g. in the study of signal-flow graphs [17] and concurrent systems [15].

The future. New papers on ZX-calculus are appearing at a steadily increasing rate, and we can only expect that increase to continue. There is a regularly updated list of papers on ZX-calculus available here that you may want to consult in the future:

https://zxcalculus.com/publications.html

References

[1] S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 415–425, 2004. arXiv:quant-ph/0402130.

[2] M. Amy, J. Chen, and N. J. Ross. A finite presentation of cnot-dihedral operators. In Bob Coecke and Aleks Kissinger, editors, Proceedings 14th International Conference on *Quantum Physics and Logic,* Nijmegen, The Netherlands, 3-7 July 2017, volume 266 of *Electronic Proceedings in Theoretical Computer Science*, pages 84–97. Open Publishing Association, 2018.

[3] M. Backens. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*, 16:093021, 2014. arXiv:1307.7025.

[4] M. Backens. The ZX-calculus is complete for the single-qubit Clifford+T group. In B. Coecke, I. Hasuo, and P. Panangaden, editors, *Proceedings of the 11th workshop on Quantum Physics and Logic*, volume 172 of *Electronic Proceedings in Theoretical Computer Science*, pages 293–303. Open Publishing Association, 2014.

[5] M. Backens. Making the stabilizer ZX-calculus complete for scalars. *Electronic Proceedings in Theoretical Computer Science*, 195:17–32, November 2015.

[6] M. Backens, H. Miller-Bakewell, G. de Felice, and J. van de Wetering. There and back again: A circuit extraction tale. *arXiv preprint arXiv:2003.01664*, 2020.

[7] M. Backens and A. Nabi Duman. A complete graphical calculus for Spekkens' toy bit theory. *Foundations of Physics*, 2015. arXiv:1411.1618.

[8] M. Backens, S. Perdrix, and Q. Wang. Towards a Minimal Stabilizer ZX-calculus. *Logical Methods in Computer Science*, Volume 16, Issue 4, December 2020. arXiv:1709.08903.

[9] J. C. Baez and J. Erbele. Categories in control, 2014. arXiv:1405.6881.

[10] J. C. Baez and B. Fong. A compositional framework for passive linear networks. *arXiv:1504.05625*, 2015.

[11] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179. IEEE, 1984.

[12] G. Birkhoff and J. von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37:823–843, 1936.

[13] J. Bolt, B. Coecke, F. Genovese, M. Lewis, D. Marsden, and R. Piedeleu. Interacting conceptual spaces I. In M. Kaipainen, A. Hautamäki, P. Gärdenfors, and F. Zenker, editors, *Concepts and their Applications*, Synthese Library, Studies in Epistemology, Logic, Methodology, and Philosophy of Science. Springer, 2018. to appear.

[14] F. Bonchi, F. Gadducci, A. Kissinger, P. Sobocinski, and F. Zanasi. String diagram rewrite theory i: Rewriting with frobenius structure. *arXiv preprint arXiv:2012.01847*, 2020.

[15] F. Bonchi, J. Holland, R. Piedeleu, P. Sobociński, and F. Zanasi. Diagrammatic algebra: from linear to concurrent systems. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–28, 2019.

[16] F. Bonchi, R. Piedeleu, P. Sobociński, and F. Zanasi. Graphical affine algebra. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2019.

[17] F. Bonchi, R. Piedeleu, P. Sobociński, and F. Zanasi. Contextual equivalence for signal flow graphs. In Jean Goubault-Larrecq and Barbara König, editors, *Foundations of Software Science and Computation Structures*, pages 77–96, Cham, 2020. Springer International Publishing.

[18] F. Bonchi, P. Sobocinski, and F. Zanasi. Interacting bialgebras are Frobenius. In *17th International Conference on Foundations of Software Science and Computation Structures (FOSSACS)*, pages 351–365, 2014.

[19] F Bonchi, P. Sobociński, and F. Zanasi. Full abstraction for signal flow graphs. In *Principles of Programming Languages, POPL'15.*, 2015.

[20] F. Bonchi, P. Sobociński, and F. Zanasi. Interacting hopf algebras. *Journal of Pure and Applied Algebra*, 221(1):144–184, 2017.

[21] A. Carboni and R. F. C. Walters. Cartesian bicategories I. *Journal of Pure and Applied Algebra*, 49:11–32, 1987.

[22] P. Cartier. A primer of Hopf algebras. In *Frontiers in number theory, physics, and geometry II*, pages 537–615. Springer, 2007.

[23] B. Coecke. Kindergarten quantum mechanics. In A. Khrennikov, editor, *Quantum Theory: Reconsiderations of the Foundations III*, pages 81–98. AIP Press, 2005. arXiv:quant-ph/0510032.

[24] B. Coecke. Quantum picturalism. *Contemporary Physics*, 51:59–83, 2009. arXiv:0908.1787.

[25] B. Coecke. A universe of processes and some of its guises. In H. Halvorson, editor, *Deep Beauty: Understanding the Quantum World through Mathematical Innovation*, pages 129–186. Cambridge University Press, 2011. arXiv:1009.3786.

[26] B. Coecke. Terminality implies no-signalling... and much more than that. *New Generation Computing*, 34:69–85, 2016.

[27] B. Coecke, G. de Felice, K. Meichanetzidis, and A. Toumi. Foundations for near-term quantum natural language processing. *arXiv preprint arXiv:2012.03755*, 2020.

[28] B. Coecke and R. Duncan. A graphical calculus for quantum observables. *zxcalculus.com/publications.html*, 2007.

[29] B. Coecke and R. Duncan. Interacting quantum observables. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science, 2008.

[30] B. Coecke and R. Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13:043016, 2011. arXiv:quant-ph/09064725.

[31] B. Coecke and R. Duncan. Tutorial: Graphical calculus for quantum circuits. In *International Workshop on Reversible Computation*, pages 1–13. Springer, 2012.

[32] B. Coecke, R. Duncan, A. Kissinger, and Q. Wang. Strong complementarity and non-locality in categorical quantum mechanics. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science (LICS)*, 2012. arXiv:1203.4988.

[33] B. Coecke, R. Duncan, A. Kissinger, and Q. Wang. Generalised compositional theories and diagrammatic reasoning. In G. Chiribella and R. W. Spekkens, editors, *Quantum Theory: Informational Foundations and Foils*, Fundamental Theories of Physics. Springer, 2016. arXiv:1203.4988.

[34] B. Coecke and B. Edwards. Three qubit entanglement within graphical Z/X-calculus. *Electronic Proceedings in Theoretical Computer Science*, 52:22–33, 2010.

[35] B. Coecke and B. Edwards. Toy quantum categories. *Electronic Notes in Theoretical Computer Science*, 270(1):29 – 40, 2011. arXiv:0808.1037.

[36] B. Coecke, B. Edwards, and R. W. Spekkens. Phase groups and the origin of non-locality for qubits. *Electronic Notes in Theoretical Computer Science*, 270(2):15–36, 2011. arXiv:1003.5005.

[37] B. Coecke and S. Gogioso. Quantum theory in pictures, 2020. Tutorial.

[38] B. Coecke and A. Kissinger. The compositional structure of multipartite quantum entanglement. In *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 297–308. Springer, 2010. arXiv:1002.2540.

[39] B. Coecke and A. Kissinger. *Picturing Quantum Processes. A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017.

[40] B. Coecke, A. Kissinger, A. Merry, and S. Roy. The GHZ/W-calculus contains rational arithmetic. *Electronic Proceedings in Theoretical Computer Science*, 52:34–48, 2010.

[41] B. Coecke, D. Pavlović, and J. Vicary. A new description of orthogonal bases. *Mathematical Structures in Computer Science, to appear*, 23:555–567, 2013. arXiv:quant-ph/0810.1037.

[42] B. Coecke, M. Sadrzadeh, and S. Clark. Mathematical foundations for a compositional distributional model of meaning. In J. van Benthem, M. Moortgat, and W. Buszkowski, editors, *A Festschrift for Jim Lambek*, volume 36 of *Linguistic Analysis*, pages 345–384. 2010. arxiv:1003.4394.

[43] B. Coecke and Q. Wang. ZX-rules for 2-qubit Clifford+T quantum circuits. In Jarkko Kari and Irek Ulidowski, editors, *Reversible Computation - 10th International Conference, RC 2018, Leicester, UK, September 12-14, 2018, Proceedings*, volume 11106 of *Lecture Notes in Computer Science*, pages 144–161. Springer, 2018.

[44] N. de Beaudrap, X. Bian, and Q. Wang. Fast and effective techniques for t-count reduction via spider nest identities. In *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. arXiv:2004.05164.

[45] N. de Beaudrap, R. Duncan, D. Horsman, and S. Perdrix. Pauli fusion: a computational model to realise quantum transformations from ZX terms. In Bob Coecke and Matthew Leifer, editors, Proceedings 16th International Conference on *Quantum Physics and Logic*, Chapman University, Orange, CA, USA., 10-14 June 2019, volume 318 of *Electronic Proceedings in Theoretical Computer Science*, pages 85–105. Open Publishing Association, 2020.

[46] N. de Beaudrap and D. Horsman. The ZX calculus is a language for surface code lattice surgery. *Quantum*, 4:218, January 2020.

[47] L. Dixon and A. Kissinger. Open-graphs and monoidal theories. *Mathematical Structures in Computer Science*, 23(02):308–359, 2013.

[48] R. Duncan and K. Dunne. Interacting frobenius algebras are hopf. In *Proceedings of the 31st Annual IEEE Symposium on Logic in Computer Science (LICS)*, 2016. arXiv:1601.04964.

[49] R. Duncan, A. Kissinger, S. Perdrix, and J. Van De Wetering. Graph-theoretic simplification of quantum circuits with the ZX-calculus. *Quantum*, 4:279, 2020.

[50] R. Duncan and S. Perdrix. Graph states and the necessity of euler decomposition. In *Conference on Computability in Europe*, pages 167–177. Springer, 2009.

[51] R. Duncan and S. Perdrix. Rewriting measurement-based quantum computations with generalised flow. In *Proceedings of ICALP*, Lecture Notes in Computer Science, pages 285–296. Springer, 2010.

[52] R. Duncan and S. Perdrix. Pivoting makes the ZX-calculus complete for real stabilizers. In *Proceedings of the 10th International Workshop on Quantum Physics and Logic*, 2013. arXiv:1307.7048.

[53] W. Dür, G. Vidal, and J. I. Cirac. Three qubits can be entangled in two inequivalent ways. *Physical Review A*, 62(062314), 2000.

[54] H. Ehrig, M. Pfender, and H. J. Schneider. Graph-grammars: An algebraic approach. In *14th Annual Symposium on Switching and Automata Theory (swat 1973)*, pages 167–180. IEEE, 1973.

[55] A. Erhard, H. Poulsen Nautrup, M. Meth, L. Postler, R. Stricker, M. Stadler, V. Negnevitsky, M. Ringbauer, P. Schindler, H. J. Briegel, R. Blatt, N. Friis, and T. Monz. Entangling logical qubits with lattice surgery. *Nature*, 589(7841):220–224, Jan 2021.

[56] N. Ghani and J. Hedges. A compositional approach to economic game theory. *arXiv preprint arXiv:1603.04641*, 2016.

[57] D. R. Ghica and A. Jung. Categorical semantics of digital circuits. In *2016 Formal Methods in Computer-Aided Design (FMCAD)*, pages 41–48, Oct 2016.

[58] C. Gidney and A. G. Fowler. Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation. *Quantum*, 3:135, April 2019.

[59] S. Gogioso. Categorical semantics for schrödinger's equation. *arXiv:1501.06489*, 2015.

[60] S. Gogioso. A diagrammatic approach to quantum dynamics. In *8th Conference on Algebra and Coalgebra in Computer Science*, 2019.

[61] S. Gogioso and C. M. Scandolo. Categorical probabilistic theories. *EPTCS*, 266, 2018.

[62] S. Gogioso and C. M. Scandolo. Density hypercubes, higher order interference and hyper-decoherence: a categorical approach. In *International Symposium on Quantum Interaction*, pages 141–160. Springer, 2018.

[63] S. Gogioso and W. Zeng. Fourier transforms from strongly complementary observables. *arXiv preprint arXiv:1501.04995*, 2015.

[64] S. Gogioso and W. Zeng. Generalised mermin-type non-locality arguments. *Logical Methods in Computer Science (LMCS)*, 15(2), 2019.

[65] D. Gottesman. The heisenberg representation of quantum computers. *arXiv:quant-ph/9807006*, 1998.

[66] A. Hadzihasanovic. A diagrammatic axiomatisation for qubit entanglement. In *Proceedings of the 30th Annual IEEE Symposium on Logic in Computer Science (LICS)*, 2015. arXiv:1501.07082.

[67] A. Hadzihasanovic. *The algebra of entanglement and the geometry of composition*. PhD thesis, University of Oxford, 2017.

[68] A. Hadzihasanovic, K. F. Ng, and Q. Wang. Two complete axiomatisations of pure-state qubit quantum computing. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 502–511. ACM, 2018.

[69] M. Hanks, M. P. Estarellas, W. J. Munro, and K. Nemoto. Effective compression of quantum braided circuits aided by ZX-calculus. *Phys. Rev. X*, 10:041030, Nov 2020.

[70] C. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, dec 2012.

[71] E. Jeandel, S. Perdrix, and R. Vilmart. A complete axiomatisation of the ZX-calculus for Clifford+T quantum mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 559–568, 2018. arXiv preprint arXiv:1705.11151.

[72] E. Jeandel, S. Perdrix, and R. Vilmart. Diagrammatic reasoning beyond Clifford+T quantum mechanics. *arXiv preprint arXiv:1801.10142*, 2018.

[73] E. Jeandel, S. Perdrix, and R. Vilmart. A generic normal form for ZX-diagrams and application to the rational angle completeness. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10, 2019.

[74] A. Kissinger and S. Uijlen. A categorical semantics for causal structure. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

[75] A. Kissinger and J. van de Wetering. Universal MBQC with generalised parity-phase interactions and Pauli measurements. *Quantum*, 3:134, April 2019.

[76] A. Kissinger and J. van de Wetering. Pyzx: Large scale automated diagrammatic reasoning. In Bob Coecke and Matthew Leifer, editors, Proceedings 16th International Conference on *Quantum Physics and Logic,* Chapman University, Orange, CA, USA., 10-14 June 2019, volume 318 of *Electronic Proceedings in Theoretical Computer Science*, pages 229–241. Open Publishing Association, 2020.

[77] A. Kissinger and J. van de Wetering. Reducing the number of non-clifford gates in quantum circuits. *Phys. Rev. A*, 102:022406, Aug 2020.

[78] A. Kissinger and V. Zamdzhiev. Quantomatic: A proof assistant for diagrammatic reasoning. In *International Conference on Automated Deduction*, pages 326–336. Springer, 2015.

[79] S. Lack. Composing PROPs. *Theory and Applications of Categories*, 13:147–163, 2004.

[80] Y. Lafont. Towards an algebraic theory of boolean circuits. *Journal of Pure and Applied Algebra*, 184(2):257–310, 2003.

[81] A. Lang and B. Coecke. Trichromatic open digraphs for understanding qubits. In *Proceedings of the 10th workshop on Quantum Physics and Logic*, volume 95 of *Electronic Proceedings in Theoretical Computer Science*, pages 193–209, 2012.

[82] C. M. Lee and J. H. Selby. A no-go theorem for theories that decohere to quantum mechanics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2214):20170732, 2018.

[83] D. Litinski. A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery. *Quantum*, 3:128, March 2019.

[84] S. Majid. *Foundations of quantum group theory*. Cambridge University Press, 2000.

[85] K. Meichanetzidis, S. Gogioso, G. de Felice, A. Toumi, N. Chiappori, and B. Coecke. Quantum natural language processing on near-term quantum computers. *Accepted for QPL 2020*, 2020. arXiv preprint arXiv:2005.04147.

[86] K. Meichanetzidis, A. Toumi, G. de Felice, and B. Coecke. Grammar-aware question-answering on quantum computers. *arXiv preprint arXiv:2012.03756*, 2020.

[87] K. F. Ng and Q. Wang. A universal completion of the ZX-calculus. *arXiv preprint arXiv:1706.09877*, 2017.

[88] K. F. Ng and Q. Wang. Completeness of the ZX-calculus for pure qubit clifford+t quantum mechanics. 2018. arXiv preprint arXiv:1801.07993.

[89] D. Pavlovic. Monoidal computer i: Basic computability by string diagrams. *Information and computation*, 226:94–116, 2013.

[90] R. Penrose. Applications of negative dimensional tensors. In *Combinatorial Mathematics and its Applications*, pages 221–244. Academic Press, 1971.

[91] S. Perdrix and Q. Wang. Supplementarity is Necessary for Quantum Diagram Reasoning. In *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *LIPIcs*, pages 76:1–76:14, 2016.

[92] N. Pinzani and S. Gogioso. Giving operational meaning to the superposition of causal orders. *arXiv preprint arXiv:2003.13306*, 2020.

[93] N. Pinzani, S. Gogioso, and B. Coecke. Categorical semantics for time travel. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–20. IEEE, 2019.

[94] R. Raussendorf, D.E. Browne, and H.J. Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68(2):22312, 2003.

[95] M. Sadrzadeh, S. Clark, and B. Coecke. The Frobenius anatomy of word meanings I: subject and object relative pronouns. *Journal of Logic and Computation*, 23:1293–1317, 2013. arXiv:1404.5278.

[96] C. Schröder de Witt and V. Zamdzhiev. The ZX calculus is incomplete for quantum mechanics. arXiv:1404.3633, 2014.

[97] C. Schröder de Witt and V. Zamdzhiev. The ZX-calculus is incomplete for quantum mechanics. *EPTCS*, 172:285–292, December 2014.

[98] J. H. Selby and B. Coecke. Leaks: quantum, classical, intermediate and more. *Entropy*, 19(4):174, 2017.

[99] J. H. Selby, C. M. Scandolo, and B. Coecke. Reconstructing quantum theory from diagrammatic postulates. *arXiv preprint arXiv:1802.00367*, 2018.

[100] P. Selinger and X. Bian. Relations for Clifford+T operators on two qubits. *talk available from https://www.mathstat.dal.ca/ xbian/talks/*, 2015.

[101] B. Shi. Towards minimality of Clifford+T ZX-calculus. Master's thesis, University of Oxford, 2018.

[102] C. M. Signorelli, Q. Wang, and I. Khan. A compositional model of consciousness based on consciousness-only. 2020. arXiv preprint arXiv:2007.16138.

[103] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, and R. Duncan. t|ket>: A retargetable compiler for NISQ devices. *arXiv preprint arXiv:2003.10611*, 2020.

[104] P. Sobocinski. Graphical linear algebra 4: Dumbing down and magic lego. https://graphicallinearalgebra.net/2015/04/29/dumbing-down-magic-lego-and-the-rules-of-the-game-part-1/.

[105] P. Sobocinski. Representations of petri net interactions. In *CONCUR 2010 - Concurrency Theory*, volume 6269 of *LNCS*, pages pp 554–568. Springer, 2010.

[106] S. Tull and J. Kleiner. Integrated information in process theories. 2020. arXiv preprint arXiv:2002.07654.

[107] J. van de Wetering. ZX-calculus for the working quantum computer scientist. *arXiv:2012.13966*, 2020.

[108] R. Vilmart. A ZX-Calculus with Triangles for Toffoli-Hadamard, Clifford+T, and Beyond. In Peter Selinger and Giulio Chiribella, editors, *Proceedings of the 15th International Conference on Quantum Physics and Logic, Halifax, Canada, 3-7th June 2018*, volume 287 of *Electronic Proceedings in Theoretical Computer Science*, pages 313–344. Open Publishing Association, 2019.

[109] R. Vilmart. A near-minimal axiomatisation of ZX-calculus for pure qubit quantum mechanics. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10. IEEE, 2019.

[110] Q. Wang. Completeness of algebraic ZX-calculus over arbitrary commutative rings and semirings. 2020. arXiv preprint arXiv:1912.01003.

[111] Q. Wang. Enter a visual era: process theory embodied in ZX-calculus. *Slides available from 10.13140/RG.2.2.17289.67682*, 2020.

[112] R. Yeung. Diagrammatic design and study of ansätze for quantum machine learning. *arXiv preprint arXiv:2011.11073*, 2020.

[113] W. Zeng and B. Coecke. Quantum algorithms for compositional natural language processing. arXiv:1608.01406.

[114] C. Zhao and X.-S. Gao. Analyzing the barren plateau phenomenon in training quantum neural network with the ZX-calculus. *arXiv preprint arXiv:2102.01828*, 2021.