SZX-Calculus: Scalable Graphical Quantum Reasoning

Titouan Carette
CNRS, LORIA, Inria Mocqua, Université de Lorraine, F 54000 Nancy, France
titouan.carette@loria.fr

Dominic Horsman
LIG, Université Grenoble Alpes, France
dominic.horsman@univ-grenoble-alpes.fr

Simon Perdrix
CNRS, LORIA, Inria Mocqua, Université de Lorraine, F 54000 Nancy, France
`https://members.loria.fr/SPerdrix/`
simon.perdrix@loria.fr

Abstract

We introduce the Scalable ZX-calculus (SZX-calculus for short), a formal and compact graphical language for the design and verification of quantum computations. The SZX-calculus is an extension of the ZX-calculus, a powerful framework that captures graphically the fundamental properties of quantum mechanics through its complete set of rewrite rules. The ZX-calculus is, however, a low level language, with each wire representing a single qubit. This limits its ability to handle large and elaborate quantum evolutions. We extend the ZX-calculus to registers of qubits and allow compact representation of sub-diagrams via binary matrices. We show soundness and completeness of the SZX-calculus and provide two examples of applications, for graph states and error correcting codes.

2012 ACM Subject Classification Theory of computation → Quantum computation theory

Keywords and phrases Quantum computing, categorical quantum mechanics, completeness, scalability

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.55

Related Version Full proofs for this paper can be found at `arXiv:1905.00041` [7].

Funding This work is funded by ANR-17-CE25-0009 SoftQPro, PIA-GDN/Quantex, and LUE project UOQ.

1 Introduction

The ZX-calculus is an intuitive and powerful graphical language for quantum computing, introduced by Coecke and Duncan [11]. Quantum processes can be represented by ZX-diagrams, which can be seen intuitively as a generalisation of quantum circuits. The language is also equipped with a set of rewrite rules which preserves the represented quantum evolution. Unlike quantum circuits, the ZX-calculus has been proved to be complete for various universal fragments of pure quantum mechanics [26, 23, 27, 28, 43], and also mixed states quantum mechanics [8]. Completeness means that any equality can be derived in this language: if two diagrams represent the same quantum process then they can be transformed one into the other using the rewriting rules of the language. Completeness opens avenues for various applications of the ZX-calculus in quantum information processing, including circuit optimisation [16, 30] – which out-performs all other technics for T-count reductions [34] – error correcting codes [17, 21, 9], lattice surgery [14], measurement-based quantum computing [19, 15, 32] *etc.* Automated tools for quantum reasoning, e.g. Quantomatic [35] and PyZX [33], are also based on the ZX-calculus. The ZX-calculus is also used as intermediate representation in a commercial quantum compiler [12].

44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).
Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 55; pp. 55:1–55:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The cornerstone of the ZX-calculus is that fundamental properties of quantum mechanics can be captured graphically. The language remains, however, relatively low level: each wire represents a single qubit, a feature that limits the design of larger-scale and more complex quantum procedures. We address in this paper the problem of scalability of the ZX-calculus. In [9], the authors – including one of the present paper – demonstrated that the ZX-calculus can be used in practice to design and verify quantum error correcting codes. They introduced various shortcuts to deal with the scalability of the language: mainly the use of thick wires to represent registers of qubits and matrices to represent sub-diagrams, and hence reason about families of diagrams in a compact way. However, the approach lacked a general theory and fundamental properties like soundness and completeness.

Contributions. We introduce the Scalable ZX-calculus, SZX calculus for short, to provide theoretical foundations to this approach. We extend the ZX-calculus to deal with registers of qubits by introducing some new generators and rewrite rules. We show soundness – i.e. the new generators can be used in a consistent way – as well as completeness of the SZX-calculus. A simple but key ingredient is the introduction of two generators, not present in [9], for dividing and gathering registers of qubits. A wire representing a register of $(n+m)$-qubits can be divided into two wires representing respectively n and m qubits. Similarly two registers can be gathered into a single larger one. We also extend the generators of the ZX-calculus so that they can act not only on a single qubit but on a register of qubits. The SZX-calculus is then constructed as a combination of the ZX-calculus and the sub-language made of the divider and the gatherer, by adding the necessary rewrite rules describing how these two sub-languages interact. We show that the SZX-calculus is universal, sound, and complete, providing an intuitive and formal language to represent quantum operations on an arbitrarily large finite number of qubits. The use of the divider and the gatherer allows one to derive inductive (graphical) proofs.

Furthermore, the SZX-calculus provides the fundamental structures – namely the (co)commutative Hopf algebras – to develop a graphical theory of binary matrices, following work on graphical linear algebra [5]. As a consequence, we introduce an additional generator parametrized by a binary matrix together with four simple rewrite rules. Note that, while matrices were also used in [9], we introduce here a more elementary generator acting on a single register (1 input/1 output) rather than two registers (2 inputs/2 outputs). We prove completeness of the SZX-calculus augmented with these matrices. The use of matrices allows a compact representation where subdiagrams can be replaced by matrices. Moreover, basic matrix arithmetic can be done graphically. It makes the SZX-calculus with matrices a powerful tool for formal and compact quantum reasoning.

In section 5, we show the SZX-calculus in action. The main application of the SZX-calculus we consider in this paper is the graph state formalism [24]. We show how graph states can be represented using SZX-diagrams and how some fundamental properties like fixpoint properties, local complementation, and pivoting can be derived in the calculus. We also consider error correcting code examples in order to show that the techniques for the design and verification of codes developed in [9] can be performed smoothly in the SZX-calculus.

Related works. Scalability is crucial in the development of the ZX-calculus and more generally for graphical languages. We review here some contributions in this domain that we briefly compare to our approach.

The !-boxes formalism [31] is a meta language for graphical languages, which has been extensively used in the development of the automated tool Quantomatic. A !-box is a region (subdiagram) of a diagram which can be discarded or duplicated. There is also a first order logic handling families of equations between concrete (i.e. !-box free) diagrams. In contrast, the scalable ZX is not a meta-language but an actual graphical language equipped with an equational theory (namely a coloured PROP). There is no obvious way to compare these two approaches (even in terms of expressive power).

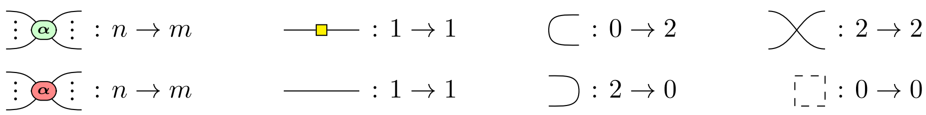
Monoidal multiplexing [10] corresponds to two categorical constructions which allow representing n diagrams in parallel. Roughly speaking, one of the two constructions would be equivalent to the use of big wires for the subclass of SZX-diagrams which are matrix, divider and gatherer-free. It is worth noticing that, to our knowledge, monoidal multiplexing has never been combined with the matrix approach, even though both were developed in the same line of research on graphical linear algebra.

Recently, Miatto [39] has independently introduced a graphical calculus involving matrices, and the equivalent of green spiders, dividers and gatherers. This graphical calculus has been developed in the context of the tensor networks, and the author mainly shows that 6 kinds of matrix products can be represented graphically. We note that the represented matrices do not coincide with the ones we are axiomatising: the matrices represented in Miatto's language correspond to $\mathbb{C}^{2^m\times 2^n}$ matrices whereas ours are in $\mathbb{F}_2^{m\times n}$, hence the equations differ. It is however worth noting that equation Fig.6 in [39] essentially corresponds to the equation governing the interaction between green spiders and the divider given in section 3.3.

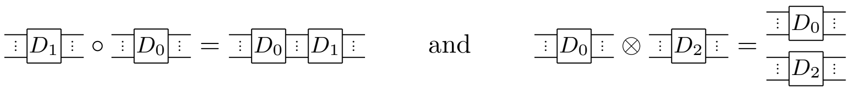
Structure of the paper. We first present the ZX-calculus in section 2, then we introduce the SZX-calculus in section 3, and an axiomatisation of binary matrices for compressing diagrams in section 4. Finally, in section 5, we use the SZX-calculus for error correcting codes and graph states. Full proofs for this paper can be found at arXiv:1905.00041 [7].

2 Background: the ZX-calculus

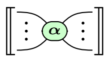
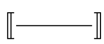
A ZX-diagram $D : k \to \ell$ with k inputs and ℓ outputs is generated by: $\forall n, m \in \mathbb{N}$, $\forall \alpha \in \mathbb{R}$,

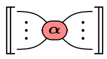
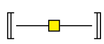


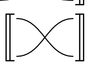
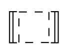
and the two compositions: for any ZX-diagrams $D_0 : a \to b$, $D_1 : b \to c$, and $D_2 : c \to d$:

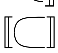


For any n, m, $ZX[n, m]$ is the set of all ZX-diagrams of type $n \to m$. The ZX-diagrams are representing quantum processes: for any ZX-diagram $D : n \to m$ its interpretation $[\![D]\!] \in \mathcal{M}_{2^m\times 2^n}(\mathbb{C})$ is inductively defined as: $[\![D_1 \circ D_0]\!] = [\![D_1]\!] \circ [\![D_0]\!]$, $[\![D_0 \otimes D_2]\!] = [\![D_0]\!] \otimes [\![D_2]\!]$, and

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

 $:= |+^m\rangle\langle +^n| + e^{i\alpha}|-^m\rangle\langle -^n|$  $:= |+\rangle\langle 0| + |-\rangle\langle 1|$

 $:= |00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11|$  $:= 1$

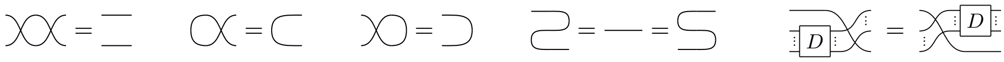
 $:= |00\rangle + |11\rangle$  $:= \langle 00| + \langle 11|$

Where $|0\rangle{:=}\binom{1}{0}$, $|1\rangle{:=}\binom{0}{1}$, $|+\rangle{:=}\frac{|0\rangle+|1\rangle}{\sqrt{2}}$, $|-\rangle{:=}\frac{|0\rangle-|1\rangle}{\sqrt{2}}$, $|a^{k+1}\rangle{:=}|a\rangle\otimes|a^k\rangle$, $|a^0\rangle{:=}1$, and $\langle a|{:=}|a\rangle^\dagger$, moreover n and m are respectively the number of inputs and outputs of the spiders.

When equal to zero, the angle of the green or red spider is omitted:

ZX-diagrams are *universal* for pure qubit quantum mechanics: $\forall n, m \in \mathbb{N}$, and $\forall M \in \mathcal{M}_{2^n \times 2^m}(\mathbb{C})$, there exists a ZX-diagram $D : n \to m$ such that $[\![D]\!] = M$.

ZX-diagrams also come with a set of graphical rewrite rules, or axioms, which allows one to transform a diagram preserving its interpretation. Some of them are gathered under the *Only Topology Matters* paradigm. When using these we label the equality by top. Two diagrams that can be transformed into each other by moving around the wires are equal. This can be derived from the following rules:



The last set of rule expresses the naturality of the swap; in other words, that all the generators can be passed through wires.

The legs of the spiders of ZX-calculus can be exchanged and bent. This implies that diagrams are essentially graphs with inputs and outputs.

Finally, the rules that are not purely topological are given in Figure 1.

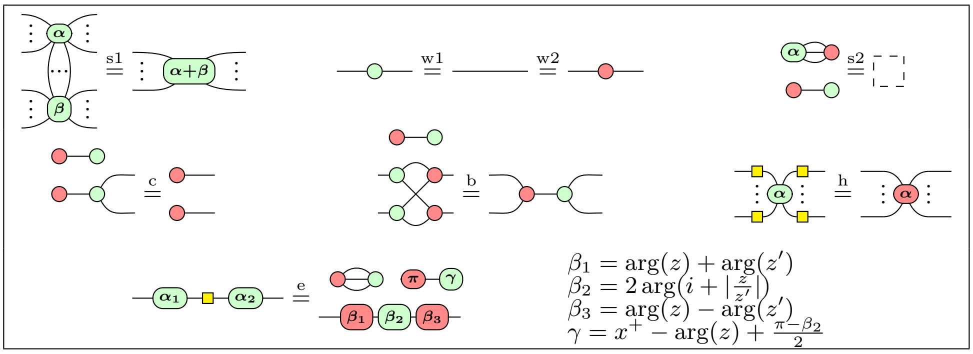


Figure 1 Axioms of the ZX-calculus, where $x^+ := \frac{\alpha_1 + \alpha_2}{2}$, $x^- := x^+ - \alpha_2$, $z := -\sin(x^+) + i\cos(x^-)$, $z' := \cos(x^+) - i\sin(x^-)$ and $z' = 0 \Rightarrow \beta_2 = 0$. In the upper left rule, there must be at least one wire between the spiders annotated by α and β. The colour-swapped version of those rules also holds. A label is given to each axiom, above the equals sign, for later reference.

We write $ZX \vdash D = D'$ when D can be transformed into D' using the rules of the ZX-calculus. The rules of the ZX-calculus are sound: for any ZX-diagrams D, D', $ZX \vdash D = D' \Rightarrow [\![D]\!] = [\![D']\!]$ i.e., the rules preserve the interpretation of the language. The language is also complete: for any ZX-diagrams D, D', $[\![D]\!] = [\![D']\!] \Rightarrow ZX \vdash D = D'$ i.e., whenever two diagrams represent the same quantum evolution, we can transform one into the other using the rules of the language [43].

3 The scalable ZX-calculus

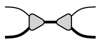
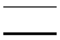
In the ZX-calculus, each wire represents a single qubit. Therefore, a system acting on n qubits will be represented by an n-input diagram. This quickly leads to intractable diagrams when it comes to big systems. The extension to the SZX-calculus presented here provides a more compact notation.

3.1 Divide and gather, a calculus for big wires

The input (resp. output) type of a ZX-diagram is its number of input wires, and hence number of input qubits. In the SZX-calculus, wires represent registers of qubits. A wire of type 1_n represents a register of n qubits. A type of the SZX-calculus is then a formal sum of the form $\sum_i 1_{n_i}$, the empty sum being denoted by 0. In other words, the set of types of SZX-calculus is the free monoid over $\mathbb{N}^*$, the set of positive integers. We denote it $\langle\mathbb{N}^*\rangle$. Graphically, we represent the wire of type 1_n by an bold font wires labelled by n[1], a label that is omitted when it is not ambiguous. A normal font wire always denotes a single qubit register of type 1_1. By convention the sum of m wires of type 1_n is denoted m_n with $0_n = m_0 = 0$. n_1 is simply written n. Given a type $a = \sum_i 1_{n_i}$, its size is defined as $S(a) := \sum_i n_i$.

Big wires can be divided into smaller ones and, conversely, can be gathered to form bigger ones. For any $n \in \mathbb{N}$, we introduce two new generators: the *divider* and *gatherer* of size n. They are depicted as follows:

We take the convention that the divider and the gatherer of size 0 are the identity. We define a fragment of the SZX, the wire calculus $\mathbb{W}$.

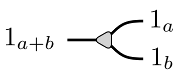
▶ **Definition 1** ($\mathbb{W}$-calculus). *The $\mathbb{W}$-calculus is defined as the graphical language generated by identity wires, the dividers, and the gatherers of any size, and satisfying the elimination rule*  $\overset{\mathrm{E}}{=}$  *and the expansion rule*  $\overset{\mathrm{P}}{=}$.

The roles of the dividers and gatherers in the equations are perfectly symmetric, so each time something is shown for dividers it also holds for gatherers by symmetry.

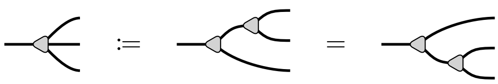
We now show a coherence theorem for scalable calculi: the rewiring theorem. It states that two diagrams of the $\mathbb{W}$-calculus with the same type are equal.

▶ **Theorem 2.** *Let $\omega \in \mathbb{W}[a,b]$ and $\omega' \in \mathbb{W}[c,d]$: $\mathbb{W} \vdash \omega = \omega' \quad \Leftrightarrow \quad a = c$ and $b = d$*

This theorem has strong consequences. We can define generalized dividers able to divide any wire of size 1_{a+b} into a wire of size 1_a and a wire of size 1_b.



Those generalized dividers have a unique possible interpretation as diagrams of $\mathbb{W}$-calculus given by their types, and we know exactly the equations they verify: all the well typed ones. In particular, an associativity-like law holds for generalized wires allowing us to define n-ary generalized dividers.



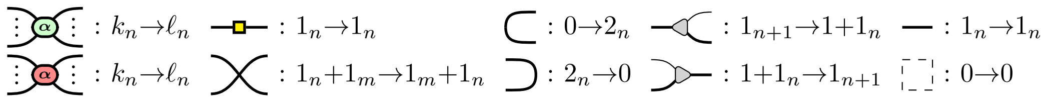
Each time we use the property that any well typed equation in $\mathbb{W}$ is true, we will label the equality by R.

[1] On a blackboard the bold font might be advantageously replaced by struck-out wires.

3.2 The SZX-diagrams

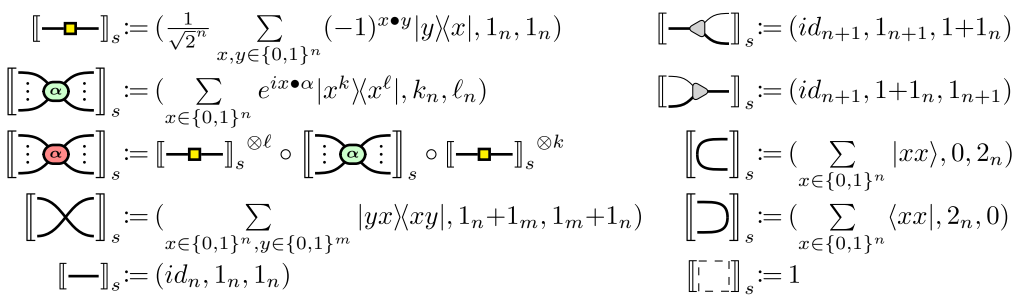
We now fuse the $\mathbb{W}$-calculus and the ZX-calculus into one language: the full SZX-calculus.

The generators of SZX-diagrams are: $\forall n, m \in \mathbb{N}^*, \forall k, \ell \in \mathbb{N}$, $\forall \alpha \in \mathbb{R}^n$,



SZX-generators can be combined using the usual sequential and spacial compositions to form SZX-diagrams. Note that for $n = m = 1$ we recover all the generators of the ZX-calculus. We denote them, as in the ZX-calculus, using thin wires e.g. for $: 1_1 \to 1_1$. Any big wire can be labelled by its size $: 1_n \to 1_n$ to avoid ambiguity. Such labels will be used mainly for scalars i.e. diagrams with no input/output. Each green or red spider is parametrised by a vector $\alpha \in \mathbb{R}^n$ of angles. With slight abuse of notation we use a single angle $\alpha_0 \in \mathbb{R}$ to denote the vector $(\alpha_0, \ldots, \alpha_0) \in \mathbb{R}^n$ when the spider has at least one leg $(k + \ell > 0)$ so that this leg can be labelled by n to avoid a potential ambiguity. Like in the ZX-calculus, the angle α_0 is omitted when $\alpha_0 = 0$.

The interpretation of ZX-diagrams is extended to SZX-diagrams as follows: for any SZX-diagram $D : a \to b$, its interpretation $[\![D]\!]_s$ is a triplet (M, a, b) where $M \in \mathcal{M}_{2^{S(b)} \times 2^{S(a)}}(\mathbb{C})$. $[\![D]\!]_s$ is inductively defined as: $[\![D_1 \circ D_0]\!]_s = (M_1 \circ M_0, a, c)$, $[\![D_0 \otimes D_2]\!]_s = (M_0 \otimes M_2, a + c, b + d)$ where $[\![D_0]\!]_s = (M_0, a, b)$, $[\![D_1]\!]_s = (M_1, b, c)$, and $[\![D_2]\!]_s = (M_2, c, d)$. Moreover:



Where $\forall u, v \in \mathbb{R}^m$, $u \bullet v = \sum_{i=1}^m u_i v_i$, $M^{\otimes 0} = 1$, and $M^{\otimes k+1} = M \otimes M^{\otimes k}$.

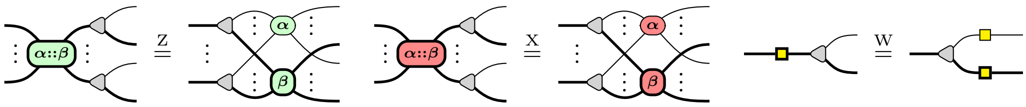
▶ **Theorem 3** (Universality). *SZX-diagrams are universal for pure qubit quantum mechanics:* $\forall a, b \in \langle \mathbb{N}^* \rangle, \forall M \in \mathcal{M}_{2^{S(b)} \times 2^{S(a)}}(\mathbb{C})$, $\exists D : a \to b$ *such that* $[\![D]\!]_s = (M, a, b)$.

3.3 The calculus

The SZX-calculus is based on distribution rules that allow dividers and gatherers to go through the big generators. For this to work we need first to ensure that the swap behaves naturally with respect to dividers and gatherers. This is given by the following two rules:

Then the rules governing the interaction between dividers, gatherers and the so-called *cups* and *caps* are:

We put labels over the equals signs to allow subsequent reference to the rules. These rules are sufficient to fully describe possible interactions between wires of any size, gatherers and dividers. It remains to specify how dividers and gatherers interact with big generators:



Where $\alpha{::}\beta$ means that we append the phase $\alpha \in \mathbb{R}$ to the (generalized) phase $\beta \in \mathbb{R}^n$.

This completes the set of rules of the SZX-calculus. Note that all rules agree with the interpretation, ensuring soundness of the SZX-calculus.

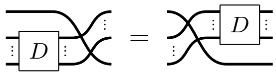
We see that any big generator s_n is in fact just n copies of the corresponding size one generator s acting in parallel. That is, a parallel composition but with a particular permutation of the inputs and outputs. Such constructions are called multiplexed diagrams in [10]. Multiplexed diagrams are shown to satisfy the same equations as size 1 diagrams. The following lemma states the same results for big generators:

▶ **Lemma 4.** *For any rule of the ZX-calculus, and any $n \in \mathbb{N}^*$, the equation obtained by replacing each generator by its big version of size n is provable in the SZX-calculus.*

We can go even further than Lemma 4. In fact, the SZX-calculus is complete:

▶ **Theorem 5.** $\forall a, b \in \langle \mathbb{N}^* \rangle, \forall D, D' \in \mathcal{S}ZX[a, b],\ \llbracket D \rrbracket_s = \llbracket D' \rrbracket_s \Rightarrow SZX \vdash D = D'$.

Theorem 5 has interesting graphical consequences, ensuring that the *Only Topology Matters* paradigm applies to the SZX-calculus. In particular, swaps of any size behave naturally with respect to any diagram:



This suggests a more compact presentation close to the one of the ZX-calculus, given in the next subsection.

3.4 Compact axiomatisation

Assuming that *Only Topology Matters*, the SZX-calculus enjoys a more compact axiomatisation:

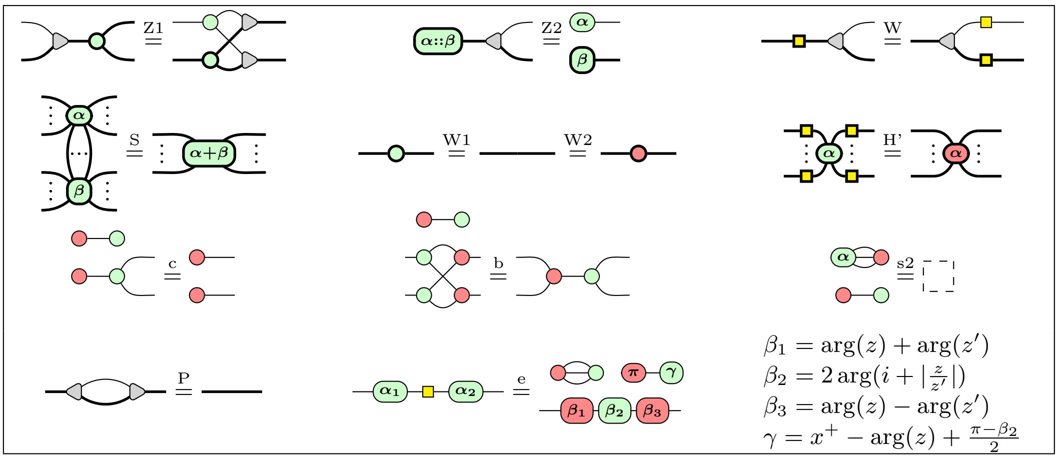
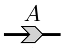
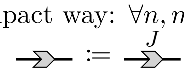
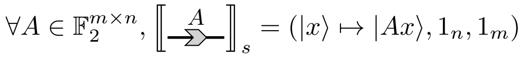


Figure 2 Axioms of the SZX-calculus, where $x^+ := \frac{\alpha_1+\alpha_2}{2}$, $x^- := x^+ - \alpha_2$, $z := -\sin(x^+) + i\cos(x^-)$, $z' := \cos(x^+) - i\sin(x^-)$ and $z' = 0 \Rightarrow \beta_2 = 0$. In the spider fusion rule, there must be at least one wire between the spiders annotated by α and β. The colour-swapped versions of those rules also hold. The bold font wires stand for wires of any size $n \geq 1$.

▶ **Lemma 6.** *All the rules of the SZX-calculus can be derived from the compact axioms of Figure 2 together with the Only Topology Matters paradigm.*

4 Axiomatising binary matrices for compressing diagrams

In this section, we introduce a new generator for the SZX-calculus, parametrized by a binary matrix, allowing us to represent large graphical structures in a compact way: $\forall n, m \in \mathbb{N}^*$, $\forall A \in \mathbb{F}_2^{m\times n}$,  : $1_n \to 1_m$. All-ones matrices will be omitted:  where $\forall i, j, J_{i,j} = 1$. The new generator is interpreted as follows:



where the matrix product Ax is in $\mathbb{F}_2$ and x is seen as a column vector i.e. $(Ax)_i = \sum_{k=1}^{n} A_{i,k}x_k \bmod 2$.

▶ Remark 7. Note that, compared to [9], the matrix is not necessarily connected to green and red spiders. It is therefore a more elementary generator.

Those matrices are required to satisfy the four axioms given in Figure 3, which are sound.

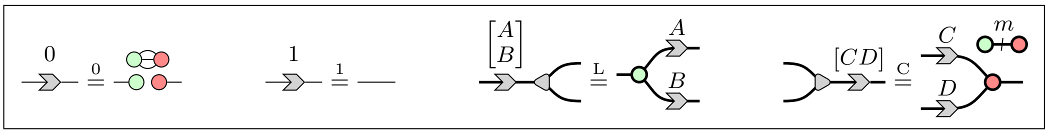
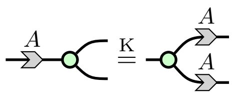
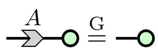


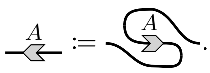
Figure 3 Axioms for matrices, where $A \in \mathbb{F}_2^{a\times n}$, $B \in \mathbb{F}_2^{b\times n}$, $C \in \mathbb{F}_2^{m\times c}$ and $D \in \mathbb{F}_2^{m\times d}$. $\begin{bmatrix} A \\ B \end{bmatrix}$ and $[CD]$ are block matrices.

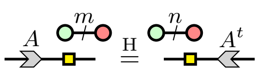
▶ Remark 8. The rules of the ZX-calculus define a scaled Hopf algebra between the green and red structure. This algebra is commutative and cocommutative with a trivial antipode. Thus, following the work of [44], the notion of $\{0, 1\}$-matrices naturally emerges. It is worth noticing that it coincides with the matrices we are introducing in this section. Notice however that our axiomatisation of the matrices strongly relies on their interaction with the divider and the gatherer, which are not present in [44].

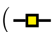
In the following, the SZX-calculus refers to the SZX-calculus augmented with the matrix generators and the axioms of Figure 3.

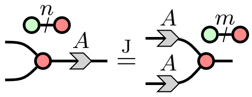
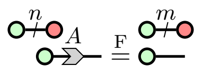
Useful equations can be derived. First, matrices are copied and erased by green nodes.

▶ **Lemma 9.** *For any $A \in \mathbb{F}_2^{m\times n}$, SZX $\vdash$*  *and SZX $\vdash$* 

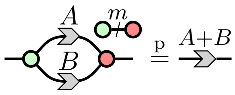
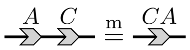
We define backward matrices as follows: .

▶ **Lemma 10.** $\forall A \in \mathbb{F}_2^{m\times n}$, *SZX $\vdash$*  *where A^t is the transpose of A.*

As a consequence, conjugating by Hadamard () reverses the orientation and transposes the matrix (up to scalars). Since conjugating by Hadamard colour-swaps the spiders and preserves the other generators of the language, one can derive from any equation a new one (up to scalars) which consists in colour-swapping the spiders, transposing the matrices and then changing their orientation. For instance Lemma 9 gives that matrices are cocopied and coerased by red nodes:

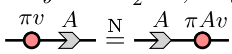
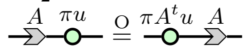
▶ **Lemma 11.** *For any $A \in \mathbb{F}_2^{m\times n}$, SZX* $\vdash$  *and SZX* $\vdash$ 

Basic matrix operations like addition and multiplication (in $\mathbb{F}_2$) can be implemented graphically:

▶ **Lemma 12.** *For any $A, B \in \mathbb{F}_2^{m\times n}$, and any $C \in \mathbb{F}_2^{k\times m}$, SZX* $\vdash$  *and SZX* $\vdash$ .

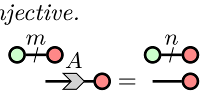
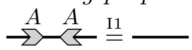
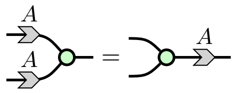
Whereas all the previous properties about matrices are angle-free, some spiders whose angles are multiple of π can be pushed through matrices as follows:

▶ **Lemma 13.** *For any $A \in \mathbb{F}_2^{m\times n}$, any $v \in \mathbb{F}_2^n$ and any $u \in \mathbb{F}_2^m$,*

SZX $\vdash$  *and* *SZX* $\vdash$ 

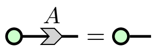
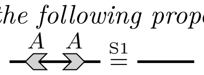
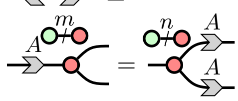
Injective matrices enjoy some specific properties:

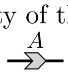
▶ **Lemma 14.** *For any $A \in \mathbb{F}_2^{m\times n}$, the following properties are equivalent:*

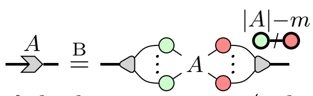
(1) *A is injective.*
(2) *SZX* $\vdash$ 
(3) *SZX* $\vdash$ 
(4) *SZX* $\vdash$ 

By Hadamard conjugation, we obtain some dual properties for surjective matrices:

▶ **Lemma 15.** *For any $A \in \mathbb{F}_2^{m\times n}$, the following properties are equivalent:*

(1) *A is surjective.*
(2) *SZX* $\vdash$ 
(3) *SZX* $\vdash$ 
(4) *SZX* $\vdash$ 

Due to the universality of the SZX-calculus, matrices are expressible as SZX-diagrams, and the matrix generator  is actually a compact representation of a green/red bipartite graphs whose biadjacency matrix is A:

▶ **Lemma 16.** *For any $A \in \mathbb{F}_2^{m\times n}$, SZX* $\vdash$  *where A represents in the RHS diagram the adjacency matrix of the bipartite green/red graph, and $|A|$ is the number of* 1 *in A.*

Lemma 16 and 5 imply the completeness of the SZX-calculus with matrices:

▶ **Theorem 17.** *SZX-calculus with matrices is complete.*

5 Applications

This section provides two examples of the SZX-calculus in action.

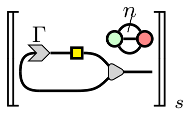
5.1 Application to graph states

Graph states [24] form a subclass of quantum states that can be represented by simple undirected graphs where each vertex represents a qubit and the edges represent intuitively the entanglement between qubits. The graph state formalism is widely used in quantum information processing, providing combinatorial characterisations of quantum properties in measurement-based quantum computing [40, 13, 6], secret sharing [37, 29, 25], error correcting codes [41, 4] etc. Graph states are also strongly related to the ZX-calculus [18] where they have been used for instance in proving the completeness of some fragments [1, 20]. A graph state is a particular kind of stabilizer state and thus can be defined as a fixpoint: given a graph G of order n, the corresponding graph state $|G\rangle$ is a the unique state (up to a global phase) such that for any vertex u, applying $X = [\![-\pi-]\!]$ on u and $Z = [\![-\pi-]\!]$ on its neighbours leaves the state unchanged. The global phase is fixed by the extra condition $\langle 0^n|G\rangle = \frac{1}{\sqrt{n}}$.

A graph state admits a simple representation as a ZX-diagram: each vertex is represented by a green spider connected to an output, and each edge is represented by a Hadamard (–□–) connecting the corresponding green dots. In the following, we provide two alternative, scalable, representations of graph states: the first is a compact matrix-based representation of bipartite graph states, the second is an inductive definition of arbitrary graph state, allowing inductive proofs. In both representations, we provide diagrammatic proofs of some key properties of the graph states.

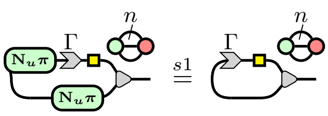
First, any bipartite graph state can be depicted with a SZX-diagram via its biadjacency matrix:

▶ **Lemma 18.** *For any bipartite graph G with biadjacency matrix $\Gamma \in \mathbb{F}_2^{m\times n}$,*

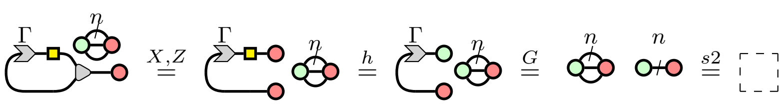


$$= (|G\rangle, 0, 1_{n+m})$$

Proof of Lemma 18. The last two components are straightforward typing, for the first one we use the characterization of $|G\rangle$ by its stabilizer [24]. $|G\rangle$ is the unique (up to a scalar) common fix point of $X_u Z_{N_u}$ for all vertices u of G. Each subset of vertices is identified with its characteristic vector, e.g. $\mathbf{N}_u\pi$ is a vector with a π at the position i if the i-th vertex is a neighbour of u, and 0 otherwise. The following proof uses the fact that $\Gamma^t u = \mathbf{N}_u$. We assume u is in the first part of the bipartite graph (the other case is similar):



It remains to take care of the scalar. We see that $\sqrt{2^{n+m}}\langle 0^{n+m}|G\rangle = 1$:

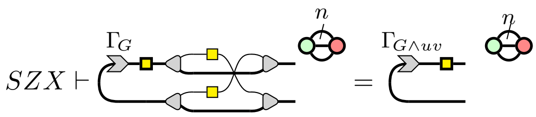


◀

A fundamental property of graph states is that graph transformations (like pivoting and local complementation) can be performed on graph states using local operations. Given a bipartite graph G, pivoting according to an edge (u, v) produces a graph denoted $G \wedge uv$ where the labels u and v are exchanged and their neighbourhood is complemented: for any $w \in \mathbf{N}_u \setminus v$ and $t \in \mathbf{N}_v \setminus u$, w and t are connected in $G \wedge uv$ iff they are not connected in G. Pivoting can be implemented on bipartite graph states by simply applying Hadamard on u

and v: $H_{u,v}|G\rangle = |G \wedge uv\rangle$ [42, 38]. This property can be derived in the SZX-calculus, and its proof, given in arXiv:1905.00041 due to space limits, provides an interesting example of the SZX-calculus in action:

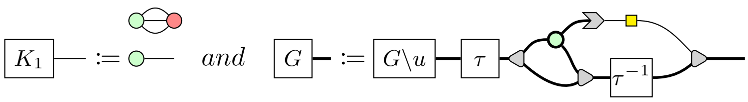
▶ **Lemma 19.** *Given a bipartite graph G and an edge (u, v),*



where Γ_G (resp. $\Gamma_{G\wedge uv}$) is the biadjacency matrix of G (resp. $G \wedge uv$) such that u corresponds to the first row (resp. column) and v to the first column (resp. row).

Now we introduce a general inductive definition of graph state boxes, which associates a SZX-diagram with any (not necessarily bipartite) graph.

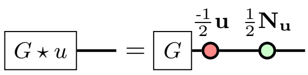
▶ **Definition 20.** *Given a graph G with ordered vertices, the corresponding graph state box is defined by:*



where K_1 is the graph of order 1, u is the first vertex of G, τ is a permutation on the list of vertices of $G\backslash u$ which puts the neighbourhood of u first and then the other vertices.

▶ **Lemma 21.** $\left\llbracket \boxed{G} \right\rrbracket_s = (|G\rangle, 0, 1_n)$.

We will now use the SZX-calculus to show the property known as local complementation. Given a vertex u of a graph G the local complementation of G according to u is the graph $G \star u$ which is G where all edges between neighbours of u have been complemented, that is, edges became non-edges and non-edges became edges.

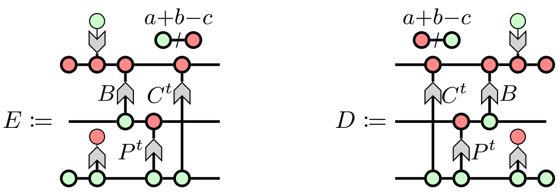
▶ **Theorem 22.** *For any graph G and vertex u, $SZX \vdash$* .

5.2 Application to error correcting codes

The original motivation for the development of a scalable ZX-calculus was the design of tripartite Coherent Parity Checking (CPC) error correcting codes [9]. We reformulate here in the SZX-calculus the definition of those codes and the proof of some elementary properties.

The idea is to spread the information of some logical qubits over a bigger number of physical qubits. In our example the code is parametrized by three matrices $B \in \mathbb{F}_2^{a\times b}$, $P \in \mathbb{F}_2^{c\times b}$ and $C \in \mathbb{F}_2^{c\times a}$. The aim is to encode b logical qubits into $a+b+c$ physical ones.

▶ **Definition 23.** *The tripartite CPC encoder $E : 1_b \to 1_a+1_b+1_c$ and decoder $D : 1_a+1_b+1_c \to 1_b$ defined by the matrices $B \in \mathbb{F}_2^{a\times b}$, $P \in \mathbb{F}_2^{c\times b}$ and $C \in \mathbb{F}_2^{c\times a}$ are:*

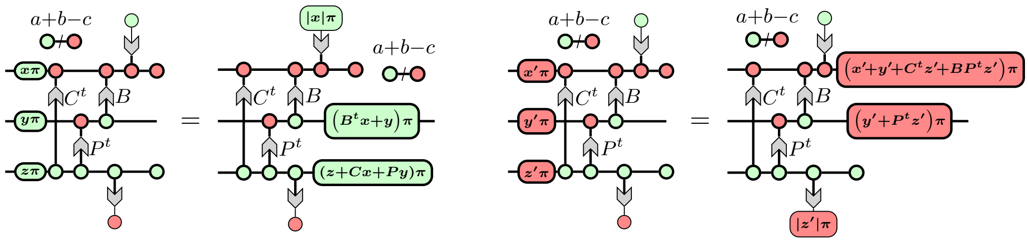


We can prove that the code is correct when there are no errors, in other words:

▶ **Lemma 24.** *The encoder is an isometry that is $SZX \vdash D \circ E = $ —.*

We now end by showing what happens when errors go through the decoder: x, y and z (resp. x', y', z') are vectors of phase flip errors (resp. bit flip errors). The implementation of the decoder involves some measurements, which according to Lemma 25, produce some syndromes ($|x| = \sum_i x_i \bmod 2$, $z{+}Cx{+}Py$, $x'{+}y'{+}C^tz'{+}BP^tz'$, and $|z'|$) which guide us to correct the middle wire. Of course the exact protocol and its efficiency depend on clever choices of B, P and C, see [9] for details.

▶ **Lemma 25.** *The following equalities hold in the SZX-calculus:*



6 Conclusion and further work

We have introduced the SZX-calculus, a formal and compact graphical language for quantum reasoning, and proved its universality, soundness, and completeness. This work is addressing two main objectives. First, to demonstrate that some of the ingredients for scalability which were sketched out in [9] – like the thick wires and the use of matrices – together with some new ingredients – like the divider and the gatherer – can be axiomatised to provide a complete scalable graphical language. Our second objective was to provide a sufficiently precise definition of the language to consider an implementation in a graphical proof assistant like Quantomatic. This last point would pave the way towards the formal verification of large scale quantum protocols and algorithms.

We aim to provide a language ready for applications and available to most of the quantum computing community. For this reason, we have deliberately avoided a categorical presentation. A fully categorical description of the scalable construction will be the subject of further work. We nevertheless provide here a sketch of how our construction can be generalised in a categorical setting. Graphical languages can be defined as props, see [3] and [44], that is symmetric strict monoidal categories whose set of objects is freely generated by one object we denote 1. In fact it is possible to define a scalable construction for any coloured prop. Given a set C of colours we can define two $\langle\langle C\rangle - \varepsilon\rangle$-coloured props $\mathbb{D}_C$ and $\mathbb{G}_C$ whose objects are formal sums of 1_{n_c} and morphisms are respectively generated by dividers and gatherers for each pair (n, c). The elimination rule is a distribution rule as in [36], which allows us to define the composed prop $\mathbb{D}_C; \mathbb{G}_C$. The prop of wires $\mathbb{W}_C$ is then defined as this composition quotiented by the expansion rule. This last pro satisfies a rewire theorem similar to Theorem 2. Then given a C-coloured prop $\mathbf{P}$, we define the $\langle\langle C\rangle - \varepsilon\rangle$-coloured prop $\overline{\mathbf{P}}$ which has the same generators and equations as those of $\mathbf{P}$ on wires of size 1. Finally the scalable prop $\mathcal{S}\mathbf{P}$ is defined as the composition of prop $\mathbb{D}_C; \overline{\mathbf{P}}; \mathbb{G}_C$ quotiented by the expansion rule. The corresponding distribution rules follows the same pattern as in 3.3. Such a generalization gives scalable versions of any graphical language based on props such as the ZW–calculus [22], the ZH-calculus [2] or $\mathbb{IH}$ [5].

References

1 Miriam Backens. The ZX-Calculus is Complete for Stabilizer Quantum Mechanics. *New Journal of Physics*, 16(9):093021, September 2014. doi:10.1088/1367-2630/16/9/093021.

2 Miriam Backens and Aleks Kissinger. ZH: A complete graphical calculus for quantum computations involving classical non-linearity. In Peter Selinger and Giulio Chiribella, editors, Proceedings of the 15th International Conference on *Quantum Physics and Logic,* Halifax, Canada, 3-7th June 2018, volume 287 of *Electronic Proceedings in Theoretical Computer Science*, pages 23–42. Open Publishing Association, 2019. doi:10.4204/EPTCS.287.2.

3 John C Baez, Brandon Coya, and Franciscus Rebro. Props in network theory. *arXiv preprint arXiv:1707.08321*, 2017. arXiv:1707.08321.

4 BA Bell, DA Herrera-Martí, MS Tame, D Markham, WJ Wadsworth, and JG Rarity. Experimental demonstration of a graph state quantum error-correction code. *Nature communications*, 5:3658, 2014.

5 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. Interacting Hopf algebras. *Journal of Pure and Applied Algebra*, 221(1):144–184, 2017.

6 Daniel E Browne, Elham Kashefi, Mehdi Mhalla, and Simon Perdrix. Generalized flow and determinism in measurement-based quantum computation. *New Journal of Physics*, 9(8):250, 2007. URL: http://stacks.iop.org/1367-2630/9/i=8/a=250.

7 Titouan Carette, Dominic Horsman, and Simon Perdrix. SZX-calculus: Scalable graphical quantum reasoning, 2019. arXiv:1905.00041.

8 Titouan Carette, Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Completeness of Graphical Languages for Mixed States Quantum Mechanics. In *International Colloquium on Automata, Languages, and Programming (ICALP'19)*, 2019.

9 Nicholas Chancellor, Aleks Kissinger, Joschka Roffe, Stefan Zohren, and Dominic Horsman. Graphical structures for design and verification of quantum error correction. *arXiv preprint arXiv:1611.08012*, 2016. arXiv:1611.08012.

10 Apiwat Chantawibul and Paweł Sobociński. Monoidal Multiplexing. In *International Colloquium on Theoretical Aspects of Computing*, pages 116–131. Springer, 2018.

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

12 Alexander Cowtan, Silas Dilkes, Ross Duncan, Alexandre Krajenbrink, Will Simmons, and Seyon Sivarajah. On the qubit routing problem. *arXiv preprint arXiv:1902.08091*, 2019. arXiv:1902.08091.

13 Vincent Danos, Elham Kashefi, Prakash Panangaden, and Simon Perdrix. Extended Measurement Calculus. *Semantic Techniques in Quantum Computation*, 2010.

14 Niel de Beaudrap and Dominic Horsman. The ZX calculus is a language for surface code lattice surgery, 2017. arXiv:1704.08670.

15 Ross Duncan. A graphical approach to measurement-based quantum computing. *arXiv preprint arXiv:1203.6242*, 2012. arXiv:1203.6242.

16 Ross Duncan, Aleks Kissinger, Simon Pedrix, and John van de Wetering. Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus. *arXiv preprint arXiv:1902.03178*, 2019. arXiv:1902.03178.

17 Ross Duncan and Maxime Lucas. Verifying the Steane code with Quantomatic. *Electronic Proceedings in Theoretical Computer Science*, 171:33–49, 2014. arXiv preprint arXiv:1902.03178. arXiv:1902.03178.

18 Ross Duncan and Simon Perdrix. Graph states and the necessity of Euler decomposition. In *Conference on Computability in Europe (CiE)*, pages 167–177. Springer, 2009.

19 Ross Duncan and Simon Perdrix. Rewriting Measurement-Based Quantum Computations with Generalised Flow. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming*, pages 285–296, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

20 Ross Duncan and Simon Perdrix. Pivoting Makes the ZX-Calculus Complete for Real Stabilizers. In *QPL 2013*, Electronic Proceedings in Theoretical Computer Science, pages 50–62, 2013. doi:10.4204/EPTCS.171.5.

21 Craig Gidney and Austin G Fowler. Efficient magic state factories with a catalyzed |CCZ> to 2|T> transformation. *arXiv preprint arXiv:1812.01238*, 2018. arXiv:1812.01238.

22 Amar Hadzihasanovic. The algebra of entanglement and the geometry of composition. *arXiv preprint arXiv:1709.08086*, 2017. arXiv:1709.08086.

23 Amar Hadzihasanovic, Kang Feng Ng, and Quanlong Wang. Two Complete Axiomatisations of Pure-state Qubit Quantum Computing. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 502–511, New York, NY, USA, 2018. ACM. doi:10.1145/3209108.3209128.

24 Marc Hein, Wolfgang Dür, Jens Eisert, Robert Raussendorf, M Nest, and H-J Briegel. Entanglement in graph states and its applications. *Proceedings of the International School of Physics "Enrico Fermi" on "Quantum Computers, Algorithms and Chaos"*, 2006. arXiv:quantph/0602096.

25 Jérôme Javelle, Mehdi Mhalla, and Simon Perdrix. New Protocols and Lower Bounds for Quantum Secret Sharing with Graph States. In Kazuo Iwama, Yasuhito Kawano, and Mio Murao, editors, *Theory of Quantum Computation, Communication, and Cryptography*, volume 7582 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-35656-8_1.

26 Emmanuel Jeandel, Simon Perdrix, and Renaud 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 (LICS)*, pages 559–568. ACM, 2018.

27 Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A Generic Normal Form for ZX-Diagrams and Application to the Rational Angle Completeness. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2019. arXiv:1805.05296.

28 Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Completeness of the ZX-Calculus. *arXiv preprint arXiv:1903.06035*, 2019. arXiv:1903.06035.

29 Elham Kashefi, Damian Markham, Mehdi Mhalla, and Simon Perdrix. Information flow in secret sharing protocols. In *Proc. 5th Workshop on Developments in Computational Models (DCM).*, volume 9, pages 87–97. EPTCS, 2009.

30 Aleks Kissinger and Arianne Meijer-van de Griend. CNOT circuit extraction for topologically-constrained quantum memories. *arXiv preprint arXiv:1904.00633*, 2019. arXiv:1904.00633.

31 Aleks Kissinger and David Quick. A first-order logic for string diagrams. *arXiv preprint arXiv:1505.00343*, 2015. arXiv:1505.00343.

32 Aleks Kissinger and John van de Wetering. Universal MBQC with generalised parity-phase interactions and Pauli measurements. *arXiv:1704.06504*, 2017. arXiv:1704.06504.

33 Aleks Kissinger and John van de Wetering. PyZX, 2018. URL: https://github.com/Quantomatic/pyzx.

34 Aleks Kissinger and John van de Wetering. Reducing T-count with the ZX-calculus. *arXiv preprint arXiv:1903.10477*, 2019. arXiv:1903.10477.

35 Aleks Kissinger and Vladimir Zamdzhiev. Quantomatic: A proof assistant for diagrammatic reasoning. In *International Conference on Automated Deduction*, pages 326–336. Springer, 2015.

36 Stephen Lack. Composing props. *Theory and Applications of Categories*, 13(9):147–163, 2004.

37 Damian Markham and Barry C. Sanders. Graph States for Quantum Secret Sharing. *Physical Review A*, 78:042309, 2008. URL: doi:10.1103/PhysRevA.78.042309.

38 Mehdi Mhalla and Simon Perdrix. Graph states, pivot minor, and universality of (X, Z)-measurements. *arXiv preprint arXiv:1202.6551*, 2012. arXiv:1202.6551.

39 Filippo M Miatto. Graphical Calculus for products and convolutions. *arXiv preprint arXiv:1903.01366*, 2019. arXiv:1903.01366.

40 Robert Raussendorf, Daniel E. Browne, and Hans J. Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68(2), August 2003. doi:10.1103/physreva.68.022312.

41 D. Schlingemann and R. F. Werner. Quantum error-correcting codes associated with graphs. *Physical Review A*, 65, 2001. doi:10.1103/PhysRevA.65.012308.

42 Maarten Van den Nest. *Local equivalence of stabilizer states and codes.* PhD thesis, Faculty of Engineering, K. U. Leuven, Belgium, May 2005.

43 Renaud Vilmart. A Near-Optimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2019. arXiv:1812.09114.

44 Fabio Zanasi. Interacting Hopf Algebras: the theory of linear systems. *arXiv preprint arXiv:1805.03032*, 2018. arXiv:1805.03032.