UNIVERSITY OF BIRMINGHAM

Research at Birmingham

A complete graphical calculus for Spekkens' Toy Bit theory

Backens, Miriam; Duman, Ali Nabi

A complete graphical calculus for Spekkens' toy bit theory

Miriam Backens

Department of Computer Science,
University of Oxford,
Oxford, United Kingdom

`miriam.backens@cs.ox.ac.uk`

Ali Nabi Duman

Department of Mathematics,
King Fahd University of Petroleum & Minerals,
Dhahran, Saudi Arabia

`alinabi@gmail.com`

While quantum theory cannot be described by a local hidden variable model, it is nevertheless possible to construct such models that exhibit features commonly associated with quantum mechanics. These models are also used to explore the question of ψ-ontic versus ψ-epistemic theories for quantum mechanics. Spekkens' toy theory is one such model. It arises from classical probabilistic mechanics via a limit on the knowledge an observer may have about the state of a system. The toy theory for the simplest possible underlying system closely resembles stabilizer quantum mechanics, a fragment of quantum theory which is efficiently classically simulable but also non-local. Further analysis of the similarities and differences between those two theories can thus yield new insights into what distinguishes quantum theory from classical theories, and ψ-ontic from ψ-epistemic theories.

In this paper, we develop a graphical language for Spekkens' toy theory. Graphical languages offer intuitive and rigorous formalisms for the analysis of quantum mechanics and similar theories. To compare quantum mechanics and a toy model, it is useful to have similar formalisms for both. We show that our language fully describes Spekkens' toy theory and in particular, that it is complete: meaning any equality that can be derived using other formalisms can also be derived entirely graphically. Our language is inspired by a similar graphical language for quantum mechanics called the ZX-calculus. Thus Spekkens' toy bit theory and stabilizer quantum mechanics can be analysed and compared using analogous graphical formalisms.

1 Introduction

The study of the differences between quantum physical behaviour and classical behaviour is at the heart of much foundational research in quantum physics. The usual way of analysing these differences is by finding phenomena that are intrinsic to one theory and do not appear in the other, for example the violation of Bell inequalities [5]. Yet there is also another approach: that of building toy models which reproduce phenomena generally considered quantum even though their description is entirely rooted in classical physics.

One such toy model is Spekkens' toy theory, first introduced in [25]. The original description of the model was informal, but it was put into equational form in [12]. Furthermore, the toy model was redefined in a rigorous way in [26]; this redefinition is consistent with [12]. Despite being a local hidden variable theory, Spekkens' toy theory reproduces many features of quantum mechanics, e.g. incompatibility of certain observables, teleportation, and no-cloning. The toy theory for the simplest kind of system – the toy bit theory – closely resembles the theory of stabilizer quantum mechanics [25, 21]. Stabilizer quantum mechanics is a fragment of quantum theory resulting from a restriction of the allowed operations to preparation of states in the computational basis, unitary Clifford operations, and computational basis measurements [16].

There are also some phenomena that appear in stabilizer quantum theory but are not replicated in the toy model, e.g. the above-mentioned violation of Bell inequalities. This is related to the fact that

Spekkens' toy theory is a ψ-epistemic theory by construction. A ψ-epistemic theory is a theory where the state that an observer assigns to a system, is not real: it is only an artefact of the restricted knowledge of the observer. Quantum theory on the other hand is considered to be ψ-ontic, i.e. it is a theory where the states an observer assigns to a system are real [22].

To compare and contrast the two theories – Spekkens' toy bit theory and stabilizer quantum theory – in more detail, it is useful to have similar mathematical formalisms for describing both. Particularly, what are needed are *high-level* formalisms, which hide some of the details of the underlying theories so as to focus on conceptual properties. Graphical languages are high-level languages that use two-dimensional diagrams. These two-dimensional languages allow *parallel composition* – applying transformations to two different systems at the same time – to be separated from *sequential composition* – the application of transformations to the same system at different times – by designating one dimension to roughly correspond to "space" and the other to "time". A major difference between classical physics and quantum physics is the way the state spaces of systems compose in parallel, i.e. when the systems are put "side by side" [3]: classically, the resulting state space is the Cartesian product of the original spaces, meaning each state of the joint system can be described by specifying separate states for each of the component systems. For quantum systems, on the other hand, the joint state space is the tensor product of the original state spaces and joint states may not correspond to well-defined states of the separate systems: they can be entangled. Thus in the study of quantum foundations, the study of composite systems is central, and graphical languages offer an intuitive way of doing that.

Spekkens' toy bit theory and stabilizer quantum theory have previously been studied using the stabilizer formalism [21]. The two theories have also been compared using methods from categorical quantum mechanics [10]. Categorical quantum mechanics is the analysis of quantum mechanics via the mathematical formalisms of category theory, pioneered by Abramsky and Coecke [2]. While categorical quantum mechanics has given rise to a range of graphical languages for quantum theory, Spekkens' toy theory has not been analysed graphically before. The ZX-calculus is one such graphical language based on categorical quantum mechanics [8, 9]; we use it as inspiration for the construction of a graphical calculus for the toy theory.

For a graphical formalism to capture a model fully, it needs to satisfy several properties. Firstly, it should be *universal*, i.e. it should be possible to represent graphically any process allowed in the model. The formalism should furthermore be *sound*, i.e. any equality that can be derived graphically should be derivable using other standard formalisms for the model. Lastly, the graphical formalism should be *complete*, meaning that any equality that can be derived using other standard formalisms can also be derived graphically. The ZX-calculus is universal, sound [9], and complete [4] for pure state qubit stabilizer quantum mechanics with post-selected measurements. We show that our graphical calculus satisfies all three of these properties for the maximal knowledge fragment of the toy theory (which corresponds to pure states in quantum theory) with post-selected measurements.

We introduce Spekkens' toy theory and stabilizer quantum mechanics in section 2. Section 3 contains an overview over graphical languages and how to make them rigorous. We then define the graphical calculus for Spekkens' toy bit theory in section 4 and prove that it is universal and sound. The completeness proof for the graphical calculus is given in section 5, with conclusions in section 6.

2   Spekkens' toy theory and stabilizer quantum mechanics

In this section we introduce Spekkens' toy bit theory as well as stabilizer quantum mechanics. We also present some of the standard formalisms used to analyse the two theories.
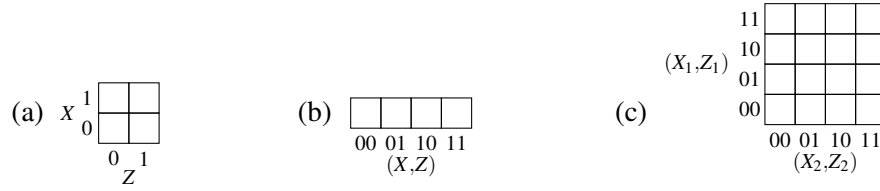
Figure 1: (a) & (b) Visualisations of the state space of a single toy bit. (c) Visualisation of the joint state space of two toy bits. Specific states can be represented by colouring in cells in the diagram.

2.1 Definition of Spekkens' toy bit theory

Spekkens' toy theory is a local hidden variable theory that nevertheless displays many of the same properties and effects as quantum mechanics [25]. It arises from classical probabilistic mechanics via an epistemic restriction, i.e. a restriction on the knowledge an observer may have about the state of the system [26].

We shall only consider the toy theory for the simplest non-trivial system here: the toy bit theory. A single toy bit is a system with four states, these are the *ontic states* or states of reality. These states are often drawn on a 2 by 2 grid as in Fig. 1 a. An ontic state can be described by giving the values – 0 or 1 – for two variables X and Z.

An observer or experimenter working with toy bits does not have direct access to the ontic states, instead they assign to a system an *epistemic state*, a state of knowledge. The observer can learn about the state of a system by measuring *quadrature variables*, which are linear combinations of the variables X and Z – for a single toy bit, these are X, Z, or $X \oplus Z$, where $\oplus$ denotes addition modulo 2. As in quantum mechanics, the quadrature variables X and Z for the same toy bit are considered non-commuting: this is done by imposing a commutation relation $[\cdot, \cdot]$ satisfying:

$$[X,X] = 0 = [Z,Z] \qquad \text{and} \qquad [X,Z] = 1 = [Z,X], \tag{1}$$

which is furthermore linear, so that e.g.:

$$[X \oplus Z, Z] = 1. \tag{2}$$

Now the knowledge an observer may have is determined by the *principle of classical complementarity* [26]:

> The valid epistemic states are those where an agent knows the values of a set of commuting quadrature variables and is ignorant otherwise.

States of maximal knowledge are those epistemic states where the observer knows the values of a maximal set of commuting quadrature variables, i.e. a set with which no other quadrature variable commutes. These states correspond to pure states in quantum theory, and we will only consider states of maximal knowledge in this paper.

There are six states of maximal knowledge of a single toy bit. Single-toy bit states can be visualised on the 2 by 2 grid by colouring in those ontic states that are consistent with the knowledge of the observer. For example, the state $X \oplus Z = 0$ is shown in Fig. 2 a.

Multiple toy bits can be considered jointly, in which case the variables X and Z for separate subsystems are considered to commute, i.e.:

$$[X_i, Z_j] = \delta_{ij}, \tag{3}$$

Figure 2: (a) The single-toy bit state characterised by $X \oplus Z = 0$. (b) The joint state of two toy bits corresponding to $Z_1 = 1 \wedge X_2 = 0$.

where the subscripts denote the subsystem to which the variable belongs and δ_{ij} is 1 if $i = j$ and 0 otherwise. Thus, e.g. $Z_1 = 1 \wedge X_2 = 0$ is an example of a valid joint state of two toy bits. Joint states of two toy bits can be visualised by stretching the diagram for one bit into a line (see Fig. 1 b) and then combining two such lines into a 4 by 4 grid as shown in Fig. 1 c. The above-mentioned state is drawn in Fig. 2 b.

The valid reversible transformations in the toy theory are those permutations of the ontic states that map all valid epistemic states to valid epistemic states. Any set of commuting quadrature variables is a valid measurement.

As a different notation, the ontic states of the toy theory are sometimes numbered 1 through 4 in the order in which they appear in Fig. 1 b [11]. Epistemic states can then be denoted by sets, e.g. the state $X \oplus Z = 0$ corresponds to the set $\{1,4\}$ and the two-toy bit state $Z_1 = 1 \wedge X_2 = 0$ corresponds to:

$$\{(2,1),(2,2),(4,1),(4,2)\}. \tag{4}$$

Rather than considering states of n toy bits to be sets, they can also be seen as relations from the one-element set $I = \{\bullet\}$ into IV^n, the n-fold Cartesian product of the set $IV = \{1,2,3,4\}$. Formally, a relation R between sets A and B is a subset of the Cartesian product $A \times B$. We use the notation $a \sim b$ to indicate that $(a,b) \in R$. Thus, the state $X \oplus Z = 0$ shown in Fig. 2 a corresponds to the relation:

$$\bullet \sim \{1,4\}. \tag{5}$$

Similarly, post-selected measurements on n toy bits can be seen as relations from IV^n to I, e.g. the single-toy bit measurement of the Z variable with outcome 1 corresponds to:

$$\{2,4\} \sim \bullet. \tag{6}$$

Reversible transformations can also be considered as relations. This perspective puts state preparation and post-selected measurements on an equal footing with reversible transformations and allows any process on toy bits to be considered as a relation.

In a slight abuse of notation, we use states and states-as-relations interchangeably.

2.2 Stabilizer quantum mechanics

Stabilizer quantum mechanics is a restriction of the full quantum theory. It includes only those states that are simultaneous eigenstates of several tensor products of Pauli operators, and unitary transformations that map this set of states back to itself. This theory was first introduced in the context of error-correcting codes [16].

In general, 2^n complex numbers are required to specify a quantum state on n qubits – these can be, for example, the components of the vector describing the state in terms of the computational basis. For stabilizer states, there exists a more efficient description by specifying a generating set for the group of

Pauli products that *stabilizes* the state, i.e. maps it back to itself. As an example, consider the Bell state:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{7}$$

which is a stabilizer state. It is stabilized by the following group of Pauli products:

$$\{I \otimes I, X \otimes X, Z \otimes Z, -Y \otimes Y\}, \tag{8}$$

which is generated e.g. by:

$$\langle X \otimes X, Z \otimes Z \rangle. \tag{9}$$

There exists a representation of pure n-qubit stabilizer states in terms of $2n$ by n binary matrices called *check matrix* [6]. Each column in the matrix corresponds to one of the Pauli products generating the stabilizer group, ignoring the factor of ± 1. A single Pauli matrix is encoded in two bits as follows:

$$I \mapsto 00, \tag{10}$$
$$X \mapsto 01, \tag{11}$$
$$Y \mapsto 11, \text{ and} \tag{12}$$
$$Z \mapsto 10. \tag{13}$$

For a Pauli product on n qubits, the m-th factor in the tensor product is represented by the m-th and $(m+n)$-th components of the vector. Thus the generating set from (9) yields the following check matrix:

$$\begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}, \tag{14}$$

where the first column represents $X \otimes X$ and the second column $Z \otimes Z$.

The pure state qubit stabilizer fragment can also be defined operationally in terms of the following transformations [20]:

- preparation of states in the computational basis

- unitary Clifford operations, generated by the single-qubit Hadamard operator, H, and the phase operator, S:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},$$

  as well as the two-qubit controlled-NOT operator:

$$C_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{15}$$

- measurements in the computational basis.

We make use of both the operational description and the binary formalism in later parts of this paper.
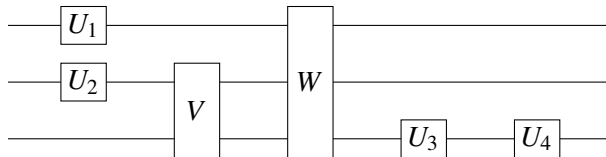
Figure 3: A quantum circuit diagram on three qubits. The operators U_1, U_2, U_3, and U_4 are single-qubit unitaries, V is a two-qubit unitary, and W a three-qubit unitary.

3   Graphical languages

Graphical languages provide an intuitive and high-level way of reasoning. One of the reasons for this is the fact that they allow parallel and sequential composition to be denoted along two different dimensions. By parallel and sequential composition, we mean that there are two different ways of composing transformations: Parallel composition corresponds, roughly, to applying transformations to different (e.g. spatially separated) systems at the same time. Sequential composition, on the other hand, corresponds to applying transformations to the same system at different times.

In linear algebraic notation, parallel and sequential composition are usually distinguished by different operator symbols. For example, the parallel composition of two matrices A and B is the tensor product, denoted by $A \otimes B$. The sequential composition of two matrices A and B is the matrix product, usually written simply as AB. Long algebraic expressions can thus become difficult to parse: it is not easy to see how different components of the expression compose.

In a graphical language, parallel composition can be denoted by stacking symbols vertically, and sequential composition by juxtaposing them horizontally (or conversely, depending on convention). An example of such a graphical notation is the quantum circuit notation [14, 20]. With a two-dimensional notation, it is much easier to see how components compose, even in long and complicated expressions. For example, the quantum circuit in Fig. 3 can be written algebraically as:

$$(I \otimes I \otimes (U_4 U_3)) W (I \otimes V)(U_1 \otimes U_2 \otimes I), \tag{16}$$

where I denotes the single-qubit identity operator. The diagram is much easier to take in at a glance than the algebraic expression.

Yet graphical languages are often introduced as informal personal short-hands and used to develop an intuitive understanding of a problem that can then be confirmed using a more rigorous but less intuitive language. This means doing the same work twice: once graphically, then again in the alternative formalism. To avoid this, the graphical languages need to be made rigorous; then reliable results can be derived entirely graphically.

3.1   Making graphical languages rigorous

There are two steps to the process of making graphical languages rigorous: firstly, one needs to give an explicit translation between diagrams and algebraic terms. Secondly, one needs to prove that two diagrams that seem intuitively equal translate to algebraic terms that are equal.

We are here considering languages that are similar to quantum circuits in that they consist of boxes, which denote transformations, and wires, which correspond to systems or the identity transformations on those systems.
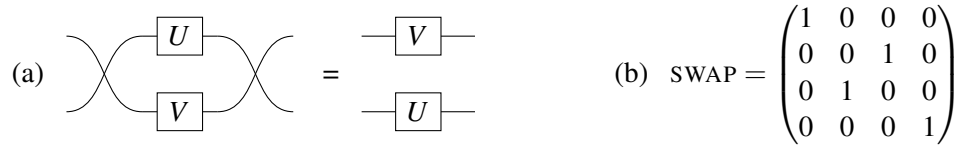
$$\text{(a)} \qquad \text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 4: (a) A quantum circuit equality where a wire crossing denotes the SWAP map and U, V are arbitrary single-qubit unitaries. (b) The SWAP matrix.

Figure 5: (a) & (b) The cap and cup denoting a compact structure. (c) & (d) The snake equations satisfied by caps and cups. Note that this graphical notation is read from bottom to top rather than left-to-right.

The natural formalism for making graphical languages rigorous is category theory, as *monoidal categories* are the most general mathematical structures incorporating both parallel and sequential composition of transformations. This research programme was begun by Joyal and Street, who used the theory of monoidal categories to give rigorous underpinnings to a range of graphical notations from Feynman diagrams to Petri Nets [17]. An introduction to category theory aimed at physicists, can be found in [7]; the standard textbook is [18].

There are other types of categories with additional structure which can be used to model more complicated graphical languages. For example, *symmetric monoidal categories* include a map that swaps two systems that have been composed in parallel. These categories can be used to model graphical languages with wire crossings where "wires don't tangle": i.e. two wire crossings are the same as no crossing at all. An example of this are quantum circuits, where wire crossings are used to denote the transformation interchanging two qubits, for example the circuit shown in Fig. 4.

Quantum mechanics and similar theories are modelled as *dagger compact closed categories* [2]. These are categories which have a dagger map – a generalisation of the Hermitian adjoint for linear maps – and a compact structure, which corresponds to completely entangled states and measurement effects. Graphically, the compact structure is denoted by curved wires – "cups" and "caps" in a language which is read from bottom to top – which satisfy the *snake equations* shown in Fig. 5. In quantum theory, the "cup" can be thought of as the preparation of a completely entangled state on two systems; the "cap" is the outcome of finding that same state when doing a joint measurement of two systems. The snake equations thus correspond to a post-selected formulation of quantum teleportation.

In graphical languages based on dagger compact closed categories, two diagrams represent the same map if they are equal up to topological transformations that keep the inputs and outputs of the diagrams as a whole invariant [24]. Topological transformations here are operations like lengthening or shortening wires, bending or straightening wires, or moving boxes around while keeping their connections the same. Both the equality in Fig. 4 and the snake equations in Fig. 5 are examples of such topological transformations.

Yet topological transformations are not enough to yield all desired equalities between diagrams: e.g. the diagram equality in Fig. 6 is true, but it is not a topological transformation, and cannot be made into one by a change of notation either. By assuming a set of axiomatic equalities called *rewrite rules*, further graphical equalities can be derived using graphical rewriting. The idea is that whenever two diagrams are equal, any time one of those diagrams appears as part of a larger diagram, it can be "cut out" and replaced by the other diagram. We gloss over the details of this rewrite process here and assume that a
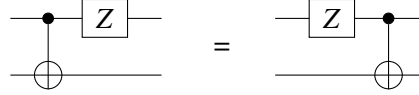
Figure 6: A quantum circuit equality, where Z denotes the Pauli-Z gate and the other symbol is the controlled-NOT gate as defined in (15).
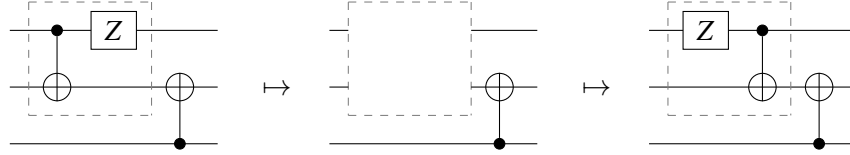

Figure 7: Rewriting a quantum circuit using the rewrite rule given in Fig. 6: the left-hand side of that equality matches part of the diagram here, that part is "cut out", and the right-hand side of Fig. 6 is pasted in.

simple copy-paste process works. An example using quantum circuits is given in Fig. 6 and Fig. 7.

This process of introducing rewrite rules is fairly arbitrary, posing the question of which equalities should become rewrite rules and how many rewrite rules are necessary. That issue can be simplified by considering the graphical calculus as a formal system.

3.2   Graphical calculi as formal systems

A graphical calculus with rewrite rules can be considered as a formal system that allows the derivation of equalities from the axioms given by the rewrite rules. For such a calculus to be a useful alternative to a more standard algebraic language, it needs to satisfy certain properties, which are all relative to the interpretation of the diagrams. Given a diagram D, we denote by $[\![D]\!]$ its interpretation, i.e. the process corresponding to D. The desired properties of a graphical language include universality, soundness, and completeness, which are defined as follows:

- A graphical calculus for a theory is *universal* if any state or transformation allowed by the theory can be represented graphically, i.e. for any process P there exists a diagram D such that:

$$[\![D]\!] = P. \tag{17}$$

- It is *sound* if any equality that can be derived graphically can also be derived in the underlying theory, i.e. if for any two diagrams D_1 and D_2:

$$D_1 = D_2 \implies [\![D_1]\!] = [\![D_2]\!]. \tag{18}$$

- A graphical calculus is *complete* if any equality that can be derived in the underlying theory can also be derived graphically, i.e. if for any two diagrams D_1 and D_2:

$$[\![D_1]\!] = [\![D_2]\!] \implies D_1 = D_2. \tag{19}$$

Universality and soundness are usually straightforward to determine: The property of universality does not involve the rewrite rules at all, it only depends on the diagram components and their interpretations. Soundness does depend on the rewrite rules, but it can be checked on a rule-by-rule basis: if each rule is sound, then the graphical calculus as a whole is sound. Completeness is more difficult to prove because it relies on the interaction of all the rewrite rules.

4 A graphical calculus for Spekkens' toy bit theory

In this section, we construct a graphical calculus for the maximal knowledge fragment of Spekkens' toy bit theory with post-selected measurements. This graphical calculus is modelled after the ZX-calculus for quantum mechanics [8]. We define basic elements of the graphical notation and show how to combine them into more complicated diagrams. Next, we give rewrite rules for those diagrams. We argue that the calculus is universal and sound for Spekkens' toy bit theory. Finally, we compare the graphical calculus for the toy theory to the ZX-calculus.

4.1 Components of the toy theory graphical calculus

The graphical calculus for the toy theory is read from bottom to top. Rather than denoting maps by labelled boxes, most maps are denoted by circular nodes which may have labels attached. As before, we use $[\![D]\!]$ to denote the process corresponding to a diagram D.

Define $\bigvee$ to be the following map from one toy bit to two toy bits:

$$\left[\!\!\left[\bigvee\right]\!\!\right] := \begin{cases} 1 \sim \{(1,1),(2,2)\} \\ 2 \sim \{(1,2),(2,1)\} \\ 3 \sim \{(3,3),(4,4)\} \\ 4 \sim \{(3,4),(4,3)\}. \end{cases} \tag{20}$$

This is a valid process in the toy theory; it can be considered to consist of the preparation of an ancilla in some fixed state followed by some joint reversible operation on the original toy bit and the ancilla.

The *converse* of a relation R, denoted by $R^\dagger$, is defined as:

$$R^\dagger = \{(b,a) \mid (a,b) \in R\}. \tag{21}$$

Let $\bigwedge$ be the converse of $\bigvee$:

$$\left[\!\!\left[\bigwedge\right]\!\!\right] := \left[\!\!\left[\bigvee\right]\!\!\right]^\dagger. \tag{22}$$

This is also a valid process in the toy theory, which can be thought of as a reversible operation on two toy bits, followed by a post-selected measurement of one of them. As indicated by this notation, the relational converse is the toy theory equivalent of the Hermitian adjoint [11].

More complicated diagrams in the toy theory graphical calculus can be built by putting smaller diagrams side-by-side, which corresponds to taking the Cartesian product of the corresponding relations; i.e. if:

$$\boxed{D} \quad \text{and} \quad \boxed{D'}$$

denote two arbitrary diagrams, then:

$$\left[\!\!\left[\boxed{D}\ \boxed{D'}\right]\!\!\right] = \left[\!\!\left[\boxed{D}\right]\!\!\right] \times \left[\!\!\left[\boxed{D'}\right]\!\!\right]. \tag{23}$$

Connecting the inputs of some diagram to the outputs of another corresponds to the operation of relational composition: if $R : A \to B$ and $S : B \to C$ are two relations, their composite $S \circ R$ is:

$$S \circ R = \{(a,c) \mid \exists b \in B \text{ s.t. } (a,b) \in R \wedge (b,c) \in S\}. \tag{24}$$

Graphically, assuming the number of outputs of D is equal to the number of inputs of D':

$$\left[\!\!\left[\boxed{\begin{array}{c} \cdots \\ \boxed{D'} \\ \cdots \\ \boxed{D} \\ \cdots \end{array}} \right]\!\!\right] = \left[\!\!\left[\boxed{\begin{array}{c} \cdots \\ D' \\ \cdots \end{array}} \right]\!\!\right] \circ \left[\!\!\left[\boxed{\begin{array}{c} \cdots \\ D \\ \cdots \end{array}} \right]\!\!\right] . \tag{25}$$

We introduce a short-hand notation for specific diagrams built from and : a green node with n inputs and m outputs for positive integers n, m is defined as follows:

$$\tag{26}$$

Such a node is called a *spider*.

Represent the following four single-toy bit states by green nodes with *phase labels*:

$$\left[\!\!\left[\boxed{\bullet\, 00} \right]\!\!\right] := \{1, 3\}, \tag{27}$$

$$\left[\!\!\left[\boxed{\bullet\, 01} \right]\!\!\right] := \{1, 4\}, \tag{28}$$

$$\left[\!\!\left[\boxed{\bullet\, 10} \right]\!\!\right] := \{2, 3\}, \quad \text{and} \tag{29}$$

$$\left[\!\!\left[\boxed{\bullet\, 11} \right]\!\!\right] := \{2, 4\}, \tag{30}$$

and let $\bullet$ be short-hand for $\boxed{\bullet\, 00}$. These two alternative notations make later definitions consistent. Spiders can now be given phase labels via the following definition:

$$\tag{31}$$

where $x, y \in \{0, 1\}$. Furthermore, spiders without inputs can be defined by composing $\bullet$ and a spider with one input. Let $\circ$ be the converse of $\bullet$, seen as a relation:

$$\left[\!\!\left[\bullet \right]\!\!\right] := \begin{cases} 1 \sim \bullet \\ 3 \sim \bullet. \end{cases} \tag{32}$$

Then arbitrary spiders with no outputs can be defined as composites of a one-output spider and $\circ$. In this way, definitions (26) and (31) can be extended to arbitrary non-negative numbers of inputs and outputs n and m.

Let $\square$ be the following reversible single-toy bit operation:

$$\left[\!\!\left[\boxed{} \right]\!\!\right] := \begin{cases} 1 \sim 1 \\ 2 \sim 3 \\ 3 \sim 2 \\ 4 \sim 4. \end{cases} \tag{33}$$

As a final short-hand, define red spiders as green spiders with copies of ▯ on all inputs and outputs:

$$xy := xy \tag{34}$$

A single straight wire corresponds to the identity relation and a wire crossing is the obvious SWAP relation interchanging the states of the two subsystems. A "cup" is interpreted as follows:

$$[\![\smile]\!] = \{(1,1),(2,2),(3,3),(4,4)\}, \tag{35}$$

and the cap is its converse.

As , , xy , and are all special cases of green spiders, the graphical calculus can be considered to consist of green phased spiders with n inputs and m outputs, where n and m are now non-negative integers; red phased spiders with arbitrary numbers of inputs and outputs; and ▯.

Spiders with exactly one input and one output are called *phase shifts*, these are the only spiders representing reversible operations. A diagram with neither inputs nor outputs is called a *scalar diagram*; the same terminology is also used for parts of larger diagrams that are disconnected from all inputs or outputs of the large diagram.

4.2 Rewrite rules of the toy theory graphical calculus

We postulate the following rewrite rules for the toy theory graphical calculus. Any rule given here can also be used with the colours red and green swapped. Rules can furthermore be used upside-down.

Spider rule and loop rule: Two nodes of the same colour can merge if they are connected by an edge, in that case their phase labels combine by bit-wise addition modulo 2. Self-loops can be removed.

$$ab \quad cd \ = \ (a \oplus c)(b \oplus d) \qquad\qquad ab \ = \ ab$$

Identity rule, bialgebra rule, and copy rule: A node with one input and one output and no phase label (or, equivalently, phase 00) is the same as an edge. The bialgebra rule allows a certain pattern of two red and two green nodes to be replaced by just one red and green node. A node of one colour with one input and two outputs copies the zero phase state of the other colour.

$$= | \qquad\qquad = \qquad\qquad =$$

11-copy rule and 11-commutation rule: A 11-phase shift is copied by a node of the other colour. It can also be moved past any phase shift of the other colour, swapping the two bits of that phase label in the process.

$$= \ 11 \ \cdots \ 11 \qquad\qquad \begin{matrix} 11 \\ ab \end{matrix} \ = \ \begin{matrix} ba \\ 11 \end{matrix}$$

Colour change rule and Euler decomposition of ▯: The ▯ node swaps the colour of red and green nodes when it is applied to each input and output. Furthermore, ▯ can be replaced by three green and red nodes of alternating colours, each with phase 01. This rule is called Euler decomposition in analogy to the Euler decomposition of general rotations in three-dimensional space into three rotations about two distinct axes.
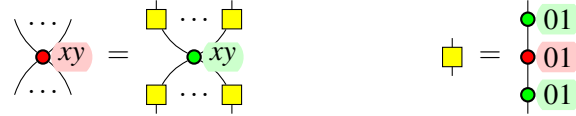
Whenever a rule holds for any number of edges, that number may be zero. For example, the colour change rule with zero inputs and zero outputs implies that $\bullet\, xy = \circ\, xy$ for any $x, y \in \{0, 1\}$.

There are also two meta rules, rules that are not specified in terms of diagram equalities. The first of these is that "only the topology matters". This means that two diagrams represent the same process whenever they contain the same set of nodes connected up in the same ways, no matter how those nodes are arranged on the plane. Secondly, we "ignore all scalars that do not represent the empty relation", meaning scalar subdiagrams can simply be dropped as long as they do not represent the empty relation.

4.3 Universality of the graphical calculus

The graphical calculus for the toy theory as defined in the previous two subsections is universal for the maximal knowledge fragment of Spekkens' toy bit theory with post-selected measurements. This follows from the category-theoretical formulation of the toy theory in [11], where it is shown that all processes in the toy theory arise – via parallel and sequential composition, and taking the relational converse – from the 24 reversible transformations of a single toy bit together with a specific map called δ from one toy bit to two toy bits, and a post-selected measurement outcome denoted ε. It is straightforward to see that $\square$ and the phase shifts suffice to construct all 24 reversible single-toy bit transformations, which correspond to the 24 permutations of the ontic states. The maps δ and ε from [11] are exactly the maps denoted by $\curlyvee$ and $\circ$ here. The graphical calculus allows parallel and sequential composition, as well as the taking of relational converses, which corresponds to flipping diagrams upside-down. Therefore any process in the maximal knowledge fragment of Spekkens' toy bit theory with post-selected measurements can be represented graphically.

4.4 Soundness of the graphical calculus

Most of the rewrite rules of the toy theory graphical calculus can straightforwardly be checked to be sound by translating the diagrams on both sides of the equality into the corresponding maps. For example, the left-hand side of the copy rule is, by definition, equal to:
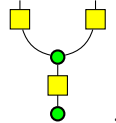
Using the definitions of $\circ$, $\square$, and $\curlyvee$, this diagram can be translated to the state:

$$\{(1,1), (1,3), (3,1), (3,3)\}. \tag{36}$$

The right-hand side of the copy rule translates to:

$$\{1,3\} \times \{1,3\}. \tag{37}$$

By explicitly constructing the Cartesian product in (37), the two relations are seen to be the same. Therefore the copy rule is sound.

The other rewrite rules with fixed numbers of inputs and outputs – i.e. the identity rule, bialgebra rule, 11-commutation rule, and Euler decomposition rule – can be checked in the same way. The colour

change rule is sound by definition. Soundness of both the 11-copy rule and the loop rule can be verified by induction over the number of outputs and/or inputs.

For soundness of the spider rule, we again rely on results from the categorical formulation of Spekkens' toy bit theory. As shown in [12], the maps $\curlyvee$ and $\bullet$ form a category-theoretical *observable*. This means that any connected diagram constructed from these maps, their converses, wire crossings, and curved wires is determined solely by its number of inputs and outputs. Graphically, this corresponds exactly to the spider law without phase labels [13]. The states $\bullet\, xy$ form a *phase group* for this observable [12]. In particular, they form a group under the operation given by composition with $\curlywedge$:

$$ \vcenter{\hbox{$\begin{array}{c}\end{array}$}}\; ab\;\bullet\;\curlyvee\;\bullet\; cd \quad = \quad \bullet\,(a \oplus c)(b \oplus d) \;, \tag{38} $$

with group identity $\bullet$ and all group elements being self-inverse. From this, it follows that the spider law with phase labels is also sound.

Soundness of the topology meta-rule also follows from the category-theoretical formulation of the toy theory: The toy theory is modelled as a dagger compact closed category, therefore the results given in section 3.1 apply.

The ignore-scalars rule is also sound by the category-theoretical formulation. There are exactly two relations from I to I, the identity relation $\{(\bullet, \bullet)\}$ and the empty relation $\emptyset$. Composing any relation with $\{(\bullet, \bullet)\}$ does not change the relation. Thus dropping scalar subdiagrams is justified, as long as they are not the empty relation.

4.5   The toy theory graphical calculus and the ZX-calculus

The toy theory graphical calculus is modelled after the ZX-calculus for pure state stabilizer quantum mechanics with post-selected measurements, which also consists of green and red phased spiders and yellow nodes that change the colour of spiders [8, 9].

The ZX-calculus arises from the categorical formulation of quantum mechanics, and as explained in the previous section, the toy theory graphical calculus is closely related to the categorical formulation of Spekkens' toy bit theory. Category-theoretically, the only difference between the toy theory and stabilizer quantum theory is the phase group of the respective observables: for the toy theory the phase group is isomorphic to the Klein Four group $\mathbb{Z}_2 \times \mathbb{Z}_2$, whereas for stabilizer quantum theory the phase group is isomorphic to the cyclic group of order 4, $\mathbb{Z}_4$ [12].

Correspondingly, the rewrite rules of the toy theory graphical calculus that do not involve specific phases are exactly the same as those of the ZX-calculus if the phase groups are swapped out. In the stabilizer ZX-calculus, the phase group is generally denoted by the angles $\{-\pi/2, 0, \pi/2, \pi\}$ under addition modulo 2π. The Euler decomposition rule and 11-copy rules also have straightforward analogues in the ZX-calculus, with $\pi/2$ phase shifts appearing in the Euler decomposition and π taking the role of 11.

The ZX-calculus analogue to the 11-commutation rule is the π-commutation rule:

$$ \begin{array}{c}\bullet\,\pi \\ \bullet\,\alpha\end{array} \quad = \quad \begin{array}{c}\bullet\,-\alpha \\ \bullet\,\pi\end{array}. \tag{39} $$

At first glance this looks different to the 11-commutation rule: the π-commutation rule sends any phase shift to its inverse whereas the 11-commutation rule swaps the two bits denoting the phase. In fact, both commutation rules can be expressed in the same way nevertheless. Let φ denote 11 or π and let θ be an

arbitrary phase label for the respective theory. We can write the two commutation rules in general form as:

$$
\begin{array}{c} \phi \\ \theta \end{array} = \begin{array}{c} f(\theta) \\ \phi \end{array}, \tag{40}
$$

where f is some map from the phase group to itself. Then in both the ZX-calculus for stabilizer quantum mechanics and in the graphical calculus for Spekkens' toy bit theory, the map f can be characterised as follows: f maps both φ and the identity of the phase group back to themselves, but it swaps the remaining two elements of the phase group.

5 Completeness of the graphical calculus

We now show that the toy theory graphical calculus is complete by adapting the completeness proof for the stabilizer ZX-calculus [4]. There are several parts to the argument: First, we show that the results characterising all true equalities between stabilizer states from [19], which are central to the ZX-calculus completeness proof, also hold in Spekkens' toy theory. We then argue that it is sufficient to consider equalities between toy states rather than more general processes in the toy theory, because the toy theory has map-state duality. Next, we prove that diagrams in the toy theory graphical calculus can be brought into a normal form called GS-LO form. Finally, we show that the rewriting strategies used in the ZX-calculus completeness proof also work in the toy theory graphical calculus.

Where the steps in the completeness argument for the toy theory differ only marginally from the corresponding steps in the stabilizer ZX-calculus completeness proof, the proofs are left out or given in sketch form. Longer proofs can be found in the appendices.

5.1 The binary stabilizer formalism and the graph state theorems

Completeness of a graphical language means that any equality that can be derived in the standard formalism for the same theory can also be derived graphically. Thus it is useful to have some simple way of characterising the equalities that can be derived in the underlying theory.

The completeness proof for the stabilizer ZX-calculus makes use of two theorems about relationships between stabilizer states under local Clifford unitaries, i.e. unitary stabilizer operations that are tensor products of single-qubit unitaries. Central to these results are graph states – a class of stabilizer states whose entanglement structure is that of a finite simple graph, i.e. a graph with finitely many vertices (corresponding to the qubits), at most one edge between each pair of vertices (corresponding to the entanglement), and no self-loops. Graph states on n qubits can be represented in the binary formalism as follows [19]:

$$
S = \begin{pmatrix} \theta \\ I \end{pmatrix}, \tag{41}
$$

where θ is a n by n symmetric matrix with zeroes along the diagonal and I is the n by n identity matrix. The matrix θ is in fact the adjacency matrix of the underlying graph: a 1 in position (p,q) means that the p-th and q-th vertices are connected by an edge, a 0 means they are not connected. The following results can be proved using the binary formalism:

Theorem 1 ([19])**.** *Any stabilizer state can be transformed into some graph state by application of a local Clifford operation.*

Theorem 2 ([19])**.** *Two stabilizer states are equivalent under local Clifford operations if and only if the underlying graphs are related by a sequence of local complementations.*

By local complementation we mean the following operation on a graph.

Definition 1. Let $G = (V,E)$ be a graph with set of vertices V and set of edges E. The *local complementation about the vertex v* is the operation that inverts the subgraph generated by the neighbourhood of v (but not including v itself). Formally, a local complementation about $v \in V$ sends G to the graph:

$$G \star v = \left(V, E \triangle \{\{b,c\} | \{b,v\}, \{c,v\} \in E \wedge b \neq c\}\right), \tag{42}$$

where $\triangle$ denotes the symmetric set difference, i.e. $A \triangle B$ contains all elements that are contained either in A or in B but not in both.

We now show that these results translate to the toy theory.

As described in section 2.1, a state of maximal knowledge on n toy bits is given by a set of n commuting quadrature variables, together with the values for each of the variables. These quadrature variables can be represented as binary vectors, similar to the representation of Pauli products, where the m-th and $(m+n)$-th component together encode the quadrature variable acting on the m-th toy bit according to the following encoding:

$$X \mapsto 01, \tag{43}$$
$$Z \mapsto 10, \text{ and} \tag{44}$$
$$X \oplus Z \mapsto 11, \tag{45}$$

with 00 indicating that no quadrature variable is acting on the given toy bit. Thus, ignoring the values of the quadrature variables, any state of maximal knowledge can be described by a binary $2n$ by n matrix in the same way as a pure quantum state.

Lemma 3. *A binary $2n$ by n matrix S represents a valid state in the toy theory if and only if $S^T J S = 0$, where:*

$$J = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} \tag{46}$$

with I the n by n identity matrix.

This follows from the principle of classical complementarity, as shown in [26]. In the binary picture, the valid reversible transformations of the toy theory are represented by $2n$ by $2n$ binary matrices Q satisfying $Q^T J Q = J$.

The conditions for $2n$ by n binary matrices to represent valid states and the condition for $2n$ by $2n$ binary matrices to represent valid transformations are exactly the same as in the binary formalism for stabilizer quantum mechanics. Therefore the binary matrix formalism for Spekkens' toy bit theory is exactly the same as the check matrix formalism for stabilizer quantum mechanics, if one ignores the values of the quadrature variables in the former and the eigenvalues in the latter. An equivalent result was shown in [21], albeit not using check matrices.

When considering graph states and local Clifford transformations in quantum mechanics, it is reasonable to ignore the eigenvalues in the stabilizer formalism because the eigenvalue for each stabilizer of a graph state can be changed by a local Clifford transformation that keeps all the other properties of the state invariant. Define graph states in Spekkens' toy bit theory to be states having the same check matrix representation as some graph state in stabilizer quantum mechanics. Then the values of the quadrature variables can be ignored for the same reason as eigenvalues in quantum theory.

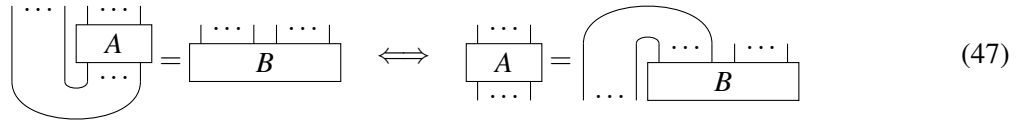Thus theorems 1 and 2 carry over to the toy theory, i.e. we have:

Theorem 4. *Any toy stabilizer state is equivalent to some toy graph state under local toy transformations* $\sigma \in (S_4)^n$.

Theorem 5. *Two toy graph states on the same number of toy bits are equivalent under local toy transformations if and only if there is a sequence of local complementations that transform one graph into other.*

Local complementations on a graph are defined just as above.

5.2 Map-state duality for the toy theory

The graph state theorems, as implied by the name, apply only to toy states, not more general processes in the toy theory. Yet it suffices to consider only equalities between states in order to get a completeness result for the entire theory. This is because the toy theory exhibits map-state duality: as in quantum theory, there exists a bijection between operators from n to m toy bits and states on $(n+m)$ toy bits. This duality is also known as the Choi-Jamiołkowski isomorphism. Denoting the map by A and its corresponding state by B, the isomorphism can be represented diagrammatically as follows:

(47)

This result follows directly from the snake equations given in Fig. 5, which hold in the toy theory graphical calculus via the topology rule.

The Choi-Jamiołkowski isomorphism allows toy theory operators to be turned into states. Any equalities derived between these states then apply also to the original operators. Thus, a completeness result for the entire toy theory can be derived by considering only toy states.

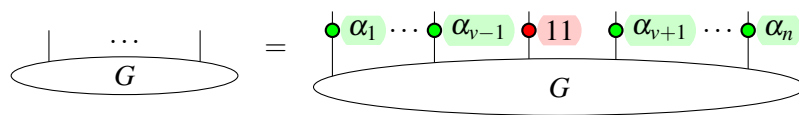## 5.3 Graph states and related diagrams in the toy theory graphical calculus

In the first part of this section, we defined graph states for the toy theory via their check matrices. We now show that they also have an elegant graphical representation.

Definition 2. Let G be a finite simple undirected graph, i.e. a graph with finitely many vertices, at most one edge between any pair of vertices, and no self-loops. Let the set of vertices be V and the set of edges E. The associated graph state in the toy theory graphical calculus comprises the following:

- for each vertex in V, a green node with one output, and
- for each edge in E, a copy of ▢ connected to the green nodes representing the vertices at either end of the edge.

To show that this definition is equivalent to the previous one, we consider the operators stabilizing the graph state.

Lemma 6. *Let* *denote the state associated with a graph G. Then for any vertex* $v \in V$,

where $\alpha_k = 11$ *if* $\{v,k\} \in E$ *and* $\alpha_k = 00$ *otherwise. This means that* *is an eigenstate of any operator that applies* •11 *to one of the vertices and* •11 *to all neighbours of that vertex.*

This lemma follows from the rules of the red-green calculus for the toy theory; the proof is entirely analogous to that for the ZX-calculus, where π-phase shifts appear instead of the 11-phase shifts [15].

Corollary 7. *Definition 2 is equivalent to the definition of toy theory graph states via their check matrices.*

Proof. Lemma 6 gives n quadrature variables of which the diagram is an eigenstate: a phase shift ● 11 on the p-th output corresponds to a term X_p in the quadrature variable, a phase shift ● 11 on the q-th output corresponds to a term Z_q. Translate each of these variables into a binary vector as described in section 5.1, and assemble the resulting vectors as the columns of a matrix with the vector for the variable involving the term X_m as the m-th column. The resulting check matrix then has the form given in (41). □

The local complementation operations from theorem 5 can be derived from the rules of the toy theory graphical calculus. Note that from now on we use the term "local complementation" in a slightly different sense as before: previously, the term referred to an operation on graphs only. From now on, we use the same term to refer to an operation on graph states together with the application of a local operation to all the toy bits that keeps the overall toy state invariant.

Lemma 8. *The following local complementation rule holds in the red-green calculus for the toy theory:*
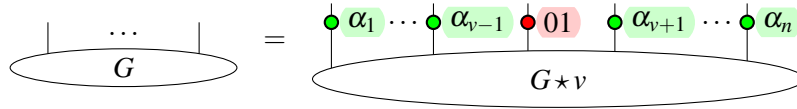
where $\alpha_k = 01$ if $\{v,k\} \in E$ and $\alpha_k = 00$ otherwise, and $G \star v$ denotes the graph-theoretical local complementation as defined in (42).

A sketch proof of this lemma can be found in appendix A.1.

A *local complementation along an edge* $\{v,w\} \in E$ consists of a local complementation about v, a local complementation about w, and another local complementation about v, yielding:
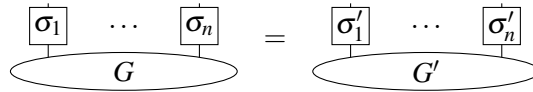
where:

$$\sigma'_j = \begin{cases} \sigma_j \circ (23) & \text{if } j \in \{v,w\} \\ \sigma_j & \text{otherwise} \end{cases}$$

and $G' = (V, E')$ satisfies the following properties:

- $G' = ((G \star v) \star w) \star v = ((G \star w) \star v) \star w$;

- $\{v,w\} \in E'$;

- for $j \in V \setminus \{v,w\}$, $\{j,v\} \in E' \Leftrightarrow \{j,w\} \in E$ and $\{j,w\} \in E' \Leftrightarrow \{j,v\} \in E$, i.e. a vertex j is adjacent to v in G' if and only if j was adjacent to w in G and correspondingly with v and w exchanged;

- for $p,q \in V \setminus \{v,w\}$, let P be the intersection of p's neighbourhood with $\{v,w\}$, i.e. $v \in P$ if $\{p,v\} \in E$ and $w \in P$ if $\{p,w\} \in E$, and define Q correspondingly. Then the edge $\{p,q\}$ is toggled if and only if P, Q and $\emptyset$ are pairwise distinct.

This operation is symmetric under interchange of the two vertices v and w.

It will be useful to have a normal form for reversible single-toy bit operators.

Lemma 9. *Any reversible single-toy bit operator, i.e. any diagram or subdiagram consisting solely of phase shifts and* ⬜*, can be written uniquely in one of the following forms:*
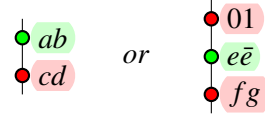
where $a,b,c,d,e,f,g \in \{0,1\}$ *and* $\bar{e} = e \oplus 1$, *with* $\oplus$ *denoting addition modulo 2.*

This is straightforward to check, analogously to the corresponding result in the ZX-calculus. In the following, whenever we talk about reversible single-toy bit operators we will assume that they are normalised as in the above lemma.

Definition 3. A diagram in the red-green calculus for Spekkens' toy theory is called a *GS-LO diagram* (graph state with local operators) if it consists of a graph state as in definition 2 with reversible single-toy bit operators on each output.

GS-LO diagrams play a central role in the graphical calculus for the toy theory, as shown by the following theorem.

Theorem 10. *Any state diagram in the red-green calculus for Spekkens's toy theory is equal to some GS-LO diagram according to the rewrite rules.*

This theorem is analogous to theorem 7 in [4]. A sketch of the parts of the proof that differ from the ZX-calculus case can be found in appendix A.1. The proof relies on the fact that diagrams can be decomposed into reversible single-toy bit operators and four basic spiders:

Local complementations together with fixpoint operations can be used to change the reversible single-toy bit operator on any given toy bit in the graph state to any desired operator, similar to the equivalent result for the ZX-calculus [4]. Thus it can be shown that any diagram consisting of a basic spider composed with a GS-LO diagram can be rewritten into GS-LO form. The only basic spider with no inputs is ●, so any state diagram must contain at least one copy of that; furthermore ● is a GS-LO diagram. Therefore, by induction, any diagram can be brought into GS-LO form.

The GS-LO form is not unique, i.e. there may be different GS-LO diagrams representing the same state. It is not clear how to define a unique normal form, but it is possible to reduce the number of diagrams needing to be considered further.

Definition 4. A diagram in Spekkens's toy theory is said to be in *reduced GS-LO (or rGS-LO) form* if it is in GS-LO form and satisfies the following additional conditions:

- All the reversible single-toy bit operators belong to the set:

$$R = \left\{ \quad \middle| \quad \overset{01}{\bullet} \quad \overset{11}{\bullet} \quad \overset{10}{\bullet} \quad \overset{01}{\underset{01}{\bullet}} \quad \overset{01}{\underset{10}{\bullet}} \right\}. \tag{48}$$

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

Theorem 11. *Any toy stabilizer state diagram is equal to some rGS-LO diagram within the graphical calculus.*

A sketch proof of this theorem can be found in appendix A.1.

Reduced GS-LO diagrams are still not unique, as shown by the following two propositions, sketch proofs of which can be found in the appendix.

Proposition 12. *Suppose a rGS-LO diagram contains a pair of neighbouring toy bits p and q in the following configuration, where $a, b \in \{0, 1\}$:*
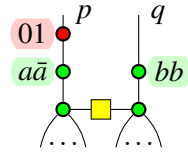
Then a local complementation about q, followed by a local complementation about p, yields a diagram which can be brought into rGS-LO form by at most two applications of the fixpoint rule.

Proposition 13. *Suppose a rGS-LO diagram contains a pair of neighbouring toy bits p and q in the following configuration, where $a, b \in \{0, 1\}$:*
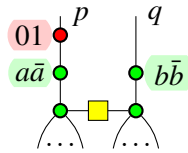
Then a local complementation along the edge $\{p, q\}$ yields a diagram which can be brought into rGS-LO form by at most two applications of the fixpoint rule.

With the definitions and results in this section, state diagrams in the toy theory graphical calculus can be simplified significantly. By map-state duality, the results can be applied to arbitrary diagrams.

For completeness it remains to be shown that whenever two rGS-LO diagrams represent the same toy state, they can be rewritten into each other using the rewrite rules for the toy theory graphical calculus.

5.4 Equalities between rGS-LO diagrams

The graphical calculus is complete for toy theory states if, given any two rGS-LO diagrams representing the same state, we can show that they are equal using the rules of the graphical calculus. In this section, we exhibit an algorithm for rewriting two diagrams representing the same toy state to be identical. As rewrite rules are invertible, this is equivalent to being able to rewrite one diagram into the other. The algorithm is adapted from a similar one for the stabilizer ZX-calculus [4].

Given two toy state diagrams on the same number of toy bits, we start by pairing up red nodes between the two diagrams.

Definition 5. A pair of rGS-LO diagrams on the same number of toy bits is called *simplified* if there are no pairs of toy bits p, q such that p has a red node in its vertex operator in the first diagram but not in the second, q has a red node in the second diagram but not in the first, and p and q are adjacent in at least one of the diagrams.

Proposition 14. *Any pair of rGS-LO diagrams on n toy bits can be simplified.*

The proof of this proposition is analogous to the stabilizer ZX-calculus case in [4]. The idea is to use the equivalence operations of rGS-LO diagrams given in propositions 12 and 13 to shift red nodes to different toy bits in the diagram.

Simplifying a pair of diagrams is not an arbitrary process: for a simplified pair of diagrams, it is straightforward to decide whether or not they are equal according to the rewrite rules. Firstly, if there exist red nodes that cannot be paired up between the two diagrams, then the diagrams cannot represent the same state, as shown by the following lemma.

Lemma 15. *Consider a simplified pair of rGS-LO diagrams and suppose there exists an unpaired red node, i.e. there is a toy bit p which has a red node in its vertex operator in one of the diagrams, but not in the other. Then the two diagrams are not equal.*

This lemma is proved in appendix A.2.

The existence of unpaired red nodes is not the only sign that a simplified pair of diagrams cannot be equal. In fact, a simplified pair of diagrams are either identical or they do not represent the same state.

Theorem 16. *The two diagrams making up a simplified pair of rGS-LO diagram are equal, i.e. they correspond to the same toy theory state, if and only if they are identical.*

The proof of this theorem is analogous to that of theorem 18 in [4].

By map-state duality for the toy theory, as given in (47), and invertibility of the rewrite rules, theorem 16 directly implies:

Theorem 17. *The red-green calculus is complete for Spekkens' toy bit theory.*

Equalities between two diagrams in the toy theory graphical calculus can be derived as follows: if the diagrams are not states, bend all inputs around to become outputs. Bring the two diagrams into GS-LO form and thus into rGS-LO form. Simplify the pair of diagrams. Then either the two diagrams are identical, in which case some of the rewrite steps can be inverted to get a sequence of rewrites transforming one diagram into the other, or they are not identical, in which case the two diagrams do not represent the same operator, so there is no equality to derive.

If the diagrams were not states to begin with, the appropriate outputs can be bent back into inputs in all diagram. This yields a sequence of valid rewrites transforming one of the original diagrams into the other.

6   Conclusions

We have defined a graphical calculus for Spekkens' toy bit theory and shown that it is universal, sound, and complete for the maximal knowledge fragment of the theory with post-selected measurements. This means that the graphical calculus has the full power of any formalism for analysing the toy theory. Our graphical calculus is modelled after the ZX-calculus, a similar universal, sound, and complete graphical calculus for pure state qubit stabilizer quantum mechanics with post-selected measurements. Therefore similarities and differences between stabilizer quantum mechanics and the toy bit theory can be analysed entirely graphically.

A potential next step for this research programme is to implement the rewrite rules and algorithms involved in the completeness proof in the software system *Quantomatic*, which enables automated and semi-automated manipulation of diagrams in the ZX-calculus and similar graphical languages [1]. That way, diagrams can be simplified and equalities between toy theory diagrams can be derived automatically. If the corresponding algorithms for the ZX-calculus are implemented as well, Quantomatic can compare the two theories automatically.

So far, we have only considered pure state qubit stabilizer quantum mechanics and the maximal knowledge fragment of Spekkens' toy bit theory. An obvious next step would be to extend the graphical calculi to mixed states in the quantum case and states of less-than-maximal knowledge in the toy theory. The category-theoretical formulations underlying the graphical calculi can easily be extended in this way using the CPM-construction [24] and these extensions carry over to categorical graphical calculi.

Furthermore, it would be interesting to extend this argument to stabilizer quantum mechanics for higher dimensional systems and the higher-dimensional toy theory. Some steps in this direction have

been made by generalising the zx-calculus to qudits and Spekkens' toy theory for systems of dimension greater than two, though it is still unclear whether these graphical languages are complete [23].

Rigorous graphical languages have many applications in the analysis of quantum physics and related theories.

Acknowledgements

References

[1] *Quantomatic*. `https://quantomatic.github.io/`. Accessed August 2015.

[2] Samson Abramsky & Bob Coecke (2004): *A categorical semantics of quantum protocols*. In: *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS'04)*, pp. 415 – 425, doi:10.1109/LICS.2004.1319636.

[3] Samson Abramsky & Bob Coecke (2008): *Categorical quantum mechanics*. In: *Handbook of quantum logic and quantum structures: quantum logic*, Elsevier, pp. 261–324, doi:10.1016/B978-0-444-52869-8.50010-4.

[4] Miriam Backens (2014): *The ZX-calculus is complete for stabilizer quantum mechanics*. *New Journal of Physics* 16(9), p. 093021, doi:10.1088/1367-2630/16/9/093021.

[5] John S. Bell (1964): *On the Einstein-Podolsky-Rosen paradox*. *Physics* 1(3), pp. 195–200.

[6] A. R. Calderbank, E. M. Rains, P. W. Shor & N. J. A. Sloane (1997): *Quantum Error Correction and Orthogonal Geometry*. *Physical Review Letters* 78(3), pp. 405–408, doi:10.1103/PhysRevLett.78.405.

[7] B. Coecke & É.O. Paquette (2010): *Categories for the Practising Physicist*. In Bob Coecke, editor: *New Structures for Physics*, 813, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 173–286, doi:10.1007/978-3-642-12821-9_3.

[8] Bob Coecke & Ross Duncan (2008): *Interacting Quantum Observables*. In: *Automata, Languages and Programming*, 5126, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 298–310, doi:10.1007/978-3-540-70583-3_25.

[9] Bob Coecke & Ross Duncan (2011): *Interacting quantum observables: categorical algebra and diagrammatics*. *New Journal of Physics* 13(4), p. 043016, doi:10.1088/1367-2630/13/4/043016.

[10] Bob Coecke & Bill Edwards (2011): *Toy Quantum Categories (Extended Abstract)*. *Electronic Notes in Theoretical Computer Science* 270(1), pp. 29–40, doi:10.1016/j.entcs.2011.01.004.

[11] Bob Coecke & Bill Edwards (2012): *Spekkens's toy theory as a category of processes*. In Samson Abramsky & Michael Mislove, editors: *Proceedings of Symposia in Applied Mathematics*, 71, American Mathematical Society, Providence, Rhode Island, pp. 61–88. Available at `http://arxiv.org/abs/1108.1978`.

[12] Bob Coecke, Bill Edwards & Robert W. Spekkens (2011): *Phase Groups and the Origin of Non-locality for Qubits*. *Electronic Notes in Theoretical Computer Science* 270(2), pp. 15–36, doi:10.1016/j.entcs.2011.01.021.

[13] Bob Coecke & Éric Oliver Paquette (2008): *POVMs and Naimark's Theorem Without Sums*. *Electronic Notes in Theoretical Computer Science* 210, pp. 15–31, doi:10.1016/j.entcs.2008.04.015.

[14] D. Deutsch (1989): *Quantum Computational Networks*. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 425(1868), pp. 73–90, doi:10.1098/rspa.1989.0099.

[15] Ross Duncan & Simon Perdrix (2009): *Graph States and the Necessity of Euler Decomposition*. In: *Mathematical Theory and Computational Practice*, 5635, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 167–177, doi:10.1007/978-3-642-03073-4_18. For the full proofs see arXiv:0902.0500.

[16] Daniel Gottesman (1997): *Stabilizer Codes and Quantum Error Correction*. Ph.D. thesis, Caltech. Available at `http://arxiv.org/abs/quant-ph/9705052`.

[17] André Joyal & Ross Street (1991): *The geometry of tensor calculus, I*. Advances in Mathematics 88(1), pp. 55–112, doi:10.1016/0001-8708(91)90003-P.

[18] Saunders Mac Lane (1998): *Categories for the working mathematician*, 2nd edition. Springer, New York.

[19] Maarten Van den Nest, Jeroen Dehaene & Bart De Moor (2004): *Graphical description of the action of local Clifford transformations on graph states*. Physical Review A 69(2), p. 022316, doi:10.1103/PhysRevA.69.022316.

[20] Michael A. Nielsen & Isaac L. Chuang (2010): *Quantum computation and quantum information*. Cambridge University Press, Cambridge; New York.

[21] Matthew F. Pusey (2012): *Stabilizer Notation for Spekkens' Toy Theory*. Foundations of Physics 42(5), pp. 688–708, doi:10.1007/s10701-012-9639-7.

[22] Matthew F. Pusey, Jonathan Barrett & Terry Rudolph (2012): *On the reality of the quantum state*. Nature Physics 8(6), pp. 476–479, doi:10.1038/nphys2309.

[23] André Ranchin (2014): *Depicting qudit quantum mechanics and mutually unbiased qudit theories*. Electronic Proceedings in Theoretical Computer Science 172, pp. 68–91, doi:10.4204/EPTCS.172.6.

[24] Peter Selinger (2007): *Dagger Compact Closed Categories and Completely Positive Maps: (Extended Abstract)*. Electronic Notes in Theoretical Computer Science 170(0), pp. 139–163, doi:10.1016/j.entcs.2006.12.018.

[25] Robert W. Spekkens (2007): *Evidence for the epistemic view of quantum states: A toy theory*. Physical Review A 75(3), p. 032110, doi:10.1103/PhysRevA.75.032110.

[26] Robert W. Spekkens (2014): *Quasi-quantization: classical statistical theories with an epistemic restriction*. arXiv:1409.5041.

A Appendix

A.1 Proofs of results about graph states, GS-LO diagrams, and rGS-LO diagrams

Here, we give the proofs for results stated in section 5.3 where they differ significantly from the corresponding proofs in the ZX-calculus. The ZX-calculus is introduced in [9] and extended in [15]. The completeness proof for the stabilizer ZX-calculus can be found in [4].

Lemma 8, sketch. The proof is similar to the ZX-calculus case as given by Duncan and Perdrix [15]. We show here as an example the case of the complete graph on three vertices (rearranged with two inputs at the bottom for ease of reading):
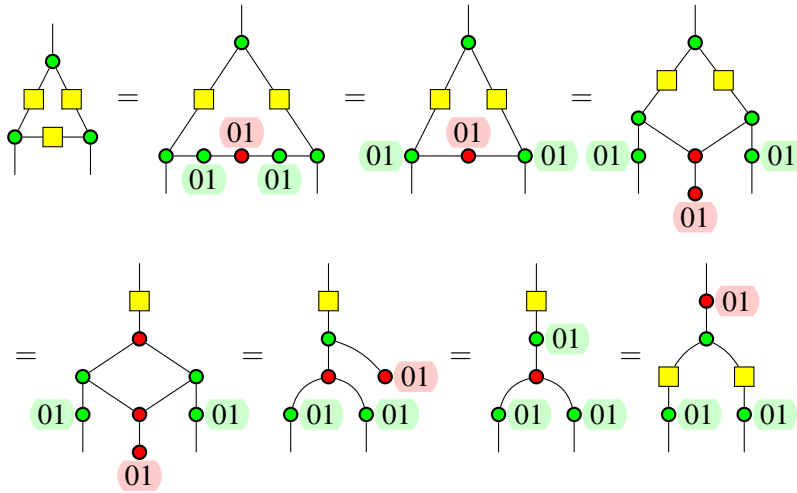
The first equality uses the decomposition of ▢ in terms of red and green phase shifts In the second step, the spider rule is used to merge the green phases with their green neighbours. Subsequently, the red and green phased spiders are "pulled apart", again using the spider law. In the fifth step, the colour change law and the fact that ▢ is self-inverse are used to change the green node at the top into a red one. The next step is an application of the bialgebra law. The penultimate step uses the fact that $\bullet\,01 = \bullet\,01$, which is the case $a = 0$ of (49) below. Lastly, the colour change rule is applied again.

The full proof then proceeds by induction over the number of vertices in the graph state. □

*Theorem 10, sketch.* The proof is analogous to the proof of Theorem 7 in [4], noting the following facts:

- Let $a \in \{0,1\}$ and $\bar{a} = a \oplus 1$, then:

$$(49)$$

Here, the first step uses the fact that ▢ is self-inverse and the second step uses the decomposition of ▢ into red and green phase shifts. The third step is an application of the spider law to merge the bottom two nodes, which is again used in the fourth step to pull apart the green node. In the

fifth step, the bottom red node is copied: this works for both values of a. The penultimate step, involves dropping the scalar diagram on the left and merging the two red nodes in the non-scalar part by the spider law. The last equality is by the colour change law.

- Any single toy bit operator can be written as

$$\begin{array}{l} \circ\ ab \\ \bullet\ cd \\ \circ\ ef \end{array}$$
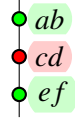
for some $a,b,c,d,e,f \in \{0,1\}$.

- $\circ\ 11$ and $\bullet\ 11$ denote the zero scalar.

- A loop with a $\square$ node in it disappears:

$$\tag{50}$$

$\square$

Theorem 11. By theorem 10, any state diagram in the toy theory is equal to some GS-LO diagram. Lemma 9 shows that each vertex operator in the GS-LO diagram can be brought into the form:

$$\begin{array}{l} \circ\ ab \\ \bullet\ cd \end{array} \quad \text{or} \quad \begin{array}{l} \bullet\ 01 \\ \circ\ e\bar{e} \\ \bullet\ fg \end{array} ,$$

where $a,b,c,d,e,f,g \in \{0,1\}$. Note that the cases $c = 0 = d$ and $f = 0 = g$ of the above normal forms correspond exactly to the elements of R as defined in (48). A local complementation about a vertex v pre-multiplies the vertex operator of v with $\bullet\ 01$ and a fixpoint operation with $\bullet\ 11$, so any vertex operator can be brought into one of the above forms by some combination of local complementations and fixpoint operations about the corresponding vertex. The other effects of local complementations are to toggle some of the edges in the graph state and to pre-multiply the vertex operators of neighbouring vertices by $\circ\ 01$, whereas fixpoint operations leave the edges invariant and pre-multiply the vertex operators of neighbouring vertices by $\circ\ 11$. The set R is not mapped to itself under repeated pre-multiplication with $\circ\ 01$: this transformation sends the set $\{\circ\ ab\}$ for $a,b \in \{0,1\}$ to itself, but it maps:

$$\left\{ \begin{array}{l} \bullet\ 01 \\ \circ\ a\bar{a} \end{array} \right\} \mapsto \left\{ \begin{array}{l} \bullet\ 01 \\ \circ\ a\bar{a} \end{array} \quad \begin{array}{l} \bullet\ 01 \end{array} \quad \begin{array}{l} \circ\ 11 \\ \bullet\ 10 \end{array} \right\}. \tag{51}$$

The normal form of a vertex operator contains at most two red nodes. Once a vertex operator is in one of the forms in R, pre-multiplication by green phase operators does not change the number of red nodes it contains when expressed in normal form. Thus the process of removing red nodes from the vertex operators by applying local complementations must terminate after at most $2n$ steps for an n-toy bit diagram, at which point all vertex operators are elements of the set R.

With all vertex operators in R, suppose there are two adjacent toy bits u and v which both have red nodes in their vertex operators, i.e. there is a subdiagram of the form:


(52)

with $a, b, \in \{0, 1\}$. A local complementation along the edge $\{u, v\}$ maps the vertex operator of u to:

(53)
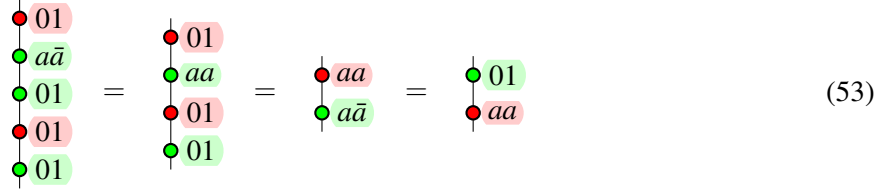
and similarly for v. After this, if $a = 1$, we apply a fixpoint operation to u and if $b = 1$ we apply a fixpoint operation to v. After this, the vertex operators on both u and v are green phase operators. Vertex operators of toy bits adjacent to u or v are pre-multiplied with some power of ● 11 , which maps $R \to R$. Thus each such operation removes the red nodes from a pair of adjacent toy bits and leaves all vertex operators in the set R. Hence after at most $n/2$ such operations, it will be impossible to find a subdiagram as in (52). Thus, the diagram is in reduced GS-LO form. $\qquad\square$

Proposition 12, sketch. The effect of the local complementations on the vertex operators of p and q is as follows:

(54)
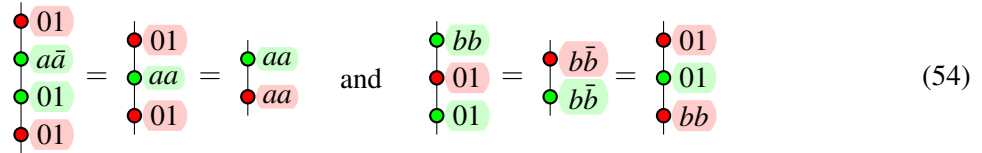
If $a = 1$, we apply a fixpoint operation to p and if $b = 1$, we apply a fixpoint operation to q; then the vertex operators of p and q are in R. The fixpoint operations add ● 11 to neighbouring toy bits, which maps the set R to itself. As fixpoint operations do not change any edges, we do not have to worry about them when considering whether the rest of the diagram satisfies definition 4.

The rest of the proof is analogous to the stabilizer QM case in [4]. $\qquad\square$

Proposition 13. After the local complementation along the edge, the vertex operator of p is given by (53). For the vertex operator of q, we have:

(55)
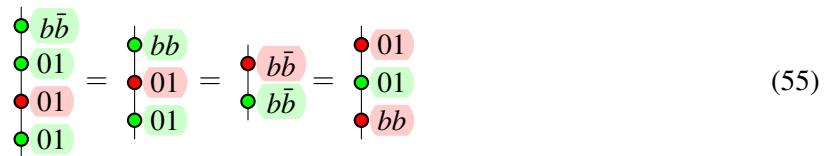
Thus if a or b is 1, we apply a fixpoint operator to the appropriate vertex. From the properties of local complementations along edges it follows that the overall transformation preserves the two properties of rGS-LO states. $\qquad\square$

A.2   Proof of completeness result

The arguments in the following proof closely follow the proof of Lemma 17 in [4]. As the diagrams are complicated and differ in subtle ways from the ZX-calculus ones, the proof is nevertheless produced in full here.

Lemma 15. Let D_1 be the diagram in which p has the red node, D_2 the other diagram. There are multiple cases:

In either diagram, p has no neighbours: In this case, the overall state factorises and the two diagrams are equal only if the two states of p are the same. But:

$$\stackrel{\bullet \, ab}{\stackrel{|}{\bullet}} \; = \; \stackrel{\bullet}{\bullet} ab \; \neq \; \stackrel{\bullet}{\bullet} cc \; = \; \stackrel{\bullet \, 01}{\stackrel{|}{\bullet \, c\bar{c}}} \; = \; \stackrel{\bullet \, 01}{\stackrel{|}{\bullet \, c\bar{c}}} \; = \; \stackrel{\bullet \, 01}{\stackrel{|}{\stackrel{\bullet \, c\bar{c}}{\stackrel{|}{\bullet}}}} \tag{56}$$

for $a,b,c \in \{0,1\}$, so the diagrams must be unequal.

p is isolated in one of the diagrams but not in the other: We argue in section 5.1 that, as in stabilizer QM, two toy graph states with local operators are equal only if one can be transformed into the other via a sequence of local complementations with corresponding changes to the local operators. As a local complementation never turns a vertex with neighbours into a vertex without neighbours, or conversely, the two diagrams cannot be equal.

p has neighbours in both diagrams: Without loss of generality, assume that p is the first toy bit. Let N_1 be the set of all toy bits that are adjacent to p in D_1, and define N_2 similarly. The vertex operators of any toy bit in N_1 must be green phases in both diagrams. In D_1, this is because of the definition of rGS-LO diagrams, in D_2 it is because the pair of diagrams is simplified. Suppose the original diagrams involve n toy bits each. Let G be the graph on n vertices (named according to the same convention as in D_1 and D_2) whose edges are $\{\{p,v\}|v \in N_1\}$. Now consider the following diagram:

 (57)

where the ellipse labelled G denotes the toy graph state corresponding to G, except that each vertex in the graph has not only an output but also an input. Call this diagram U. It is easy to see that U is invertible: composing it with itself upside-down yields the identity. Therefore composing this diagram with D_1 and D_2 will yield two new diagrams which are equal if and only if $D_1 = D_2$. We will denote the new diagrams by $U \circ D_1$ and $U \circ D_2$ and show that, no matter what the properties of D_1 and D_2 are (beyond the existence of an unpaired red node on p),

- in $U \circ D_1$, the toy bit p is in state $\stackrel{\bullet}{\bullet}$ or $\stackrel{\bullet \, 11}{}$;
- in $U \circ D_2$, p is either entangled with other toy bits, or in one of the states $\stackrel{\bullet}{} ab$, where $a,b \in \{0,1\}$.

By the arguments used in the first two cases, this implies that $U \circ D_1 \neq U \circ D_2$ and therefore $D_1 \neq D_2$.

Let $n = |N_1|$, $m = |N_1 \cap N_2|$, and suppose the toy bits are arranged in such a way that the first m elements of N_1 are those which are also elements of N_2, if there are any. Consider first the effect on

diagram D_1. The local operator on p combines with the single-toy bit operators from U to:

$$\text{(58)}$$

where $a \in \{0,1\}$. As green phase shifts can be pushed through other green nodes, the subdiagram involving p and the elements of N_1 in $U \circ D_1$ is equal to:

$$\text{(59)}$$

Here, $b_1,\ldots,b_n,c_1,\ldots,c_n \in \{0,1\}$. Note that at the end p is isolated and in the state $\bullet\, aa$. The fact that we have ignored all toy bits not originally adjacent to p in D_1 does not change that.

Next consider $U \circ D_2$. As N_1 is not in general equal to N_2, the subdiagram consisting of p and vertices in N_1 looks as follows:
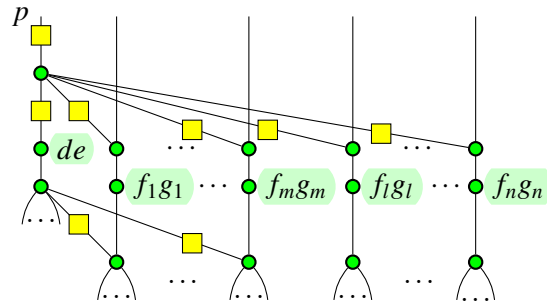
where $l = m+1$ and $d,e,f_1,\ldots,f_n,g_1,\ldots,g_n \in \{0,1\}$. Note that we neglect edges that do not involve p and also edges between p and vertices not in N_1. We will now distinguish different cases, depending on the values of d and e.

If $d = 0, e = 1$ apply a local complementation about p. This does not change the edges incident on p:
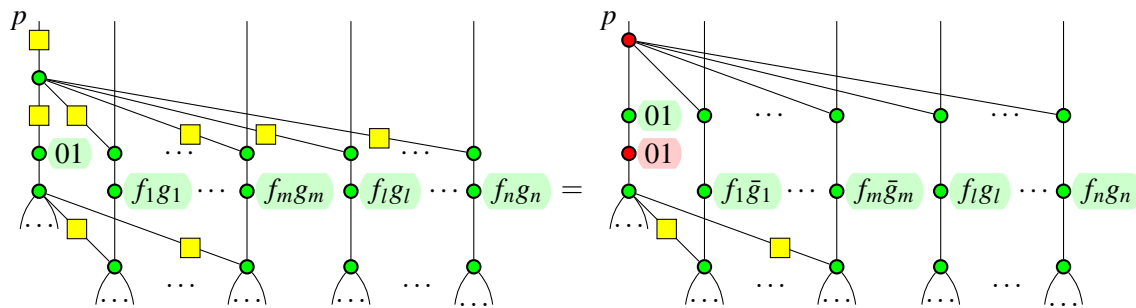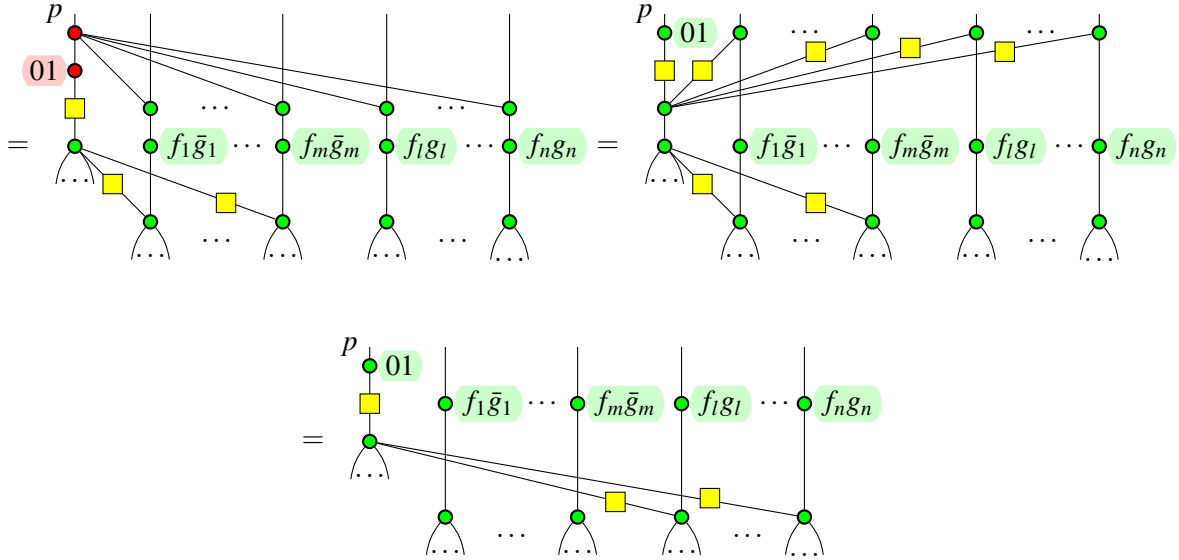
Now if $N_1 = N_2$, p has no more neighbours and is in the state ●01 . This is not the same as the state p has in diagram 1, so the diagrams are not equal. Else, after the application of U, p still has some neighbours in diagram 2. Local complementations do not change this fact. Thus the two diagrams cannot be equal. The case $d = 1, e = 0$ is entirely analogous, except that there is a fixpoint operation in addition to the local complementation at the beginning.

If $d = e = 0$, there are two sub-cases. First, suppose there exists $v \in N_2$ such that $v \notin N_1$. Apply a local complementation about this v. This operation changes the vertex operator on p to ●01 . It also changes the edges incident on p, but the important thing is that p will still have at least one neighbour. Thus we can proceed as in the case $d = 0, e = 1$.

Secondly, suppose there is no $v \in N_2$ which is not in N_1. Since $N_2 \neq \emptyset$ ($N_2 = \emptyset$ corresponds to the case "p has no neighbours in D_2", which was considered above), we must then be able to find $v \in N_1 \cap N_2$. The diagram looks as follows, where now $m > 0$ (again, we are ignoring edges that do not involve p):
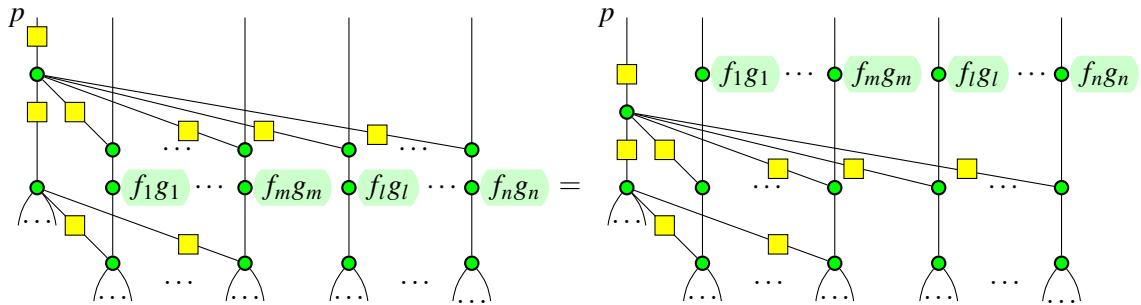
To show that the two diagrams are unequal it suffices to show that in diagram 2 the state of p either factors out, but is not ● or ●11 , or that it remains entangled with other toy bits. We are thus justified in ignoring large portions of the above diagram to focus only on p, v and the edge between the two. In particular, we will ignore for the moment the edges between p and toy bits other than v, as well as the last ▢ on p. Then:
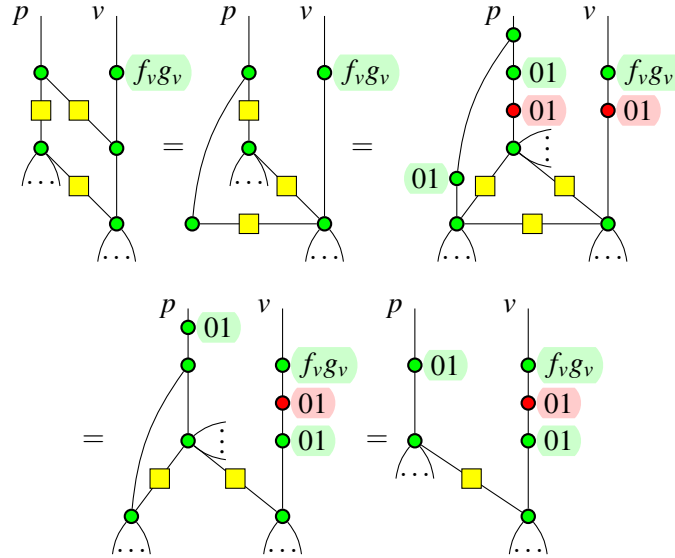
where for the second equality we have applied a local complementation to v and used the Euler decomposition, the third equality follows by a local complementation on p, and the last one comes from the merging of p with the green node in the bottom left. Note that, in the end, p and v are still connected by an edge. None of the operations we ignored in picking out this part of the diagram will change that. Thus, as before, the state of p cannot be the same as in diagram 1. The two diagrams are unequal.

The case $d = e = 1$ is analogous to $d = e = 0$, except in either sub-case we start with a fixpoint operation on the chosen v.

We have thus shown that a simplified pair of rGS-LO diagrams are not equal if there are any unpaired red nodes.   □