Rewriting measurement-based quantum computations with generalised flow

Ross Duncan[1⋆] and Simon Perdrix[2]

[1] Oxford University Computing Laboratory
ross.duncan@comlab.ox.ac.uk
[2] CNRS, LIG, Université de Grenoble
Simon.Perdrix@imag.fr

Abstract. We present a method for verifying measurement-based quantum computations, by producing a quantum circuit equivalent to a given deterministic measurement pattern. We define a diagrammatic presentation of the pattern, and produce a circuit via a rewriting strategy based on the generalised flow of the pattern. Unlike other methods for translating measurement patterns with generalised flow to circuits, this method uses neither ancilla qubits nor acausal loops.

1 Introduction

The one-way quantum computer (1WQC) [1] is model of quantum computation which is a very promising candidate for physical implementation, and also has many interesting theoretical properties (in complexity theory, for instance [2, 3]). The basis of the 1WQC is an entangled resource state, which is gradually consumed by performing local measurements upon it. By careful choice of measurements, any quantum computation can be performed. In this paper we address the task of verifying properties of one-way computations by using rewriting strategies in a graphical framework, which originates in categorical analyses of quantum mechanics [4].

The main task is to verify the correctness of a given one-way computation—presented in the pattern syntax of the *measurement calculus* [5]—by producing an equivalent quantum circuit. We will also verify that the pattern can be carried out deterministically: that is, we check that the non-deterministic effects of quantum measurement are properly corrected by the pattern.

The question of determinism in the one-way model has been previously addressed by *flow techniques*; see [6, 7]. These techniques examine the resource state: if it has the correct geometry then any errors introduced by the non-deterministic nature of quantum measurements can be corrected, and the resulting computation will be deterministic. Both causal flow [6] and generalised flow [7] do not address any concrete pattern, rather they simply assert the existence of a deterministic computation using the given resource state. In fact, generalised

⋆ Supported by EPSRC postdoctoral fellowship EP/E045006/1.

flow (gflow) characterises the notion of *uniform determinism*, where the actual choice of the measurements is irrelevant. (Causal flow provides a sufficient condition.) Our work relaxes the uniformity restriction and derive correctness proofs in cases where the choice of measurement is significant.

The problem of producing a circuit equivalent to a given measurement-based quantum computation is of great importance. In [8], an automated translation has been proposed for measurement-based computations which have a causal flow. In [9], the author presents a similar technique based on causal flow and notes that her method produces circuits with "time-like loops" if applied on measurement-based computations which do not have a causal flow. In this work we rely on the bialgebraic structure induced by quantum complementarity to produce equivalent circuits from measurement-based quantum computations which do not have causal flow. Unlike other translations from 1WQC, the circuits we generate do not make use of any ancilla qubits.

The diagrammatic calculus we employ draws from the long tradition of graphical representations of monoidal categories. Aside from providing a very intuitive notation for reasoning about information flow, the categorical approach to quantum computation (see for example [10–12]), provides a clearer view of the structure of quantum informatic phenomena than conventional approaches. The particular system of this paper is essentially that of [4], and the bialgebraic relations between complementary quantum observables exposed there form the core of the main lemma of this paper[3].

The structure of this paper is as follows: in Section 2, we introduce the diagrammatic syntax and its semantics; in Section 3 we introduce the rewrite system used to derive our results. This rewrite system has no particularly nice properties—it is neither confluent nor terminating—but in the rest of the paper we will define strategies to achieve our results. Section 4 introduces the measurement calculus and its translation into diagrams, and Section 5 how to derive the circuit-like form Due to space restrictions, the proofs have been omitted.

Notational conventions. When u, v are vertices in some graph G, we write $u \sim v$ to indicate that they are adjacent. The *neighbourhood* of u, denoted $N_G(u)$, is defined $N_G(u) = \{v \in V \ : \ u \sim v\}$. Given some $K \subseteq V$, the *odd neighbourhood of* K is defined by $\mathrm{Odd}_G(K) = \{v \in V \ : \ |N_G(v) \cap K| = 1 \mod 2\}$, i.e. $\mathrm{Odd}_G(K)$ is the set of vertices which have an odd number of neighbours in K. We use Dirac notation to denote vectors, e.g. $|\psi\rangle$. The standard basis for $\mathbb{C}^2$ is denoted $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$; we will also use the complementary basis $|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $|-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$.

[3] The calculus has been implemented in a mechanised rewriting tool: see [13].

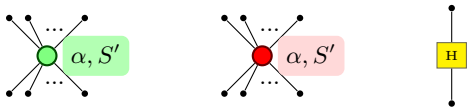


Fig. 1. Permitted interior vertices

2 Diagrams

Definition 1. *An* open graph *is a triple* (G, I, O) *consisting of an undirected graph* $G = (V, E)$ *and distinguished subsets* $I, O \subseteq V$ *of* input *and* output *vertices* I *and* O*. The set of vertices* $I \cup O$ *is called the* boundary *of* G*, and* $V \setminus (I \cup O)$ *is the* interior *of* G*.*

Definition 2. *Let* S *be some set of variables. A* formal diagram over S *is an open graph whose boundary vertices are always of degree 1, and whose interior vertices are restricted to the following types:*

- *Z vertices, labelled by an angle* $\alpha \in [0, 2\pi)$ *and some collection of variables* $S' \subseteq S$*; these are shown as (light) green circles,*
- *X vertices, labelled by an angle* $\alpha \in [0, 2\pi)$ *and some collection of variables* $S' \subseteq S$*; these are shown as (dark) red circles,*
- *H (or Hadamard) vertices, restricted to degree 2; shown as squares.*

The allowed vertices are shown in Figure 1.

Diagrams are oriented such that the inputs are at the top and the outputs are at the bottom, and hence the implied temporal (partial) order of the components is from top to bottom.

If an X or Z vertex is labelled by $\alpha = 0$ then the label is omitted. In the case where S' is not empty then the corresponding vertex is called *conditional*; if no conditional vertices occur in a diagram it is *unconditional*. For each S the formal diagrams over S form a symmetric monoidal category (in fact compact closed) in the evident way: the objects of the category are sets and an arrow $g : A \to B$ is a diagram whose underlying open graph is (G, A, B). The tensor product is disjoint union, and composition $g \circ f$ is defined by identifying the output vertices of f with the input vertices of g. For more details see [14, 15]. Denote this category $\mathbb{D}(S)$; we denote the category $\mathbb{D}(\emptyset)$ of unconditional diagrams by $\mathbb{D}$. Note that the components shown in Figure 1 are the generators of $\mathbb{D}(S)$.

Informally, the edges of a diagram are interpreted as qubits, though some caution is required. Different edges can represent the same physical qubit at different stages of the computation, and certain edges do not represent qubits at all: they encode correlations between different parts of the system. Example 7 shows how 2-qubit gates are encoded using such correlations. The vertices of the diagram are interpreted as local operations, possibly conditioned on the variables, so that the entire diagram yields a superoperator from its inputs to its outputs. We define an interpretation functor to make this intuition precise.

Definition 3. *Call $v : S \to \{0,1\}$ a* valuation *of S; for each valuation v, we define a functor $\hat{v} : \mathbb{D}(S) \to \mathbb{D}$ which simply instantiates the labels of Z and X vertices occurring in each diagram. If a vertex z is labelled by α and S', then $\hat{v}(z)$ is labelled by 0 if $\prod_{s \in S'} v(s) = 0$ and α otherwise.*

Definition 4. *Let $[\![\cdot]\!] : \mathbb{D} \to$ **FDHilb** be a traced monoidal functor; define its action on objects by $[\![A]\!] = \mathbb{C}^{2^n}$ whenever $n = |A|$; define its action on the generators as:*

$$[\![\; \alpha \;]\!] = \begin{cases} |0\rangle^{\otimes m} \mapsto |0\rangle^{\otimes n} \\ |1\rangle^{\otimes m} \mapsto e^{i\alpha} |1\rangle^{\otimes n} \end{cases} \qquad [\![\; \alpha \;]\!] = \begin{cases} |+\rangle^{\otimes m} \mapsto |+\rangle^{\otimes n} \\ |-\rangle^{\otimes m} \mapsto e^{i\alpha} |-\rangle^{\otimes n} \end{cases}$$

$$[\![\; H \;]\!] = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

The value of $[\![\cdot]\!]$ on all other arrows is then fixed by the requirement that it be a traced monoidal functor[4]*.*

Definition 5. *The denotation of a diagram D over variables S is a superoperator constructed by summing over all the valuations of S:*

$$\rho \mapsto \sum_{v \in 2^S} [\![\hat{v}(D)]\!] \rho [\![\hat{v}(D)]\!]^\dagger .$$

Example 6 (Pauli Matrices). The Pauli X and Z matrices can be defined by degree 2 vertices:

$$[\![\; \pi \;]\!] = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad [\![\; \pi \;]\!] = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Example 7 (2-qubit gates). Composing an X with a Z vertex yields the 2-qubit $\wedge X$ (controlled-NOT) gate where the Z vertex is the control qubit. The $\wedge Z$ gate is defined similarly.

$$\wedge X = [\![\; \;]\!] = [\![\; \;]\!] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}; \quad \wedge Z = [\![\; \;]\!] = [\![\; \;]\!] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

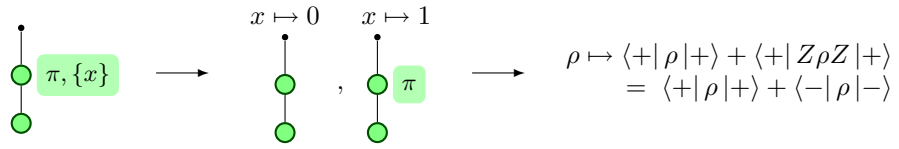
In both cases, the diagonal edge connecting the two sides of the diagram produces the correlation of the two physical qubits represented by the vertical edges.

Example 8 (Preparing and measuring qubits). The preparation of a fresh qubit is represented by a single vertex with no input edges and one output edge:

$$[\![\; \;]\!] = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle; \qquad [\![\; \;]\!] = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = |+\rangle .$$

[4] Again, the full details of this construction regarding cyclic graphs and traces can be found in [14].

To encode a projection we use a dual diagram to preparation; the non-determinism of measurement is represented using a conditional operation whose two possible valuations correspond to the two possible outcomes:



From the preceding examples, it will be obvious that our diagrammatic language can represent a universal set of quantum gates, and hence can represent all quantum circuits. However, not every diagram corresponds to a quantum circuit.

Definition 9. *A diagram is called* circuit-like *if (1) all of its boundary, X, and Z vertices can be covered by a set $\mathcal{P}$ of disjoint paths, each of which ends in an output; (2) every cycle in the diagram traverses at least one edge covered by $\mathcal{P}$ in the direction opposite to that induced by the path; and, (3) it is weakly 3-coloured in the following sense: in every connected subgraph whose vertices are all the same type, no two (non-boundary) vertices are labelled by the same set S of variables.*

The paths mentioned in the condition (1) represent the physical qubits of a quantum circuit, and condition (2) prevents information being fed from a later part of the circuit to an earlier part. Notice that condition (1) allows, but does not require, H vertices to be covered by the path; hence the $\wedge Z$ example above is circuit-like. Condition (3) is a requirement that circuit-like diagrams are normal with respect to certain rewrite rules introduced in Section 3.

The path-cover $\mathcal{P}$ of a circuit-like diagram gives a factorisation of the diagram into unitary gates, thus defining a quantum circuit, possibly with some part of its input fixed. More precisely, every such diagram can be obtained from a circuit by the rewrite rules of the following section. The following is an immediate consequence.

Proposition 10. *If D is unconditional and circuit-like then $[\![D]\!]$ is a unitary embedding.*

3 Rewrites

The map $[\![\cdot]\!]$ gives an interpretation of diagrams as linear maps. This interpretation, however, is not injective: there are numerous diagrams which denote the same linear map. To address this difficulty, an equational theory on $\mathbb{D}$ is required. We will now introduce such a theory via a set of rewriting rules.

Definition 11. *Let R be least transitive and reflexive relation on $\mathbb{D}$ generated by the local rewrite rules shown in Figure 2. Let $\leftrightarrow^*$ denote the symmetric closure of R.*

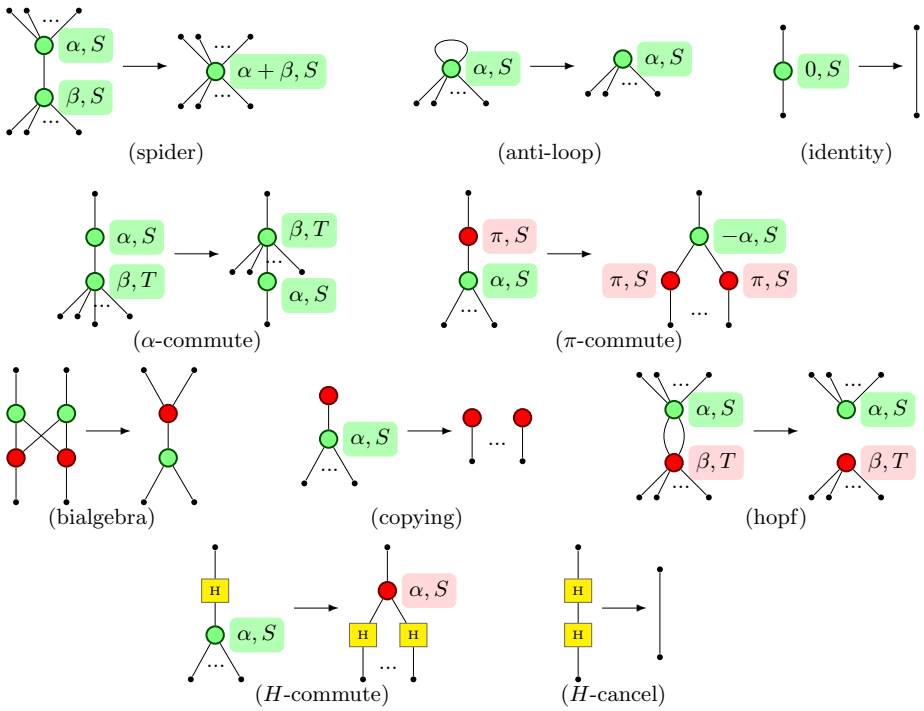


Fig. 2. Rewrite rules for system R. We present the rules for the Z subsystem; to obtain the complete set of rules exchange the colours in the rules shown above.

The diagrammatic syntax, and the equations of the rewrite rules are derived from those introduced in [4]. The Z family of vertices correspond to operations defined with respect to the eigenbasis of the Pauli Z operator; a copying operation for this basis and phase rotations which leave it fixed. Similarly the X family are defined with respect to the Pauli X. The H vertices represent the familiar 1-qubit Hadamard, which maps sends each family onto the other. Each family individually forms a special Frobenius algebra, and together they form a Hopf algebra with trivial antipode. Space does not permit a more thorough justification of these particular operations and rules, beyond the following:

Proposition 12. *The rewrite system R is sound with respect to the interpretation map $[\![\cdot]\!]$.*

Despite its soundness, this rewrite system does not have many good properties. It is not complete with respect to the interpretation in Hilbert space; an example of an unprovable true equation is discussed in [16]. It is manifestly not terminating since several of the rules are reversible, for example the α-commutation rule; other rules can generate an infinite sequence of rewrites, for example the π-commutation rule. The subsystem without H is known to be neither confluent nor terminating [17]. Rather than attempt to remedy these defects by tinkering

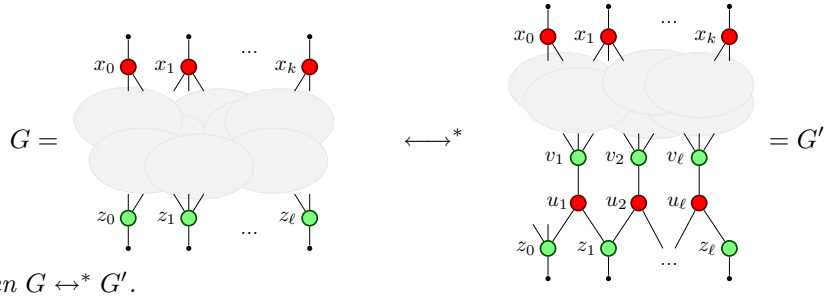
with the system, in this paper we will use particular rewrite strategies to produce circuit-like diagrams.

Proposition 10 implies that any unconditional circuit-like diagram has a natural interpretation as a quantum circuit, hence the existence of such a reduct for a given diagram shows that the diagram is equivalent to the derived circuit. In the following sections we will see how to apply this idea to the verification of one-way quantum computations.

Lemma 13 (Main Lemma). *Given a diagram D, let $\mathcal{X} = \{x_0, \ldots x_k\}$ and $\mathcal{Z} = \{z_0, \ldots, z_\ell\}$ be sets of its X and Z vertices respectively, such that the subdiagram G, induced by $\mathcal{Z} \cup \mathcal{X}$, is bipartite—that is, for all i, j, we have $x_i \not\sim x_j$ and $z_i \not\sim z_j$ in G.*

Define a new graph G' with vertices $V_{G'} = V_G \cup \{u_1, \ldots u_\ell\} \cup \{v_1, \ldots, v_\ell\}$, and such that for any $0 \leq i \leq k$ and $1 \leq j \leq \ell$,

- *there are edges (u_j, v_j), (u_j, x_{j-1}) and $(u_j, x_j) \in G'$;*
- *there is an edge $(x_i, z_0) \in G'$ iff $x_i \in \mathrm{Odd}_G(\mathcal{Z})$*
- *there is an edge $(x_i, v_j) \in G'$ iff $x_i \in \mathrm{Odd}_G(\{z_j, \ldots z_\ell\})$.*



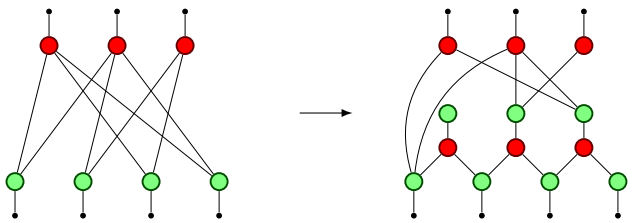
Then $G \leftrightarrow^* G'$.

Note that there is an edge between an X vertex x and a Z vertex z in G if and only if there is an odd number of paths between x and z in G'.

A direct proof of the lemma can be given using rewrites or R, although we note that is a special case of the well known normal form theorem for Hopf algebras (see [18] for a nice version).

Each instance of the Main Lemma provides a new admissible rule $G \longrightarrow G'$; since $\leftrightarrow^*$ is just the equivalence relation generated by R, these new rules are sound with respect to the interpretation map $[\![\cdot]\!]$. One way to view the lemma is as a new set of rewrite rules S compatible with R.

Example 14. An admissible rule from the schema S. :



4 The Measurement Calculus

The *measurement calculus*, introduced by Danos, Kashefi and Panangaden [5], is a formal calculus for one-way quantum computations [1]. We review here the basic features of the calculus; for a complete exposition see [5].

Definition 15. *A* measurement pattern *consists of a set V of qubits, with distinguished subsets I and O of inputs and outputs respectively, and, in addition, a sequence of* commands *chosen from the following operations.*

- *1-qubit preparations, N_i, which prepare the qubit $i \notin I$ to the state $|+\rangle$.*
- *2-qubit entangling operations, E_{ij}, which applies a $\wedge Z$ to qubits i and j.*
- *1-qubit measurements, ${}^s[M_i^\alpha]^t$, which act as destructive measurements on the qubit $i \notin O$, in the basis $|0\rangle \pm e^{(-1)^s i\alpha + t\pi} |1\rangle$, where $s, t \in \{0, 1\}$ are boolean values called* signals.
- *1-qubit corrections X_i^s and Z_j^t, which act as the Pauli X and Z operators on qubits i and j, if the signals s and t, respectively, are equal to 1; otherwise the corrections have no effect.*

A qubit is measured if and only if it is not an output. The set of signals is in bijection with the set $V \setminus O$ of measured qubits: signal s is set to 0 if the corresponding measurement yields the +1 eigenstate, and 1 otherwise.

Each pattern can be interpreted as a superoperator $\mathbb{C}^{2^{|I|}} \to \mathbb{C}^{2^{|O|}}$ via a linear map, called the *branch map*, for each possible vector of measurement outcomes, much as in Def. 5. Indeed each pattern can be translated into diagram with the same semantics.

Remark 16. The measurement operation ${}^s[M_i^\alpha]^t$ is equivalent to the sequence $M_i^\alpha X_i^s Z_i^t$. The following assumes that all measurements have been so decomposed.

Definition 17. *Let $\mathfrak{P}$ be a pattern. Define a diagram $D_\mathfrak{P}$ over $V \setminus O$ by translating the command sequence according to table 1, and composing in these elements in the the evident way.*

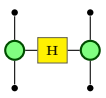
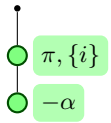
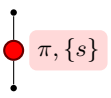
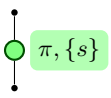
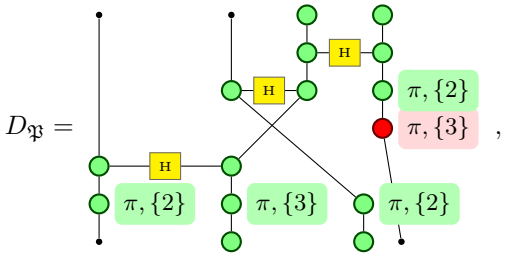
N_i	E_{ij}	M_i^α	X_i^s	Z_i^s
				

Table 1. Translation from pattern to diagram.

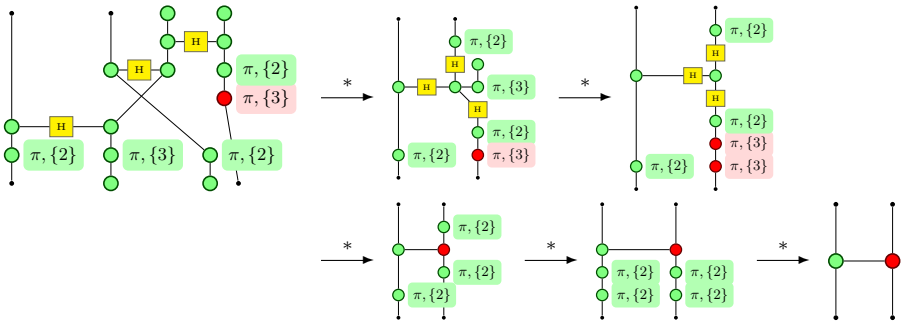
Example 18. The ubiquitous CNOT operation can be computed by the pattern $\mathfrak{P} = X_4^3 Z_4^2 Z_1^2 M_3^0 M_2^0 E_{13} E_{23} E_{34} N_3 N_4$ [5]. This yields the diagram,



where each qubit is represented by a vertical "path" from top to bottom, with qubit 1 the leftmost, and qubit 4 is the rightmost.

By virtue of the soundness of R and Proposition 10, if $D_{\mathfrak{P}}$ can be rewritten to a circuit-like diagram without any conditional operations, then the rewrite sequence constitutes a proof that the pattern computes the same operation as the derived circuit.

Example 19. Returning to the CNOT pattern of Example 18, there is a rewrite sequence, the key steps of which are shown below, which reduces the $D_{\mathfrak{P}}$ to the unconditional circuit-like pattern for CNOT introduced in Example 7. This proves two things: firstly that $\mathfrak{P}$ indeed computes the CNOT unitary, and that the pattern $\mathfrak{P}$ is *deterministic*.



One can clearly see in this example how the non-determinism introduced by measurements is corrected by conditional operations later in the pattern. The possibility of performing such corrections depends on the *geometry* of the pattern, the entanglement graph implicitly defined by the pattern.

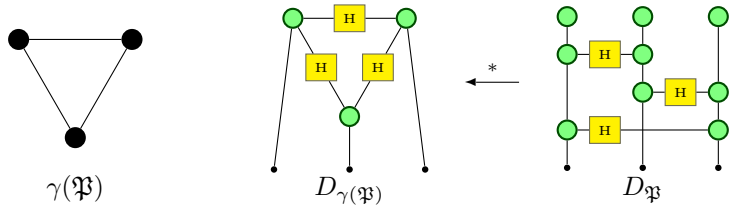
Definition 20. *Let $\mathfrak{P}$ be a pattern; the* geometry *of $\mathfrak{P}$ is an open graph $\gamma(\mathfrak{P}) = (G, I, O)$ whose vertices are the qubits of $\mathfrak{P}$ and where $i \sim j$ iff E_{ij} occurs in the command sequence of $\mathfrak{P}$.*

Definition 21. *Given a geometry $\Gamma = ((V, E), I, O)$ we can define a diagram $D_\Gamma = ((V_D, E_D), I_D, O_D)$ as follows:*

– $V_D = V + E + I + O$, *coloured such that:*
  - $v \in V$ *is an unconditional* Z *vertex in* D_Γ*, labelled by* $\alpha = 0$*;*
  - $e \in E$ *is an* H *vertex;*
  - $b \in I + O$ *is a boundary vertex.*
– *The edge relation is as follows:*
  - *if* $v \in I$*, or* $v \in O$*, in* Γ *then* $v_I \sim v_V$*, respectively* $v_O \sim v_V$*, in* D_Γ*;*
  - *if* $e = (v, v')$ *in* Γ*, then* $e_E \sim v_V$ *and* $e_E \sim v'_V$ *in* D_Γ*;*
– $v_I \in I_D$ *and* $v_O \in O_D$.

The Z vertices of $D_{\gamma(\mathfrak{P})}$ are in bijective correspondence with the qubits of $\mathfrak{P}$.

Example 22. Let $\mathfrak{P} = E_{12}E_{13}E_{23}N_1N_2N_3$. This pattern has no inputs or measurements: it simply prepares a triangular graph state. Notice that $D_{\gamma(\mathfrak{P})}$ is a reduct of $D_{\mathfrak{P}}$.



Given $D_{\gamma(\mathfrak{P})}$, we can adjoin measurements to construct a diagram $D^*_{\mathfrak{P}}$, such that $D_{\mathfrak{P}} \xrightarrow{*} D^*_{\mathfrak{P}}$. Justified by this, we shall use $D_{\gamma(\mathfrak{P})}$ in place of $D_{\mathfrak{P}}$, to allow properties based on the geometry to be imported directly. The most important such property is the generalised flow, or gflow.

Definition 23. *Let* (G, I, O) *be an open graph; a* generalised flow *(or* gflow*) is a pair* $(g, \prec)$*, with* $\prec$ *a partial order and* g *a function* $g : O^c \to \mathcal{P}(I^c)$ *which associates with every non output vertex a set of non input vertices such that:*

(G1). if $j \in g(i)$ *then* $i \prec j$*;*
(G2). if $j \in \mathrm{Odd}_G(g(i))$ *then* $j = i$ *or* $i \prec j$*;*
(G3). $i \in \mathrm{Odd}_G(g(i))$.

In the special case that $|g(v)| = 1$ *for all vertices* v*, the gflow is called a* causal flow*, or simply a* flow.

Theorem 24 ([7]). *If* (G, I, O) *has a gflow, then there exists a pattern* $\mathfrak{P}_0$ *such that* $\gamma(\mathfrak{P}_0) = (G, I, O)$ *and* $\mathfrak{P}_0$ *is deterministic, in the sense that all of its branch maps are equal. Further, this property does not depend on the angle of any measurement in* $\mathfrak{P}_0$.

Since different patterns may have the same geometry, it may be that $\gamma(\mathfrak{P}) = \gamma(\mathfrak{P}')$ but one is deterministic and the other is not. In the next section we describe how to produce a circuit-like diagram from $D_{\gamma(\mathfrak{P})}$ using a rewrite strategy based on the existence of a gflow.

5 Rewriting to circuits

Now we relate the various flow structures on a geometry, to the possibility that the corresponding pattern is deterministic.

Notice that Def. 23 can be readily adapted to define a gflow over an unconditional diagram: simply replace the vertices of the geometry with the non-H vertices of the diagram, and replace "adjacent" with "reachable via a path of zero or more H vertices". It is easy to see that the original definition on $\gamma(\mathfrak{P})$ and the modified version on $D_{\gamma(\mathfrak{P})}$ exactly coincide.

Now we demonstrate a rewriting strategy that will perform two tasks at once. If the open graph has a gflow, we will discover it. And, we will, in the process, transform the graph into a new graph which has a casual flow.

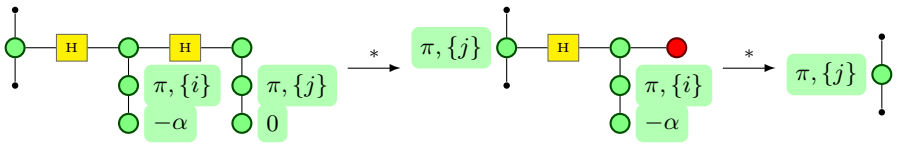
Lemma 25. *There is a convergent rewriting strategy such that if $\mathfrak{P}$ has a gflow then $D_{\gamma(\mathfrak{P})}\downarrow$ is circuit like.*

Proof (Sketch). Suppose we know the gflow on $D_{\gamma(\mathfrak{P})}$. For every non-output qubit i, the sets $g(i)$ and $\{i\} \cup \{j \in V \ : \ j \prec i\}$ provide the situation of the main lemma, hence we can rewrite the diagram according the induced rules. The new graph has again a gflow, and further, the overall size of the sets $g(i)$ has been reduced. It just remains to find the gflow if it exists; for this we can essentially simulate the method of [7] by the choice of admissible rules S.

Lemma 26. *$\mathfrak{P}$ has a causal flow if and only if $D_{\gamma(\mathfrak{P})}$ is circuit like.*

Theorem 27. *If a geometry Γ has a gflow then D_Γ can be rewritten to a circuit like diagram.*

Example 28. The existence of a gflow is a sufficient condition for a pattern $\mathfrak{P}$ to be circuit-like, but not necessary. For instance, although the pattern $\mathfrak{P} = M_3^0 M_2^\alpha E_{23} E_{12} N_2 N_3$ has no gflow, it can be rewritten to a circuit-like diagram:



This example shows that the verification using our rewriting technique is more powerful than the static gflow condition: the rewriting techniques can verify non-uniform properties, i.e. properties which depend on the actual measurement angles.

6 Conclusions, extensions, and future work

We have shown how to represent the measurement calculus in a diagrammatic form, and demonstrated how rewriting of these diagrams can prove the equivalence of computations. Our main result describes a rewriting strategy for transforming a measurement pattern it into a circuit-like diagram, which furthermore

uses no ancilla qubits. Although space limitations prevent its description here, this result can be extended. We can rewrite the resulting diagram to an unconditional diagram if and only if the given *pattern* is in fact deterministic—that is, free of programming errors. Indeed by suitable annotations of the diagram this strategy can discover where additional corrections should be added to the pattern to make it deterministic, effectively debugging the pattern. These techniques extend outside the realm of gflow since we can also show non-uniform determinism, as discussed in at the end of Section 5. One important area which we have not treated here is the depth complexity of the circuits constructed by our strategy. This will be examined in future work.

References

1. Raussendorf, R., Briegel, H.J.: A one-way quantum computer. Phys. Rev. Lett. **86** (2001) 5188–5191
2. Anne Broadbent, J.F., Kashefi, E.: Universal blind quantum computation. In: Proc. FoCS 2009. (2009)
3. Browne, D.E., Kashefi, E., Perdrix, S.: Computational depth complexity of measurement-based quantum computation. (2009) preprint: arXiv:0909.4673.
4. Coecke, B., Duncan, R.: Interacting quantum observables. In Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewic, I., eds.: Proc. ICALP 2008, Part II. Volume 5126 of LNCS., Springer (2008) 298–310
5. Danos, V., Kashefi, E., Panangaden, P.: The measurement calculus. J. ACM **54**(2) (2007)
6. Danos, V., Kashefi, E.: Determinism in the one-way model. Phys. Rev. A **74**(052310) (2006)
7. Browne, D., Kashefi, E., Mhalla, M., Perdrix, S.: Generalized flow and determinism in measurement-based quantum computation. New J. Phys. **9** (2007)
8. Duncan, R.: Verifying the measurement calculus by rewriting. In: DCM'07. (2007) Oral presentation.
9. Kashefi, E.: Lost in translation. In: Proc. DCM'07. (2007)
10. Abramsky, S., Coecke, B.: A categorical semantics of quantum protocols. In: Proc. LiCS 2004, IEEE Computer Society (2004) 415–425
11. Coecke, B., Pavlovic, D.: Quantum measurements without sums. In Chen, G., Kauffman, L.H., Lomonaco, S.J., J., eds.: The Mathematics of Quantum Computation and Technology, Taylor and Francis (2007)
12. Coecke, B., Paquette, E.O.: POVMs and Naimark's theorem without sums. In: Proceedings of QPL 2006. (2006)
13. Dixon, L., Duncan, R., Kissinger, A.: Quantomatic. Project home page: http://dream.inf.ed.ac.uk/projects/quantomatic/.
14. Duncan, R.: Types for Quantum Computing. PhD thesis, Oxford University (2006)
15. Dixon, L., Duncan, R.: Graphical reasoning in compact closed categories for quantum computation. Ann. Math. Artif. Intel. **56**(1) (2009) 23–42
16. Duncan, R., Perdrix, S.: Graph states and the necessity of Euler decomposition. In Ambos-Spies, K., Löwe, B., Merkle, W., eds.: Proc. CiE 2009. Volume 5635 of LNCS., Springer (2009) 167–177
17. Kissinger, A.: Graph rewrite systems for complementary classical structures in †-symmetric monoidal categories. Master's thesis, Oxford University (2008)
18. Lack, S.: Composing PROPs. Theor. Appl. Categ. **13**(9) (2004) 147–163