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. In addition this method can detect errors in the correction strategy used to make the pattern deterministic.

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]

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

do not address any concrete pattern, rather they simply assert the existence of a deterministic computation using the given resource state. In fact, generalised 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 improves on these methods by verifying that a given *pattern* is deterministic—i.e. that it is free of programming errors. By working directly with the pattern we can also relax 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 Sections 5 and 6 show how to derive the circuit form and correction strategies respectively. Due to space restrictions, the proofs have been relegated to an appendix.

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*.*

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

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

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

and whose boundary vertices are always of degree 1. The allowed vertices are shown in Figure 1.

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*. We only use conditional vertices of degree 2. For each S the formal diagrams over S form a

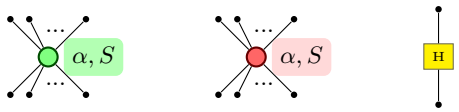


Fig. 1. Permitted vertices

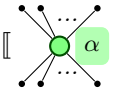
symmetric monoidal category (in fact compact closed) in the evident way: the objects of the category are sets and an arrow $g : A \to B$ is a diagram whose underlying open graph is (G, A, B). The tensor product is disjoint union, and composition $g \circ f$ is defined by identifying the output vertices 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 $\mathbb{D}(S)$.

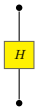
Diagrams are oriented such that the inputs are at the top and the outputs are at the bottom, and hence the implied temporal (partial) order of the components is from top to bottom. If such an order exists it is usually partial.

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 represent "virtual" qubits, which are used to encode two qubit operations such as the controlled-Z gate—see example 7. 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 $\alpha' = 0$ if $\prod_{s \in S'} v(s) = 0$ and $\alpha' = \alpha$ otherwise.*

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

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

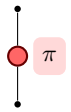
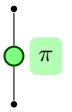
$$[\![\text{  }]\!] = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

where $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, *and* $|-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$.
The value of $[\![\cdot]\!]$ on all other arrows is then fixed by the requirement that it be a traced monoidal functor[4].

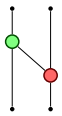
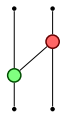
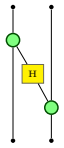
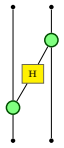
Definition 5. *The denotation of a diagram D over variables S is a superoperator constructed by summing over all the valuations of S:*

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

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

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

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

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

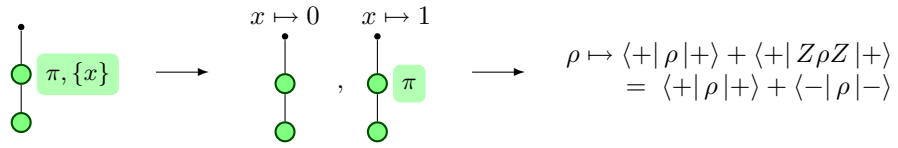
In both cases, the edge connecting the two sides of the diagram is a "virtual" qubit, representing the correlation of the two physical qubits represented by the vertical edges.

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:



Deciding whether a given edge is representing a physical or a "virtual" qubit is not obvious in general. This distinction can be made for circuit-like diagrams:

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

Remark 10. 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.

Proposition 11. *If D is unconditional and circuit-like then $[\![D]\!]$ is a unitary embedding.*

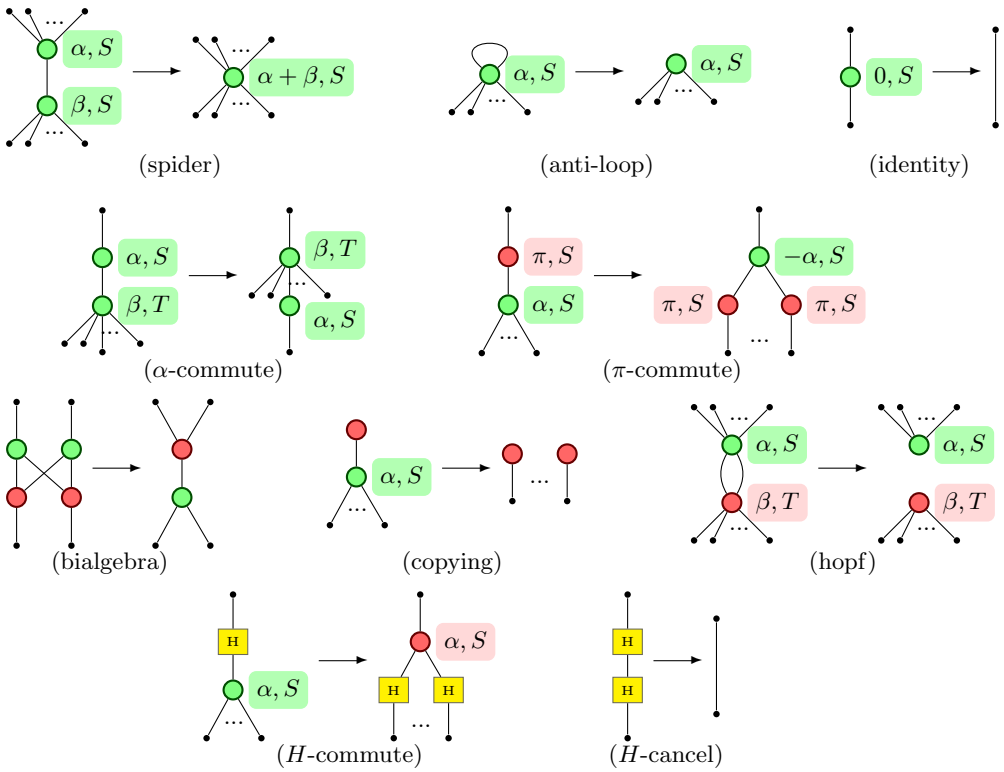
3 Rewrites

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 12. *Let R be least transitive and reflexive relation on $\mathbb{D}$ generated by the local rewrite rules shown in Figure 2.*

The diagrammatic syntax, and the equations of the rewrite rules are derived from those introduced in [4]. Space does not permit a more thorough justification of these particular operations and rules, beyond the following:

Proposition 13. *The rewrite system R is sound with respect to the interpretation map $[\![\cdot]\!]$.*



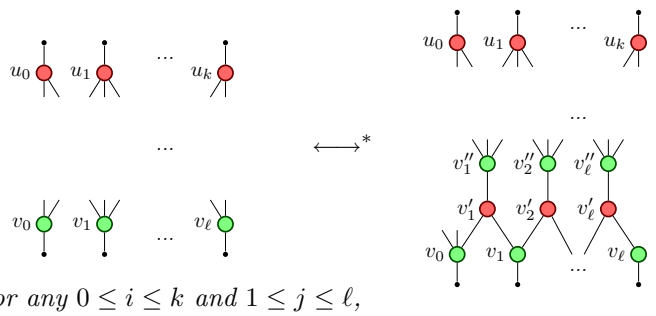
We present the rules for the "green" subsystem; to obtain the complete set of rules exchange red and green in the rules shown above.

Fig. 2. Rewrite rules for system R

Despite its soundness, this rewrite system does not have many good properties: it is manifestly not terminating since several of the rules are reversible. The subsystem without H is known to be neither confluent nor terminating [16]. 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 11 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 14 (Main Lemma). *Given a diagram D, any subset $U = \{u_0, \ldots u_k\}$ of red vertices of D and any subset $V = \{v_0, \ldots, v_\ell\}$ of green vertices of D, the subdiagram G induced by $U \cup V$ is equivalent to a diagram G':*



such that for any $0 \leq i \leq k$ *and* $1 \leq j \leq \ell$,

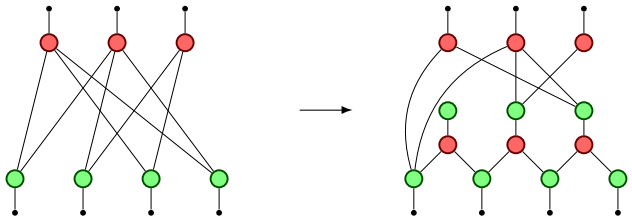
- *there is an edge* $(u_i, v_0) \in G'$ *iff* $u_i \in Odd_G(V)$
- *there is an edge* $(u_i, v''_j) \in G'$ *iff* $u_i \in Odd_G(\{v_j, \ldots v_\ell\})$

where $Odd_G(K) = \{v\,,\, |N_G(v) \cap K| = 1 \mod 2\}$ *the odd neighbourhood of* K*, i.e. the set of vertices which have an odd number of neighbours in* K*.*

Remark 15. A direct proof of the lemma is given in the appendix, although we note that it is a consequence of a theorem of Lack [17]. Note that there is an edge between a red vertex u and a green vertex v in G if and only if there is an odd number of paths between u and v in G'.

We augment the rewrite system R with the rule $G \longrightarrow G'$ (with the notation of Lemma 14). The main lemma ensures that the new system is sound with respect to the interpretation map $[\![\cdot]\!]$.

Example 16. The new rule in action:



4 The Measurement Calculus

The *measurement calculus*, introduced by Danos, Kashefi and Panangaden [5], is a formal calculus for one-way quantum computations [1, 18]. We review here the basic features of the calculus; for a complete exposition see [5].

Definition 17. *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 18. 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 19. *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.*

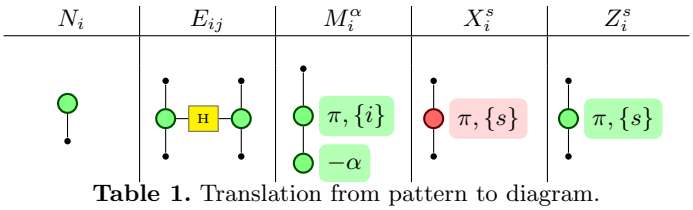
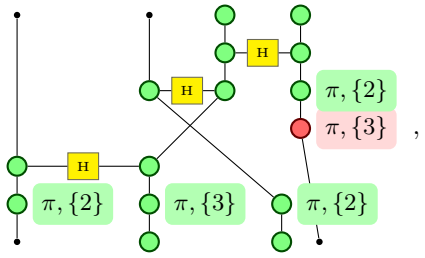


Table 1. Translation from pattern to diagram.

Example 20. 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,

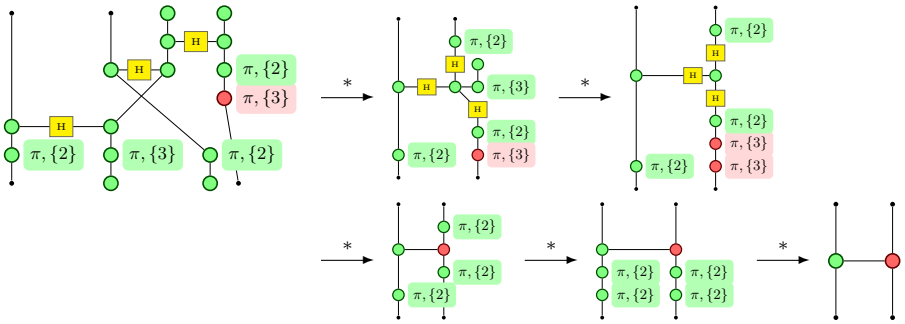
$D_{\mathfrak{P}} =$ 

,

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 11, 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 21. Returning to the CNOT pattern of Example 20, 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[5]. The possibility of performing such corrections depends on the *geometry* of the pattern, the entanglement graph implicitly defined by the pattern.

Definition 22. *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 $\mathfrak{P}$.*

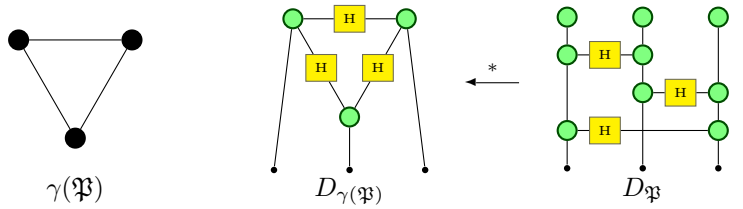
Definition 23. *Given a geometry $\Gamma = ((V, E), I, O)$ we can define a diagram $D_\Gamma = ((V_D, E_D), I_D, O_D)$ as follows:*

- *$V_D = V + E + I + O$, coloured such that:*
  - *$v \in V$ is an unconditional Z (green) 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 24. 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 $D_{\mathfrak{P}}$.

[5] Obviously, is is possible to write patterns with operations conditioned on measurements which have not yet been performed, but these do not correspond to any runnable computation.



Definition 25. *Let $\mathfrak{P}$ be a pattern in EMC form*[6] *and construct a diagram $D^m_{\mathfrak{P}}$ from $D_{\gamma(\mathfrak{P})}$ as follows.*

- *If ${}^t[M^\alpha_i]^s$ occurs in $\mathfrak{P}$ adjoin the diagram $D_{{}^t[M^\alpha_i]^s}$ at i_V as shown in Fig. 3 (a).*
- *if X^s_i (Z^s_i) appears in $\mathfrak{P}$ then compose the diagram $D_{X^s_i}$ ($D_{Z^s_i}$) to the output vertex i_O of $D_{\gamma(\mathfrak{P})}$ as shown in Fig. 3 (b).*

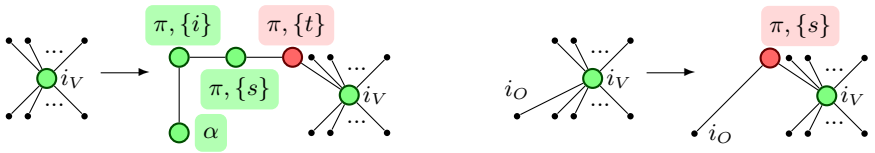


Fig. 3. Constructing $D^m_{\mathfrak{P}}$ from $D_{\gamma(\mathfrak{P})}$

Proposition 26. *Let $\mathfrak{P}$ and $D^m_{\mathfrak{P}}$ be as above; then $D_{\mathfrak{P}} \xrightarrow{*} D^m_{\mathfrak{P}}$.*

Justified by Prop. 26, 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 27. *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 $j \in \mathrm{Odd}(g(i))$ then $j = i$ or $i \prec j$;
(G3). $i \in \mathrm{Odd}(g(i))$

where $\mathrm{Odd}(K) = \{u\,, |N(u) \cap K| = 1 \mod 2\}$ is the odd neighbourhood of K, i.e. the set of vertices which have an odd number of neighbours in K.

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

[6] Every pattern is equivalent to a pattern where the commands occurs in a specific order: first the initialisations, then the entangling gates, then the measurements, and finally the X and Z corrections. See [5] for details. The rewrite rules of the measurement calculus used to prove the EMC theorem in are derivable in R. The requirement that $\mathfrak{P}$ be in EMC form is therefore not essential, but it allows a simpler statement of the proposition.

Theorem 28 ([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 the a rewrite strategy based on the existence of a gflow.

5 Rewriting to circuits

Notice that Def. 27 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.

Lemma 29. *$\mathfrak{P}$ has a causal flow if and only if $D_{\gamma(\mathfrak{P})}$ is circuit like.*

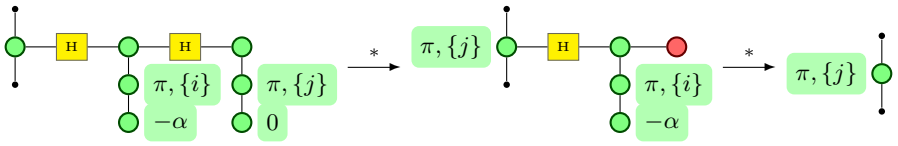
Now we demonstrate a rewriting strategy that will perform two tasks at once. If the open graph has a gflow, we will discover it. And, we will, in the process, transform the graph into a new graph which has a casual flow.

Lemma 30. *There is a normalising rewriting strategy such that if $\mathfrak{P}$ has a gflow then $D_{\gamma(\mathfrak{P})}\downarrow$ is circuit like.*

By combining Lemmas 29 and 30 we arrive at the conclusion of this section.

Theorem 31. *A geometry Γ has a gflow if and only if D_Γ can be rewritten to a circuit like diagram.*

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. Contrary to gflow, the rewriting techniques can verify non-uniform properties, i.e. properties which depend on the actual measurement angles.

6 Dealing with corrections

Now we address the question of verifying a given *pattern* rather than a geometry. We demonstrate a rewriting strategy which propagates errors induced by measurements forward to annihilate their correctors, leaving behind an unconditional diagram. The strategy terminates for circuit-like diagrams.

If a diagram has a causal flow $(f, \prec)$, we annotate the path connecting i to $f(i)$ with a direction $i \to f(i)$.

Definition 32. *Let D be a diagram with a causal flow. Let C be the least transitive, reflexive relation on $\mathbb{D}(S)$ generated by the local rewrite rules shown in Figure 4, and the rewrites (spider), (anti-loop) and (id) of system R.*

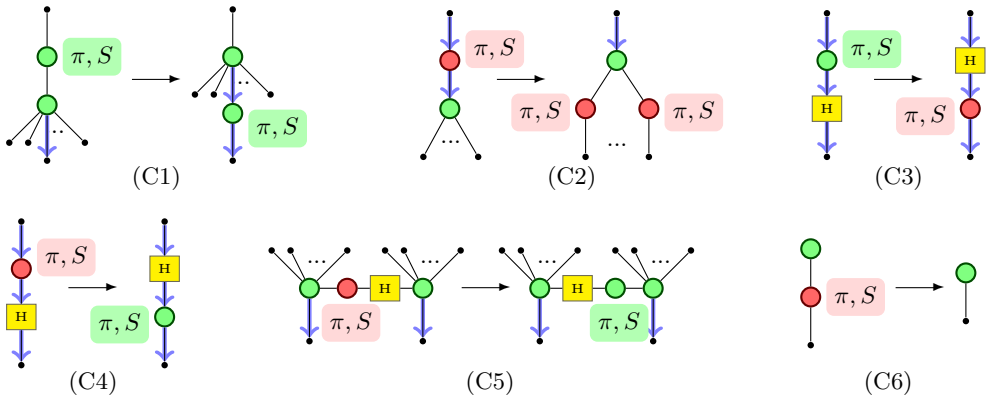


Fig. 4. Rewrite rules for system C. The signal S is required to be non-empty

System C is a restriction of R, so the interpretation map is preserved.

Proposition 33. *If $D_{\gamma(\mathfrak{P})}$ has a causal flow, then every C-rewrite sequence beginning from $D^m_{\mathfrak{P}}$ is finite and terminates in a unique normal form.*

Proposition 34. *Suppose that $D_{\gamma(\mathfrak{P})}$ can be rewritten to a circuit-like diagram; then the C-normal form of $D^m_{\mathfrak{P}}$ is unconditional if and only $\mathfrak{P}$ is deterministic.*

Remark 35. If this strategy fails to eliminate all the conditional operations then $\mathfrak{P}$ is not deterministic, but it can be made so. If we annotate the vertices to indicate to which qubit in the original pattern they correspond, then the position where the remaining conditional vertices get "stuck" shows where a correction must be added to the pattern. Alternatively, one could write the pattern omitting the corrections and use this method to determine where they should go.

7 Conclusion

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. In particular we have been able to determine if a given *pattern* is deterministic, and produce an equivalent quantum circuit which uses no ancilla qubits. These results can be extended to debug measurement patterns which are not deterministic.

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. 50th Annual Symposium on Foundation of Computer Science. (2009)
3. Browne, D.E., Kashefi, E., Perdrix, S.: Computational depth complexity of measurement-based quantum computation. (2009)
4. Coecke, B., Duncan, R.: Interacting quantum observables. In Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewic, I., eds.: Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II. Volume 5126 of Lecture Notes in Computer Science., Springer (2008) 298–310
5. Danos, V., Kashefi, E., Panangaden, P.: The measurement calculus. Journal of ACM **54**(2) (2007)
6. Danos, V., Kashefi, E.: Determinism in the one-way model. (2006)
7. Browne, D., Kashefi, E., Mhalla, M., Perdrix, S.: Generalized flow and determinism in measurement-based quantum computation. New Journal of Physics **9** (February 2007)
8. Duncan, R.: Verifying the measurement calculus by rewriting. In: Proceedings of DCM'07. (2007)
9. Kashefi, E.: Lost in translation. In: Proceedings of DCM'07. (2007)
10. Abramsky, S., Coecke, B.: A categorical semantics of quantum protocols. In: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science: 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. CRC Applied Mathematics & Nonlinear Science, Taylor and Francis (2007)
12. Coecke, B., Paquette, E.O.: POVMs and Naimark's theorem without sums. In: Proceedings of the 4th International Workshop on Quantum Programming Languages. (2006)
13. Dixon, L., Duncan, R., Kissinger, A.: Quantomatic 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. Annals of Mathematics and Artificial Intelligence **56**(1) (2009) 23–42
16. Kissinger, A.: Graph rewrite systems for complementary classical structures in †-symmetric monoidal categories. Master's thesis, Oxford University (2008)

17. Lack, S.: Composing props. Theory and Applications of Categories **13**(9) (2004) 147–163
18. Raussendorf, R., Briegel, H.J.: Computational model for the one-way quantum computer: Concepts and summary. In Leuchs, G., Beth, T., eds.: Quantum Information Processing. Wiley (2003)
19. Duncan, R., Perdrix, S.: Graph states and the necessity of euler decomposition. In Ambos-Spies, K., Löwe, B., Merkle, W., eds.: Computability in Europe: Mathematical Theory and Computational Practice (CiE'09). Volume 5635 of Lecture Notes in Computer Science., Springer (2009) 167–177
20. Mhalla, M., Perdrix, S.: Finding optimal flows efficiently. In: the 35th International Colloquium on Automata, Languages and Programming (ICALP), LNCS. Volume 5125. (2008) 857–868

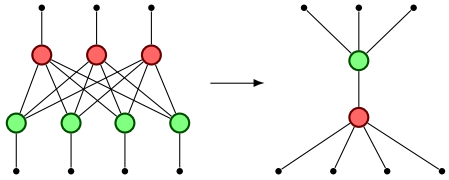
Appendix

We include proofs which the could not be accommodated in the main text due to length restrictions.

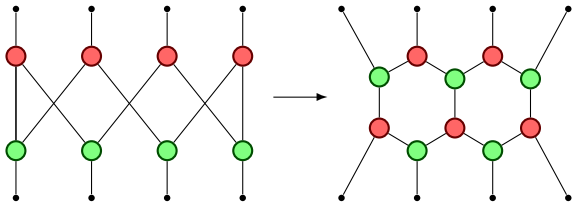
Proofs from Section 3

The following derived rules will be useful; proofs are found in [19].

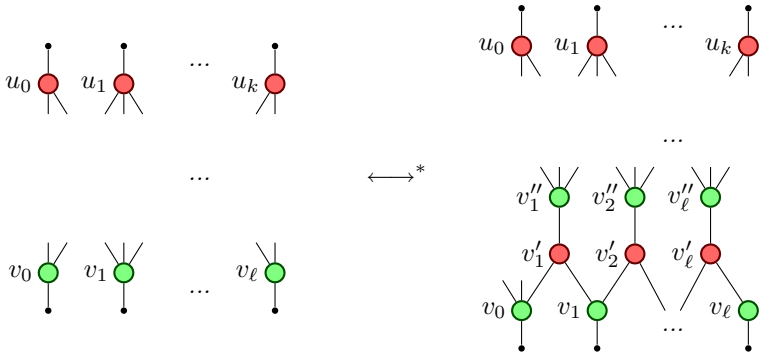
Lemma 36 (Generalised Bialgebra).



Lemma 37 (Hexagon Rule). *For n, an even cycle of size $2n$, of alternating colours, can be rewritten into hexagons. Graphically:*



Main Lemma. *Given a diagram D, any subset $U = \{u_0, \dots u_m\}$ of red vertices of D and any subset $V = \{v_0, \dots, v_\ell\}$ of green vertices of D, the subdiagram G induced by $U \cup V$ is equivalent to a diagram G':*

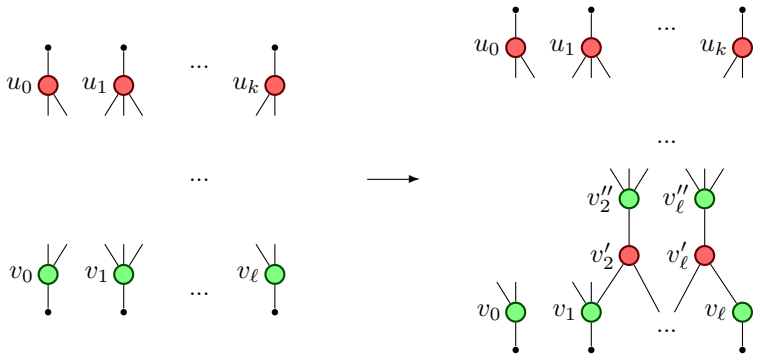


such that for any $0 \leq i \leq k$ and $1 \leq j \leq \ell$,

- *there is an edge $(u_i, v_0) \in G'$ iff $u_i \in Odd_G(V)$*
- *there is an edge $(u_i, v''_j) \in G'$ iff $u_i \in Odd_G(\{v_j, \dots v_\ell\})$*

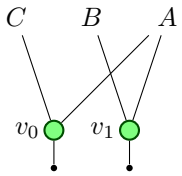
where $Odd_G(K) = \{v\,,\, |N_G(v) \cap K| = 1 \mod 2\}$ *the odd neighbourhood of* K*, i.e. the set of vertices which have an odd number of neighbours in* K*.*

Proof (of Main Lemma). The proof in by induction on the size of V. If V is of size 0 or 1 then G and G' are the same. Otherwise, by induction on $U \times \{v_1, \ldots, v_\ell\}$, G can be rewritten in G_1 as follows:

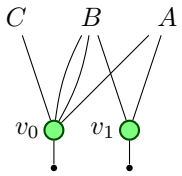


Notice that $N_{G_1}(v_0) \cap U = N_G(v_0) \cap U$, $N_{G_1}(v_1) \cap U = Odd_G(\{v_1, \ldots v_\ell\} \cap U$, and for any $i > 1$, $N_{G_1}(v_i'') = N_{G'}(v_i'')$. Now we focus on the subgraph $\{v_0, v_1\} \times (U \cap (N_{G_1}(v_0) \cup N_{G_1}(v_0))$.

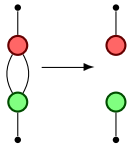
The neighbours of v_0 and v_1 in G_1 can be partitioned in three parts: the common neighbours $A = N_{G_1}(v_0) \cap N_{G_1}(v_1) \cap U$, the exclusive neighbours of v_1 $B = (N_{G_1}(v_1) \setminus A) \cap U$, and the exclusive neighbours of v_0 $C = (N_{G_1}(v_0) \setminus A) \cap U$.



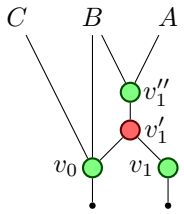
Let G_2 be the graph in which two edges are added between (u, v_0) for every $u \in B$:



Notice that $G_2 \xrightarrow{*} G_1$ using the rewriting rule



Now the generalised bialgebra (Lemma 36) is applied on the complete bipartite graph $\{v_0, v_1\} \times (B \cup A)$ in G_2, leading to G_3:

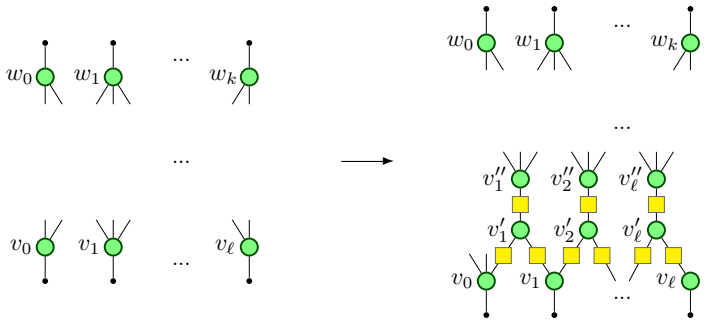


$$
\begin{aligned}
N_{G_3}(v_0) &= B \cup C \\
&= \{u \in U \mid u \notin N_G(v_0) \wedge u \in Odd_G(V \setminus v_0)\} \cup \{u \in U \mid u \in N_G(v_0) \wedge u \notin Odd_G(V \setminus v_0)\} \\
&= (N_G(v_0) \cap \overline{Odd_G(V \setminus v_0)}) \cup (\overline{N_G(v_0)} \cap Odd_G(V \setminus v_0))) \\
&= Odd_G(V) \\
N_{G_3}(v_1'') &= B \cup A \\
&= N_{G_1}(v_1) \\
&= Odd_G(V \setminus v_0) \\
&= Odd_G(v_1, \ldots, v_\ell)
\end{aligned}
$$

□

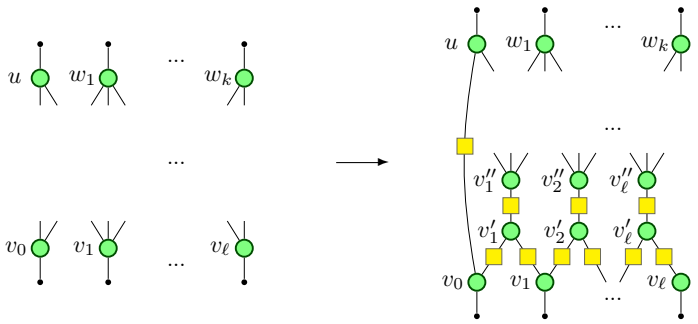
Proofs from Section 5

Proof (Proof of Lemma 30). (sketch) We use the rewriting rule induced by the main Lemma 14. Notice that in the main lemma the rewriting rule is H-free, in this proof we consider a red-free version of this rule (indeed a red vertex is a green vertex conjugated with H):



We construct the gflow of the diagram (like the efficient algorithm for finding gflow in an open graph of [20]) and simultaneously transform the diagram into a circuit-like diagram using the rewriting rule.

More precisely, the strategy is to apply the rewriting rule of the main lemma when $\{v_0, \dots v_\ell\}$ are all output vertices and $Odd(\{v_0, \dots v_\ell\}) = \{u\}$:



The existence of the gflow guarantees that such a configuration exists (see [20]). We construct the causal flow in the resulting diagram as follows: $g(u) := \{v_0\}$, $g(v_i') := \{v_i\}$, and $g'(v_i'') := \{v_i'\}$; the partial order is such that $v_\ell' \prec min(v_0, \dots v_\ell)$, $u \prec v_1'$, and for all i , $v_i' \prec v_{i+1}'$, and $v_i'' \prec v_i'$. At the end of that stage, the vertices u, v_i' and v_i'' are all considered as outputs (except u if u is an input), then the same rewriting strategy is applied until g is defined for all non-output vertices. □

Proofs from Section 6

Proof (Proof of Prop 33). The system is terminating since each rule either decreases the graph complexity (spider, anti-loop, identity, C6) or increases the position of a conditional vertex in the finite partial order given by the flow. Confluence follows from the acyclicity of the flow.

Proof (Proof of Prop 34). Note that $\mathfrak{P}$ and $D_{\mathfrak{P}}$ have the same semantics; by Lemma 26 $D_{\mathfrak{P}} \xrightarrow{*} D^m_{\mathfrak{P}}$, hence if $D^m_{\mathfrak{P}}$ can rewrite to an unconditional diagram then $\mathfrak{P}$ is deterministic.

Conversely suppose that $\mathfrak{P}$ is deterministic. Let D be the circuit-like reduct of $D_{\gamma(\mathfrak{P})}$; by Lemma 29 it has a causal flow $(f, \prec)$. Let D_m be the corresponding diagram with the measurements attached per Def 25.

Let s be a minimal (with respect to $\prec$) signal occurring as label on a conditional vertex of D_m. Let a be the least vertex (with respect to $\prec$) such that c is adjacent to a. We will procede by induction on a. By construction, initially $a = s$ and both c and a are green. We will preserve this invariant as we go.

- If a is an output qubit, then then c propagates past a via (C1); since $\mathfrak{P}$ is deterministic there must be a corrector at this output, indexed by s. This conditional vertex annihilates c via the spider and identity rules.
- Otherwise, a propagates to $f(a)$ via (C1) and (C3), and on to the neighbourhood of $f(a)$ via (C2). There are two subcases:
  - If $f(a)$ is adjacent to an output then a red correction is required here by determinism of $\mathfrak{P}$; c will annihilate this as above.

- If is not an output, it has a measurement. Determinism requires one of the following: either the measurement is conditional on s, in which case c will annihilate this corrector; or the measurement has $\alpha = 0$, in which the c can be discarded via a rewrite of type (C6).

Any other vertex b adjacent to $f(a)$ necessarily has $a \prec b$ by the flow condition. By (C5), we can rewrite each remaining copy of c to green vertex adjacent to b.

Since $\prec$ is a finite partial order this process will eventually terminate, leaving no vertex conditional on s in the diagram. By induction on s we can remove all conditional vertices this way, hence the C-normal form of D_m is unconditional.