Determinism in the one-way model

Vincent Danos
Université Paris 7 & CNRS, 175 Rue du Chevaleret, 75013 Paris, France

Elham Kashefi
Christ Church College, University of Oxford, OX1 1DP, Oxford, UK

(Dated: May 14, 2018)

We introduce a *flow condition* on open graph states (graph states with inputs and outputs) which guarantees globally deterministic behavior of a class of measurement patterns defined over them. Dependent Pauli corrections are derived for all such patterns, which equalize all computation branches, and only depend on the underlying entanglement graph and its choice of inputs and outputs.

The class of patterns having flow is stable under composition and tensorization, and has unitary embeddings as realizations. The restricted class of patterns having both flow and reverse flow, supports an operation of adjunction, and has all and only unitaries as realizations.

PACS numbers: 03.67.Lx, 03.67.-a, 03.67.Mn

I. INTRODUCTION

The recent one-way quantum computing model [1, 2, 3] has already drawn considerable attention, because it suggests different physical realizations of quantum computing [4, 5, 6, 7, 8, 9, 10, 11, 12]. However, whether this fundamentally different model may also suggest new insights in quantum information processing still stands as an open question.

Computation in this model, consists of a first phase of preparation and entanglement, followed by 1-qubit measurements and a final round of corrections. Making measurements an integral part of computation will in general induce non-deterministic behaviors. To counter this, both measurements and corrections are allowed to depend on the outcomes of previous measurements. This mechanism of feed-forwarding classical observations is known to be a necessary requirement for the model to be universal [13]. Whether and how a given pattern can be controlled so as to obtain a globally deterministic behavior is the question we address in this paper.

A variety of methods for constructing measurement patterns have been already proposed [3, 14, 15] that guarantee determinism by construction. We introduce a direct condition on open graph states (graph states with inputs and outputs) which guarantees a strong form of deterministic behavior for a class of one-way measurement patterns defined over them. Remarkably, our condition bears only on the geometric structure of the entangled graph states. This condition singles out a class of patterns with flow, which is stable under sequential and parallel compositions and is large enough to realize all unitary and unitary embedding maps.

Patterns with flow have interesting additional properties. First, they are uniformly deterministic, in the sense that no matter what the measurements angles are, the obtained set of corrections, which depends only on the underlying geometry, will make the global behavior deterministic. Second, all computation branches have equal probabilities, which means in particular these probabilities are independent of the inputs, and as a consequence, one can show that all such patterns implement unitary embeddings. Third, a more restricted class of patterns having both flow and reverse flow supports an operation of adjunction, corresponding to time-reversal of unitary operations. This smaller class implements all and only unitary transformations. Moreover, for open graph states with flow, one can derive a direct procedure for realization of unitaries as measurements patterns [16].

II. MEASUREMENT PATTERNS

We briefly recall the definition of measurement patterns and various notions of determinism. More detailed introductions can be found in [17, 18, 19]. In this paper, we will employ an algebraic approach called, the *Measurement Calculus* [13]. Computations in a pattern involve a combination of 1-qubit preparations N_i^α, 2-qubit entanglement operators $E_{ij} := \wedge Z_{ij}$ (controlled-Z), 1-qubit measurements M_i^α, and 1-qubit Pauli corrections X_i, Z_i, where i, j represent the qubits on which each of these operations apply, and α is a parameter in $[0, 2\pi)$.

Preparation N_i^α prepares qubit i in state $|+_\alpha\rangle_i$, where $|\pm_\alpha\rangle$ stand for $\frac{1}{\sqrt{2}}(|0\rangle \pm e^{i\alpha}|1\rangle)$. Measurement M_i^α is defined by orthogonal projections $|\pm_\alpha\rangle\langle\pm_\alpha|_i$, applied at qubit i, with the convention that $|+_\alpha\rangle\langle+_\alpha|_i$ corresponds to the outcome 0, while $|-_\alpha\rangle\langle-_\alpha|_i$ corresponds to 1. Note that we consider here only destructive measurements, *i.e.* a projection $|\psi\rangle\langle\psi|$ is always followed by a trace out operator and hence we might write it as $\langle\psi|$.

Qubits are measured at most once, therefore we may represent unambiguously the outcome of the measurement done at qubit j by s_j. Dependent corrections, used to control non-determinism, will be written $X_i^{s_j}$ and $Z_i^{s_j}$, with $X_i^0 = Z_i^0 = I$, $X_i^1 = X_i$, and $Z_i^1 = Z_i$.

A *measurement pattern*, or simply a pattern, is defined by the choice of V a finite set of qubits, two possibly overlapping subsets I and O determining the pattern inputs and outputs, and a finite sequence of commands acting on V.

Such a pattern is said to be *runnable* if it satisfies the following: (R0) no command depends on an outcome not yet measured, (R1) no command acts on a qubit already measured or not yet prepared (except preparation commands), and (R2) a qubit i is measured (prepared) if and only if i is not an output (input).

Write $\mathfrak{H}_I$ ($\mathfrak{H}_O$) for the Hilbert space spanned by the inputs (outputs). The run of a runnable pattern consists simply in executing each command in sequence. If n is the number of measurements (which by (R2) is also the number of non outputs) then the run may follow 2^n different branches. Each branch is associated with a unique binary string $\mathbf{s}$ of length n, representing the classical outcomes of the measurements along that branch, and a unique *branch map* $A_{\mathbf{s}}$ representing the linear transformation from $\mathfrak{H}_I$ to $\mathfrak{H}_O$ along that branch.

Branch maps decompose as $A_{\mathbf{s}} = C_{\mathbf{s}}\Pi_{\mathbf{s}}U$, where $C_{\mathbf{s}}$ is a unitary map over $\mathfrak{H}_O$ collecting all corrections on outputs, $\Pi_{\mathbf{s}}$ is a projection from $\mathfrak{H}_V$ to $\mathfrak{H}_O$ representing the particular measurements performed along the branch, and U is a unitary embedding from $\mathfrak{H}_I$ to $\mathfrak{H}_V$ collecting the branch preparations, and entanglements. Therefore

$$\textstyle\sum_{\mathbf{s}} A_{\mathbf{s}}^{\dagger}A_{\mathbf{s}} = \sum_{\mathbf{s}} U^{\dagger}\Pi_{\mathbf{s}}U = I$$

and $T(\rho) := \sum_{\mathbf{s}} A_{\mathbf{s}}\rho A_{\mathbf{s}}^{\dagger}$ is a trace-preserving completely-positive map (cptp-map), explicitly given as a Kraus decomposition. One says that the pattern *realizes* T.

A pattern is said to be *deterministic* if it realizes a cptp-map that sends pure states to pure states. This is equivalent to saying that branch maps are proportional, that is to say, for all $q \in \mathfrak{H}_I$ and all $\mathbf{s}_1$, $\mathbf{s}_2 \in \mathbb{Z}_2^n$, $A_{\mathbf{s}_1}(q)$ and $A_{\mathbf{s}_2}(q)$ differ only up to a scalar. A pattern is said to be *strongly deterministic* when branch maps are equal, *i.e.*, for all $\mathbf{s}_1$, $\mathbf{s}_2 \in \mathbb{Z}_2^n$, $A_{\mathbf{s}_1} = A_{\mathbf{s}_2}$. A pattern is said to be *uniformly deterministic* if it is deterministic for all values of its measurement angles.

Lemma 1 *Strongly deterministic patterns realize unitary embedding maps.*

Proof. If a pattern is strongly deterministic and realizes the map T then

$$T(\rho) = A\rho A^{\dagger}$$

with $A := 2^{n/2}A_{\mathbf{s}}$, and A must be a unitary embedding, because $\sum_{\mathbf{s}} A_{\mathbf{s}}^{\dagger}A_{\mathbf{s}} = A^{\dagger}A = I$. In such cases, one says that the pattern realizes the unitary embedding A. □

Example. Not all deterministic patterns are uniformly or strongly so. To see this, choose as command sequence $X_1^{s_2}M_2^0E_{12}N_2^0$, with $V = \{1, 2\}$, and $I = O = \{1\}$. The two branch maps are given by $A_0 = |0\rangle\langle 0|$, and $A_1 = |0\rangle\langle 1|$, so they are proportional, but distinct, and the pattern is deterministic, but not strongly so. The associated cptp-map $T(|\psi\rangle\langle\psi|) = \langle\psi, \psi\rangle|0\rangle\langle 0|$ projects any state onto $|0\rangle$ and does not correspond to a unitary transformation. This pattern is not uniformly deterministic either, since $\alpha = 0$ is the only angle value for M_2^{α} which makes it deterministic.

III. GEOMETRIES AND FLOWS

An *open graph states* (G, I, O) consists of an undirected graph G together with two subsets of nodes I and O, called inputs and outputs. We write V for the set of nodes in G, I^c, and O^c for the complements of I and O in V, $G(i)$ for the set of neighbors of i in G, and $E_G := \prod_{(i,j)\in G} E_{ij}$ for the global entanglement operator associated to G.

One may think of an open graph state as the beginning of the definition of a pattern, where one has already decided how many qubits will be used (V), how they will be entangled (E_G), and which will be inputs and which outputs (I and O). To complete the definition of the pattern it remains to decide which angles will be used to prepare qubits in I^c (qubits in I are given in an arbitrary states) which angles will be used to measure qubits in O^c, and most importantly, if one is interested in determinism, which dependent corrections will be used. Conversely, any pattern has a unique underlying open graph state, obtained by forgetting preparations, measurements and corrections.

For instance, the open graph state associated to the example above is the graph G with nodes $\{1, 2\}$, inputs and outputs $\{1\}$, and $E_G = E_{12}$. To complete the definition, one has to choose the angles of the measurement and preparations done at qubit 2, and define the dependent corrections.

We give a condition bearing on the geometry of open graph states, under which one can construct a set of dependent corrections such that the obtained pattern is strongly and uniformly deterministic.

Definition 2 *An open graph state* (G, I, O) *has* flow *if there exists a map* $f : O^c \to I^c$ *(from measured qubits to prepared qubits) and a partial order* $>$ *over* V *such that for all* $i \in O^c$*:*
— (F0) $(i, f(i)) \in G$
— (F1) $f(i) > i$
— (F2) for all neighbours of $f(i)$ *except* i *(*$k \in G(f(i)) \smallsetminus \{i\}$*), we also have* $k > i$

As one can see, a flow consists of two structures: a function f over vertices and a matching partial order over vertices. In order to obtain a deterministic pattern for an open graph state with flow, dependent corrections will be defined based on function f. The order of the execution of the commands is given by the partial order induced by the flow. The matching properties between the function f and the partial order $>$ will make the obtained pattern runnable.

Figure 1 shows an open graph state together with a flow, where function f represented as arrows from O^c (measured qubits, black circles) to I^c (prepared qubits, non boxed nodes). The associated partial order is given by the labeled sets of vertices. The coarsest order $>$ such that (F1) and (F2) holds is called the *dependency order* induced by the flow, and the number of the partition sets (4 in Figure 1) is called the *depth* of the flow. In general flows may or may not exist, and are not unique either.

Theorem 1 *Suppose the open graph state* (G, I, O) *has flow* $(f, >)$*, then the pattern:*

$$\mathfrak{P}_{f,G,>,\vec{\alpha}} \; := \; \textstyle\prod_{i\in O^c}^{>} (X_{f(i)}^{s_i} \prod_{k\in G(f(i))\smallsetminus\{i\}} Z_k^{s_i} M_i^{\alpha_i}) E_G N_{I^c}^0$$

where the product follows the dependency order $>$*, is runnable, uniformly and strongly deterministic, and realizes*

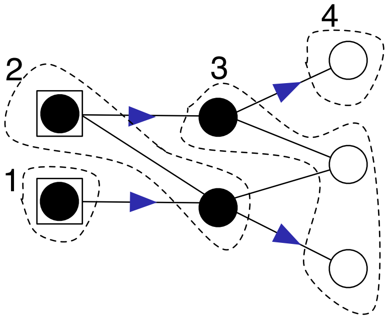

FIG. 1: An open graph state with flow. The boxed qubits are the inputs and white circles are the outputs. All the non-output qubits, black circles, will be measured during the run of the pattern. The flow function is represented as arrows and the partial order on the vertices are given by the 4 partition sets.

the unitary embedding:

$$U_{G,I,O,\vec{\alpha}} \;:=\; (\textstyle\prod_{i\in O^c}\langle +_{\alpha_i}|_i)\, E_G N^0_{I^c}$$

Proof. The proof is based on the following equations, where s stands for any arbitrary s_j:

$$\begin{aligned}
\langle +_\alpha|_i &= M_i^\alpha Z_i^{s_i} && (1)\\
Z_i^s E_{ij} &= X_j^s E_{ij} X_j^s && (2)\\
X_i^s E_{ij} &= E_{ij} Z_j^s X_i^s && (3)\\
Z_i^s E_{ij} &= E_{ij} Z_i^s && (4)\\
X_i^s N_i^0 &= N_i^0 && (5)
\end{aligned}$$

Equation (1) amounts to saying that $Z_i|\pm_\alpha\rangle_i = |\mp_\alpha\rangle_i$; notice also that this property uniquely defines Z. Equations (2), (3), and (4) come from the fact that $\wedge Z$ is in the normalizer of the Pauli group, and are easy to verify. Equation (5) is obvious. From (1) we obtain:

$$\textstyle\prod_{i\in O^c}\langle +_\alpha|_i E_G N^0_{I^c} \;=_{(1)}\; (\prod_{i\in O^c} M_i^{\alpha_i} Z_i^{s_i}) E_G N^0_{I^c}$$

so the right hand side is clearly a deterministic pattern, but just as clearly it violates condition (R0), since $Z_i^{s_i}$ depends on a measurement which has not been done yet. At that point, entanglement comes to rescue. Write $G(i)^c$ for the graph obtained by removing $G(i)$ from G. Then we can rewrite the above pattern as follows, where boxes represent the part to which we apply the rewriting equations:

$$\begin{array}{ll}
Z_i^{s_i} E_G N^0_{I^c} & = \\
Z_i^{s_i} E_{G(i)} E_{G(i)^c} N^0_{I^c} & = \\
\boxed{Z_i^{s_i} E_{if(i)}} (\prod_{k\in G(f(i))\smallsetminus\{i\}}) E_{f(i)k} E_{G(f(i))^c} N^0_{I^c} & =_{(2)} \\
X_{f(i)}^{s_i} E_{if(i)} \boxed{X_{f(i)}^{s_i} (\prod_{k\in G(f(i))\smallsetminus\{i\}} E_{f(i)k}}) E_{G(f(i))^c} N^0_{I^c} & =_{(3)} \\
X_{f(i)}^{s_i} \boxed{E_{G(f(i))} (\prod_{k\in G(f(i))\smallsetminus\{i\}} Z_k^{s_i})} X_{f(i)}^{s_i} E_{G(f(i))^c} N^0_{I^c} & =_{(4)} \\
X_{f(i)}^{s_i} (\prod_{k\in G(f(i))\smallsetminus\{i\}} Z_k^{s_i}) E_G \boxed{X_{f(i)}^{s_i} N^0_{I^c}} & =_{(5)} \\
X_{f(i)}^{s_i} (\prod_{k\in G(f(i))\smallsetminus\{i\}} Z_k^{s_i}) E_G N^0_{I^c} &
\end{array}$$

Condition (F0) is used in the third step. Finally:

$$\begin{array}{l}
\prod_{i\in O^c}\langle +_\alpha|_i E_G N^0_{I^c} = \\
(\prod_{i\in O^c} X_{f(i)}^{s_i} (\prod_{k\in G(f(i))\smallsetminus\{i\}} Z_k^{s_i}) M_i^{\alpha_i}) E_G N^0_{I^c}
\end{array}$$

By conditions (F1) and (F2) the obtained pattern is runnable, since the product can always be ordered according to $>$. Moreover, by the last equation, all branch maps are equal, and therefore the pattern is strongly deterministic. Finally, since the proof uses nowhere the particular values of the measurement angles α_i, it is also uniformly so. □

The intuition of the proof is that Equation 2 converts an anachronical Z correction at i, given in the term $M_i^\alpha Z_i^{s_i}$, into a pair of a 'future' X correction, the one sent to $f(i)$ (so in the future, by condition (F1)) and a 'past' X correction, sent to the past, until it reaches a preparation, where it is absorbed because of Equation 5.

Note that the unitary embedding associated to $\mathfrak{P}_{f,G}$ (we drop $\vec{\alpha}$ and $>$, for simplicity) does not depend on the flow. Yet, the choice of $(f,>)$ determines the structure of the corrections used by the pattern and the order of the execution, and has therefore an influence on its depth complexity, which is the depth of the flow.

Another thing worth noticing, is that using the graph stabilizer [14, 20] at i, defined as $K_{G(i)} := X_i(\prod_{j\in G(i)} Z_j)$, the pattern $\mathfrak{P}_{f,G}$ can be equivalently written as:

$$\mathfrak{P}_{f,G} \;=\; \textstyle\prod^{>}_{i\in O^c}(M_i^{\alpha_i} Z_i^{s_i} K^{s_i}_{G(f(i))}) E_G N^0_{I^c}$$

and the above proof can be reread in terms of stabilizers. In another word, for cancelling an anachronical Z correction at i it is enough to apply the dependent stablizer at qubit $f(i)$, $K^{s_i}_{G(f(i))}$ and again conditions $(F1)$ and $(F2)$ guarantee that the obtained pattern is runnable.

A. Pauli Measurements

As we saw before, not all open graph states have flow. Figure 2 shows such an example, let f be a candidate flow function, then the only choice for $f(a)$ is node c, same is true for $f(b)$. Now from condition (F2) node b must be in the future of node a and vice versa. Hence we reach a causality conflict.

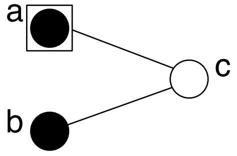

FIG. 2: An open graph state with no flow, since for any candidate function f we have $f(a) = f(b) = c$ and therefore there exists no matching partial order as a should be in future of b and b in future of a.

However, one can still obtain a deterministic pattern for the open graph state in Figure 2 by fixing the angle of the measurement of node b to be $\pi/2$. To see why, recall that Condition (F1) forbids $f(i) = i$, yet, in the special case where qubit i is measured with angle $\frac{\pi}{2}$ (Pauli Y measurement), choosing $f(i)$ to be i will work, since:

$$M_i^{\frac{\pi}{2}} X_i^s = M_i^{\frac{\pi}{2}} Z_i^s$$

Hence to correct the Y measurement at qubit i one can apply the dependent stabilizer, $(Z_i(\prod_{j\in G(i)} Z_j))^{s_i}$, at the same qubit i instead of a neighboring qubit, Figure 3. However the obtained pattern is deterministic only if qubit b is measured with angle $\frac{\pi}{2}$, and is therefore not uniformly deterministic.

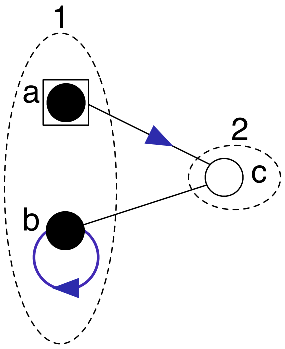

FIG. 3: An open graph state where the node with a loop, b must be measured with a Pauli Y measurement. The matching partial order has two levels given by the doted partitions.

Note that in the above example we fixed $f(b) = b$ but condition $F(3)$ still need to be verified. And this is indeed the case since in the given partial order the qubit c which is neighbour of qubit b is in the next level. To make this point clear consider the open graph state in Figure 4.

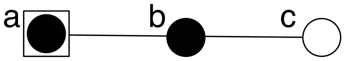

FIG. 4: An open graph sate where node b will be measured with Y measurement however one cannot apply the special case described above.

The only choice for $f(a)$ is b and hence $a < b$ but then letting $f(b) = b$ will violate the $F(3)$ condition. Therefore the only solution is to consider Y measurement as an arbitrary measurement then we obtain a flow, Figure 5.

Another special case is when qubit $f(i)$ is measured with angle 0 (Pauli X measurement). Again the requirement that $f(i) > i$ can be dropped because:

$$M_i^0 X_i^s = M_i^0$$

Therefore in the flow construction where the neighboring qubit $f(i)$ receives $X_{f(i)}^{s_i}$, if it is measured with angle 0 this correction can be ignored.

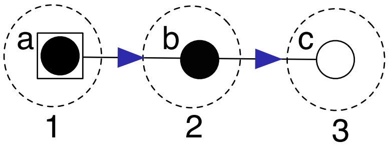

FIG. 5: An open graph sate where node b will be measured with Y with a corresponding flow without any loop.

The special cases of Pauli measurements can be related to the fact that Pauli measurements transform one graph state to another one [14]. Hence one can observe that for open graph states without flow, there might exists a set of Pauli measurements that transform it to one with flow.

B. Circuit Decomposition

Flow also provides a decomposition into simple building blocks, called *star patterns*, from which one can derive a corresponding circuit implementation of the pattern. Define the star pattern $\mathfrak{S}(n,\alpha)$ as:

$$X_2^{s_1} M_1^\alpha E_{12} E_{13} \cdots E_{1n}$$

where 1 is the only input and $2, \cdots, n$ are the outputs, for $n \geq 2$. The underlying graph has a simple flow function with $f(1) = 2$ and a two level partial order (see Figure 6). It is easy to verify that the Star pattern implements the unitary given by the circuit in Figure 7.

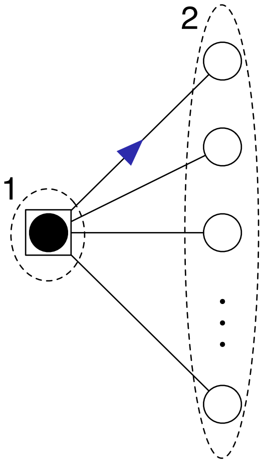

FIG. 6: Star pattern $\mathfrak{S}(n,\alpha)$ with one input, the boxed circle, and n outputs, white circles. The input qubit will be measured with angle α and one of the outputs receives a dependent correction X^{s_1}. The flow is given by the single arrow from the input to one of the outputs and two level partial order.

Every pattern such that the underlying open graph state has flow can be decomposed into star patterns. The construction

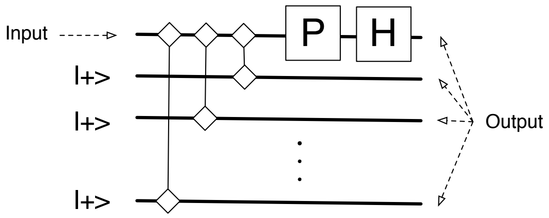

FIG. 7: The circuit implementation of Star pattern in Figure 6 , with controlled-Z, phase $P(-\alpha)$ and Hadamard H gates.

starts by picking a qubit in the first level of the partial order, exhausts all qubits in the first level before going to the next level. Each time a qubit i is picked the associated star pattern S_i is taken to have i as input, and all its remaining current neighbours as outputs. Then we remove this qubit from the graph and carry on the construction till we reach to the final level of partial order. The final deterministic pattern is the sequential and tensor composition of the obtained star patterns with the final $\wedge Z$ between the output qubits:

$$\mathfrak{P} = \prod_{m,n\in O} E_{mn} \prod^{>}_{i\in O^c} S_i$$

Now each Start pattern can be replaced by its corresponding circuit to give a circuit decomposition for the pattern $\mathfrak{P}$. In the obtained circuit each wire represents either an input qubit or an auxillary one prepared in $|+\rangle$ state, where the case is determined during the above construction. This construction can be easily formalized.

IV. ALGEBRAIC STRUCTURE

As yet, Theorem 1 is only valid when preparations are all of the form N_i^0 since Equation (5) in the proof is valid only for such preparations. Define $X_i^\alpha = Z_i^\alpha X_i Z_i^{-\alpha}$, with Z_i^α the phase operator with angle α applied at i. One has $Z_i = Z_i^\pi$. To handle general phase preparations, one only needs the analog of equations (2), (3) and (5):

$$\begin{array}{rcl} Z_i^s E_{ij} & = & (X_i^\alpha)^s E_{ij} (X_i^\alpha)^s \\ (X_i^\alpha)^s E_{ij} & = & E_{ij} Z_j^s (X_i^\alpha)^s \\ (X_j^\alpha)^s N_i^\alpha & = & N_i^\alpha \end{array}$$

and now Theorem 1 works as before. Note that we had to extend the set of corrections to include X_i^α. This extension will prove natural below, when we deal with adjunction.

Say an open graph state (G, I, O) has *bi-flow*, if both (G, I, O) and its dual state (G, O, I) have flow. Say a pattern has flow (bi-flow) if its underlying open graph state does.

The class of patterns with flows (bi-flows) is closed under composition and tensorization. It is also universal, in the sense that all unitaries can be realised within this class. This follows from the existence of a set of generating patterns having bi-flow [21].

Figure 8 shows the open graph state corresponding to a pattern realizing $\wedge U$ (controlled-U), for U an arbitrary 1-qubit unitary [21].

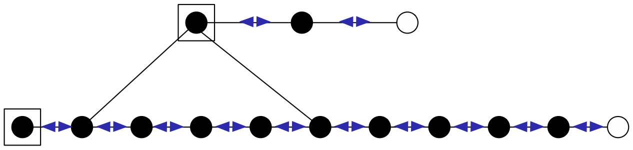

FIG. 8: An open graph state with bi-flow.

Patterns with bi-flows realize unitary operators. Indeed, by (F2), a flow $(f, >)$ is one-to-one. Therefore the orbits $f^n(i)$ for $i \in I$ define an injection from I into O. In the case of a bi-flow, I and O are therefore in bijection, and since one knows already that patterns with flows realize unitary embeddings, it follows that patterns with bi-flow implement unitaries.

Interestingly, one can define directly the adjoint of a pattern in the subcategory of patterns with bi-flows. Specifically, given $(f, >)$ a flow for (G, I, O), and angles $\{\alpha_i; i \in I^c\}$ for preparations, and $\{\beta_j; j \in O^c\}$ for measurements, we write $\mathfrak{P}_{f,G,\vec{\alpha},\vec{\beta}}$ for the pattern obtained as in the extension to general preparations of Theorem 1. Suppose a reverse flow $(g, >)$ is given on (G, O, I), one can define:

$$\mathfrak{P}^\dagger_{f,G,\vec{\alpha},\vec{\beta}} := \mathfrak{P}_{g,G,\vec{\beta},\vec{\alpha}}$$

There are two things to note here: first, for this definition to make sense, one needs to have general preparations as we described above; second, this adjunction operation depends on the choice of a reverse flow $(g, >)$. It is easy to see that $\mathfrak{P}^\dagger_{f,G,\alpha,\beta}$ and $\mathfrak{P}_{f,G,\beta,\alpha}$ realize adjoint unitaries.

An example is the pattern $\mathfrak{H} := X_2^{s_1} M_1^0 E_{12} N_2^0$ with $I = \{1\}$ and $O = \{2\}$. It has a unique bi-flow, and is self-adjoint in the sense that $\mathfrak{H}^\dagger = \mathfrak{H}$, therefore it must realize a self-adjoint operator, and indeed it realizes the Hadamard transformation.

V. CONCLUSION

Whereas the one-way model had been mostly thought of in relation with the traditional circuit model, we have proposed here a flow condition, which is clearly divorced from the circuit model, and guarantees the existence of a set of Pauli corrections obtaining a (strongly and uniformly) deterministic behavior. In essence, while dealing with patterns with flow, one can wholly forget about corrections, and think of measurements as being simply projections. This in turn may help in revealing the new perspective on quantum computing which is implicit in measurement based models. Following this work, a polynomiual time algorithm for finding flow was proposed in [22] which then extended to an algorithmic method for circuit design for unitaries thoroughly based on the one-way model [16]. Furthermore one can see that given an open graph state as a resource for computation, flow condition characterizes the set of all unitaries implementable on that given state.

If one is ready to lose uniform determinism, this condition can be somewhat extended when dealing with Pauli measurements. It may be however that strong and uniform determinism is an interesting property, when it comes to fault-tolerant computing in the one-way model.

Another point worth making is that the notion of flow gives a better understanding of why X^α, Z corrections and N^α preparations are needed. From the point of view of our determinism theorem (Theorem 1), they represent a natural and universal way to control the non deterministic evolutions induced by 1-qubit $X-Y$ measurements on a graph state.

Finally, although the obtained class of patterns with flow is universal, it remains to be seen whether this condition is also necessary for determinism. One also need to extend the flow condition to deal with $X-Z$ and $Y-Z$ plane measurements, which are the topics of our future work.

Acknowledgments

The authors wish to thank Niel de Beaudrap, Anne Broadbent, Ignacio Cirac, Paul Dumais, Damian Markham, Keiji Matsumoto for useful comments and discussions. EK was partially supported by the ARDA, MITACS, ORDCF, and CFI projects during her stay at Institute for Quantum Computing at University of Waterloo.

[1] R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Physical Review Letters*, 86:5188–5191, 2001.

[2] R. Raussendorf and H. Briegel. Computational model underlying the one-way quantum computer. *Quantum Information & Computation*, 2:443–486, 2002.

[3] R. Raussendorf, D. E. Browne, and H.-J. Briegel. Measurement-based quantum computation on cluster states. *Phys. Rev. A*, 68(022312), 2003.

[4] M. A. Nielsen. Optical quantum computation using cluster states. *Physical Review Letters*, 93:040503, 2004.

[5] S. R. Clark, C. Moura Alves, and D. Jaksch. Efficient generation of graph states for quantum computation. *New Journal of Physics*, 7:124, 2005.

[6] D. E. Browne and T. Rudolph. Resource-efficient linear optical quantum computation. *Physical Review Letters*, 95:010501, 2005.

[7] M. S. Tame, M. Paternostro, M. S. Kim, and V. Vedral. Toward a more economical cluster state quantum computation. quant-ph/0412156, 2004.

[8] M. S. Tame, M. Paternostro, M. S. Kim, and V. Vedral. Natural three-qubit interactions in one-way quantum computing. *Physical Review A*, 73:022309, 2006.

[9] P. Walther, K. J. Resch, T. Rudolph, E. Schenck, H. Weinfurter, V. Vedral, M. Aspelmeyer, and A. Zeilinger. Experimental one-way quantum computing. *Nature*, 434:169–176, March 2005.

[10] A. Kay, J. K. Pachos, and C. S. Adams. Graph-state preparation and quantum computation with global addressing of optical lattices. *Physical Review A*, 73:022310, 2006.

[11] S. C. Benjamin, J. Eisert, and T. M. Stace. Optical generation of matter qubit graph states. *New Journal of Physics*, 7:194, 2005.

[12] Q. Chen, J. Cheng, K.-L Wang, and J. Du. Efficient construction of two-dimensional cluster states with probabilistic quantum gates. *Physical Review A*, 73:012303, 2006.

[13] V. Danos, E. Kashefi, and P. Panangaden. The measurement calculus. quant-ph/0412135, 2004.

[14] M. Hein, J. Eisert, and H.-J. Briegel. Multi-party entanglement in graph states. *Phys. Rev. A*, 69:62311, 2004.

[15] A. M. Childs, D. W. Leung, and M. A. Nielsen. Unified derivations of measurement-based schemes for quantum computation. quant-ph/0404132, 2004.

[16] N. de Beaudrap, V. Danos, and E. Kashefi. Phase map decomposition for unitaries. quant-ph/0603266, 2006.

[17] Michael A. Nielsen. Cluster-state quantum computation. 2005. quant-ph/0504097.

[18] Richard Jozsa. An introduction to measurement based quantum computation. 2005. quant-ph/0508124.

[19] Dan E. Browne and Hans J. Briegel. One-way quantum computation - a tutorial introduction. 2006. quant-ph/0603226.

[20] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

[21] V. Danos, E. Kashefi, and P. Panangaden. Parsimonious and robust realizations of unitary maps in the one-way model. *Physical Review A*, 72, 2005.

[22] N. de Beaudrap. Characterizing & constructing flows in the one-way measurement model in terms of disjoint $i-o$ paths. quant-ph/0603072, 2006.