Generating one-way computations with flow: flow-preserving rewriting that ignores the interpretation

Miriam Backens

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

The one-way model of measurement-based quantum computing (MBQC) is a universal model of quantum computation, analogous to quantum circuits. It is a promising candidate for implementing photonic quantum computation as its structure of single-qubit measurements on an entangled resource state enables 'front-loading' the entangling gates, which are particularly difficult to implement in photonics. At the same time, the one-way model also has applications in theoretical quantum computing: its structure is much more flexible than that of quantum circuits, leading to advantages in circuit optimisation [6, 10]. This additional flexibility has also been exploited for generating ansätze for quantum architecture search and quantum machine learning [11, 8]. Throughout these applications, flow properties – which ensure deterministic implementability of a one-way computation as well as efficient translation to quantum circuits – are vital [9, 7]. A flow consists of a partial order on the measurements and a procedure for correcting undesired measurement outcomes, which must satisfy certain compatibility conditions. There are different variants, from causal flow, which characterises one-way computations that are very close to quantum circuits, via gflow and extended gflow, to Pauli flow.

The ZX-calculus is a graphical formalism for reasoning about quantum computations with applications to a variety of tasks at different levels of the quantum computational stack. It also provides a common language for one-way computations and quantum circuits, and for translations between them [10, 2]. While it is easy to translate a circuit into the ZX-calculus, translating a ZX-diagram into a quantum circuit is #P-hard in general [5]. Yet it is straightforward to bring a ZX-diagram into a 'MBQC form' which means it can be interpreted as a one-way computation, and if this computation has flow, then there is also an efficient translation to a circuit [10, 2, 18]. This means flow properties are important to anyone working with the ZX-calculus, whether or not they they are using the one-way model directly.

One major strength of the ZX-calculus are its complete equational theories: whenever two diagrams represent the same linear map, these equational theories allow one diagram to be transformed into the other entirely graphically [1, 12, 19]. Yet the traditional equational theories for the ZX-calculus do not preserve flow, or even necessarily the form of a one-way computation. Over the last years, researchers have thus developed flow-preserving rewrite rules: i.e. equations that preserve not only the interpretation of a ZX-diagram but also the MBQC form and the existence of flow [10, 2, 18, 13, 15, 4]. These have been used for a range of applications in compilation and optimisation.

In this work, we broaden the concept of flow-preserving rewriting by dropping the requirement that the interpretation has to be preserved. Instead of transforming a given computation into a more preferred form, this changes the goal to that of generating one-way computations with flow. These are needed as examples for research into the one-way model and as test cases for software that handles one-way computations, such as the library *Graphix*[1]. Rewriting without preserving the interpretation also makes the increased flexibility of one-way computations (compared to quantum circuits) available to quantum architecture search and other quantum machine learning applications. In particular, we give a set of three rewrite rules, stated using the ZX-calculus in Figure 1, that preserve flow but not necessarily the interpretation, as well as any side conditions that are required for flow-preservation. This set is very

[1] Available at `https://github.com/teamgraphix/graphix`.

close to the complete set of flow- and interpretation-preserving rewrite rules for one-way computations where all measurements are Pauli (so that the overall computation lives in the stabiliser fragment) [14].

Sketch of the one-way model Computation in the one-way model proceeds via single-qubit measurements on an entangled graph state [17]. Measurements can be of six different types: X, Y, Z denote the eigenbases of the three Pauli matrices, while the *planar measurements* XY, XZ, YZ are families of measurements in a plane of the Bloch sphere spanned by the eigenbases of two Paulis. The term 'Z-like' refers to the measurement types Z, XZ, YZ.

Whether or not a computation can be implemented deterministically depends on the underlying graph, the choice of qubits to be inputs and outputs of the computation, and the choice of measurement type for each non-output qubit; these data are combined into a tuple called a labelled open graph. A labelled open graph corresponds to a deterministic computation if and only if it has Pauli flow. If all measurements are planar, the condition for determinism is called gflow instead [7].

The set of flow-preserving rewrite rules Rule (IO) allows the extension or contraction of 'dangling wires', which corresponds to inserting or removing degree-2 vertices along a path graph beginning at an input or output vertex. That inserting or removing *pairs* of vertices in this way preserves both gflow and interpretation is implicit in Refs. [10, Equation (7)] and [2, Section 4.1]. The fact that inserting or removing *single* vertices in this way is also gflow-preserving is also known [2, Section 4.1]; it is straightforward to see that this simpler variant does not generally preserve the interpretation. Both sources work with gflow, hence without loss of generality, any new vertices arising from this rule may be considered to be planar measured.

Rule (LC) is a local complementation, an equivalence operation that complements all edges among neighbours of some chosen vertex (i.e. deleting existing edges and inserting edges that were not previously there) and applies certain local Clifford operations to the chosen vertex and the neighbours [16]. Applying four successive local complementations about the same vertex has a trivial net effect, i.e. (LC) effectively provides its own inverse. This is why the rule is expressed as directed. Local complementation always preserves the existence of flow assuming the central vertex is not an input, see Ref. [2, Lemma 4.3] for the gflow case and Ref. [18, Lemma D.15] for Pauli flow. It preserve the interpretation if phase angles are updated suitably, which leads to problems if the transformation is applied to an output or neighbours of an output. If the interpretation is not of interest, phase angles can be ignored instead. In particular, no particular care is needed for outputs or neighbours of an output in that scenario.

Rule (ZL) is the deletion or insertion of a Z-like measured vertex. Deletion of Z-like measurements always preserves the existence of flow; it preserves the interpretation only for certain angles [2, Section 4.3] [18, Section D.2]. Insertion of Pauli Z-measurements is always flow-preserving and preserves the interpretation if the angle is 0 [14, Prop. 4.1]. For insertion of XZ and YZ-measurements, flow preservation depends on the choice of neighbourhood as shown in [4, Theorems 4.3 and 5.2]. These are stated for Pauli flow but also applies to gflow if all measurements are planar.

Sketch of completeness proof We show that these three rules suffice to transform any one-way computation with Pauli flow or gflow to a trivial normal form with the same number of inputs and outputs, and conversely. This trivial normal form is a diagram in which each vertex is either a joint input-output, or only an output, and there are no edges: this is the simplest possible diagram with flow for any given numbers of inputs and outputs. The trivialisation algorithm first removes all Pauli measurements, then all XZ- and YZ-measurements, and then pairs of adjacent internal XY-measurements. Finally, it uses derived rules that enable the toggling of certain edges and the merging of vertices along other edges to get to the normal form while preserving the property that any remaining measurements are XY.

Minimality Removing any one of the rules would mean losing the ability to generate arbitrary diagrams with flow: in other words, the set is minimal. This is straightforward to see: (IO) is the only rule that can turn a simultaneous input/output into two vertices of which one is an input and non-output and the

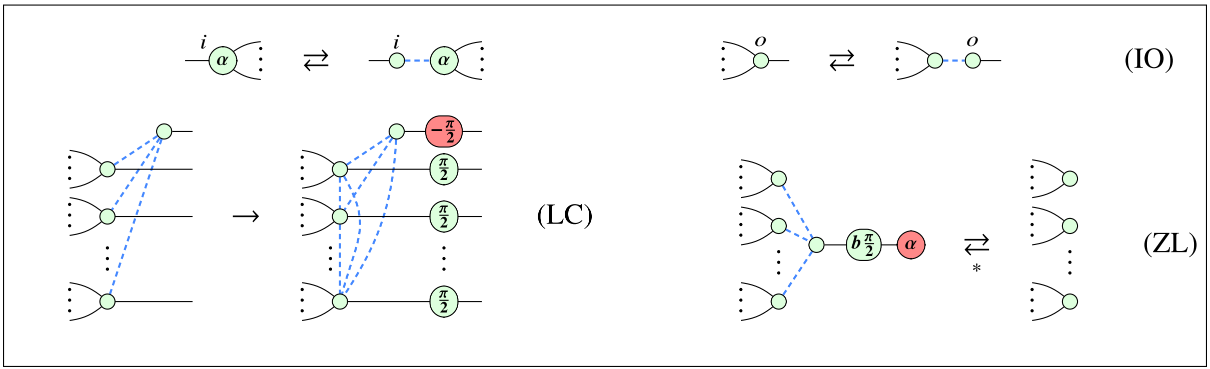

Figure 1: The complete flow-preserving (but not necessarily interpretation-preserving) rule set. The vertices marked i in (IO) are inputs and the vertices marked o are outputs. In both (IO) and (ZL), α is an arbitrary phase angle; in the latter rule moreover $b \in \{0, 1\}$. The top vertex in (LC) cannot be an input. The right-to-left application of (ZL) (marked with an asterisk) is flow-preserving only under certain conditions. All other rule directions always preserve the existence of gflow and Pauli flow.

other is an output and non-input. The local complementation rule (LC) is the only rule that can change measurement labels on vertices, which is the only way of increasing the number of XY-measured vertices with degree more than 2. Finally, the Z-like insertion and deletion rule ZL is the only rule that can change the number of connected components of the diagram.

Conclusions We have introduced a set of three rewrite rules for computations in the one-way model which preserve the existence of flow but not necessarily the interpretation. The rule set is simple and minimal. It can be used for Pauli flow and for extended gflow, where measurements can be in all three planes; both are preserved by the rules of Figure 1 (as long as (ZL) is used only to insert planar measurements in the gflow case). Moreover, it is also possible to work in the setting where all measurements are XY. This property is not preserved by individual applications of (LC) and (ZL), but these rules can nevertheless be composed into derived rewrite rules that do preserve the property. Indeed, the final steps of the trivialisation algorithm operate entirely on one-way computations where all measurements are XY, so the reverse direction would first generate only XY-measurements.

Regarding the conditions for flow preservation: the 'deletion' direction of (ZL) and both directions of (IO) preserve flow unconditionally; (LC) breaks flow only if the top vertex (the centre of the local complementation) is an input, which is straightforward to avoid. This leaves the 'insertion' direction of (ZL) as the only complicated rule. Nevertheless, as pointed out in Ref. [4], at least when working with unitary one-way computations, it is more efficient to keep track of and update the unique focused flow, rather than run a full flow-finding algorithm for each insertion.

It is interesting that our not necessarily interpretation preserving rule set is so close to the complete set of rewrite rules for one-way computations with only Pauli measurements [14]. Indeed, the additional rules of the full flow- and interpretation-preserving rule set are all concerned with handling phase labels [3], indicating a clear split between 'structural' rules that belong to the Clifford fragment and 'phase-managing' rules.

While we have shown which rules are fundamentally necessary and sufficient for generating arbitrary one-way computations with flow, their practical application has not yet been explored. Even from a more theoretical side, questions remain: for example, what strategies should be used to create interesting and useful computations with flow? Or, how should one go about generating computations with flow that satisfy certain properties of interest? We leave these questions to future work.

Acknowledgements

Thank you to Ivica Turkalj for many interesting and useful conversations during the preparation of this paper.

References

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

[2] Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski & John van de Wetering (2021): *There and Back Again: A Circuit Extraction Tale*. *Quantum* 5, p. 421, doi:10.22331/q-2021-03-25-421.

[3] Miriam Backens & Simon Perdrix (2026): *Completeness for flow-preserving rewrite rules*. To appear.

[4] Miriam Backens & Thomas Perez (2025): *Inserting Planar-Measured Qubits into MBQC Patterns While Preserving Flow*. *Electronic Proceedings in Theoretical Computer Science* 426, pp. 100–126, doi:10.4204/EPTCS.426.4.

[5] N. de Beaudrap, Aleks Kissinger & John van de Wetering (2022): *Circuit Extraction for ZX-Diagrams Can Be #P-Hard*. In Mikołaj Bojańczyk, Emanuela Merelli & David P. Woodruff, editors: 49*th International Colloquium on Automata, Languages, and Programming* (*ICALP* 2022), *Leibniz International Proceedings in Informatics* (*LIPIcs*) 229, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 119:1–119:19, doi:10.4230/LIPIcs.ICALP.2022.119.

[6] Anne Broadbent & Elham Kashefi (2009): *Parallelizing Quantum Circuits*. *Theoretical Computer Science* 410(26), pp. 2489–2510, doi:10.1016/j.tcs.2008.12.046.

[7] Daniel E. Browne, Elham Kashefi, Mehdi Mhalla & Simon Perdrix (2007): *Generalized Flow and Determinism in Measurement-Based Quantum Computation*. *New Journal of Physics* 9(8), p. 250, doi:10.1088/1367-2630/9/8/250.

[8] Luis Mantilla Calderón, Robert Raussendorf, Polina Feldmann & Dmytro Bondarenko (2025): *Measurement-Based Quantum Machine Learning*. arXiv:2405.08319.

[9] Vincent Danos & Elham Kashefi (2006): *Determinism in the One-Way Model*. *Physical Review A* 74(5), p. 052310, doi:10.1103/PhysRevA.74.052310.

[10] Ross Duncan, Aleks Kissinger, Simon Perdrix & John van de Wetering (2020): *Graph-Theoretic Simplification of Quantum Circuits with the ZX-calculus*. *Quantum* 4, p. 279, doi:10.22331/q-2020-06-04-279.

[11] Tom Ewen, Ivica Turkalj, Patrick Holzer & Mark-Oliver Wolf (2025): *Application of ZX-calculus to quantum architecture search*. *Quantum Machine Intelligence* 7(1), p. 34, doi:10.1007/s42484-025-00264-6.

[12] Amar Hadzihasanovic, Kang Feng Ng & Quanlong Wang (2018): *Two complete axiomatisations of pure-state qubit quantum computing*. In: *Proceedings of the* 33*rd Annual ACM/IEEE Symposium on Logic in Computer Science - LICS* '18, ACM Press, Oxford, United Kingdom, pp. 502–511, doi:10.1145/3209108.3209128.

[13] Tommy McElvanney & Miriam Backens (2023): *Complete Flow-Preserving Rewrite Rules for MBQC Patterns with Pauli Measurements*. *Electronic Proceedings in Theoretical Computer Science* 394, pp. 66–82, doi:10.4204/EPTCS.394.5.

[14] Tommy McElvanney & Miriam Backens (2023): *Complete Flow-Preserving Rewrite Rules for MBQC Patterns with Pauli Measurements*. *Electronic Proceedings in Theoretical Computer Science* 394, pp. 66–82, doi:10.4204/EPTCS.394.5.

[15] Tommy McElvanney & Miriam Backens (2023): *Flow-Preserving ZX-calculus Rewrite Rules for Optimisation and Obfuscation*. *Electronic Proceedings in Theoretical Computer Science* 384, pp. 203–219, doi:10.4204/EPTCS.384.12.

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

[17] Robert Raussendorf & Hans J. Briegel (2001): *A One-Way Quantum Computer*. *Physical Review Letters* 86(22), pp. 5188–5191, doi:10.1103/PhysRevLett.86.5188.

[18] Will Simmons (2021): *Relating Measurement Patterns to Circuits via Pauli Flow*. *Electronic Proceedings in Theoretical Computer Science* 343, pp. 50–101, doi:10.4204/EPTCS.343.4.

[19] Renaud Vilmart (2019): *A Near-Minimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics*. In: 2019 34*th Annual ACM/IEEE Symposium on Logic in Computer Science* (*LICS*), pp. 1–10, doi:10.1109/LICS.2019.8785765.