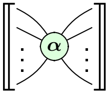
Flow-preserving ZX-calculus rewrite rules for optimisation and obfuscation

Tommy McElvanney and Miriam Backens

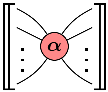
July 2023

Introduction to ZX

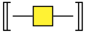
ZX diagrams are generated by green and red “spiders” corresponding to the Z-basis $\{|0\rangle, |1\rangle\}$ and the X-basis $\{|+\rangle, |-\rangle\}$ respectively, along with the (yellow) Hadamard box which is the basis change matrix between the Z and X bases.



$$= |0\rangle^{\otimes n} \langle 0|^{\otimes m} + e^{i\alpha} |1\rangle^{\otimes n} \langle 1|^{\otimes m}$$

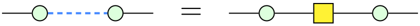


$$= |+\rangle^{\otimes n} \langle +|^{\otimes m} + e^{i\alpha} |-\rangle^{\otimes n} \langle -|^{\otimes m}$$



$$= |+\rangle \langle 0| + |-\rangle \langle 1|$$

It shall often be convenient for us to denote a hadamard gate on an edge connecting two green spiders by a dashed blue line.



ZX-diagrams

From the definitions of the generators, we are immediately able to write down the ZX-diagrams corresponding to some commonly used states and unitary maps.
Spiders with exactly one input and no outputs are states.

○— $= |0\rangle + |1\rangle \cong |+\rangle$

●— $= |+\rangle + |-\rangle \cong |0\rangle$

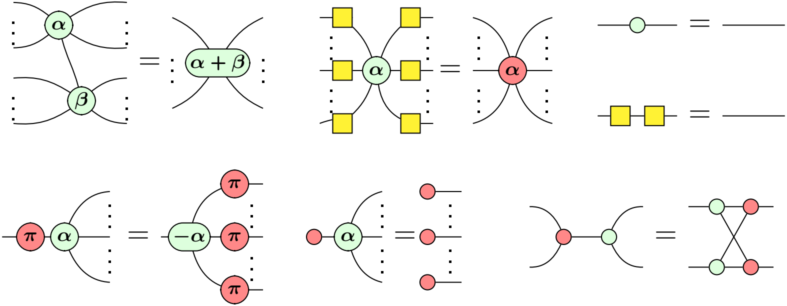
Spiders with exactly one input and output are unitary; in particular, we have the following.

$$[\![-\textcircled{\alpha}-]\!] = |0\rangle\langle 0| + e^{i\alpha}|1\rangle\langle 1| = Z_\alpha$$

$$[\![-\textcircled{\alpha}-]\!] = |+\rangle\langle +| + e^{i\alpha}|-\rangle\langle -| = X_\alpha$$

The ZX-calculus

One of the main reasons why ZX-diagrams are so powerful is due to the rewrite rules that allow us to rewrite ZX-diagrams into other diagrams implementing the same linear map. The following set of rewrite rules was proved to be complete for stabilizer quantum mechanics [Backens,2014]; that is, any two stabilizer ZX-diagrams that implement the same linear map can be rewritten into one another using the following rules.



Measurement-based Quantum computation

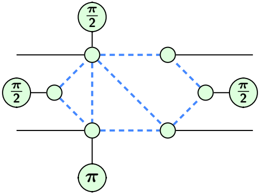
Measurement-based Quantum computation (MBQC) involves preparing a highly entangled resource state (usually a graph state) that only depends on the “size” of the computation that one wants to perform, then applying single-qubit measurements in a specific order to achieve the desired computation.

Graph states and MBQC-form diagrams

- A graph state (GS) diagram is a ZX-diagram where every spider is green and has no phase, every spider is connected to an output wire and each edge connecting spiders is a Hadamard edge.

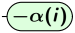
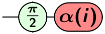
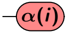
Graph states and MBQC-form diagrams

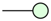
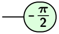
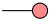
- A graph state (GS) diagram is a ZX-diagram where every spider is green and has no phase, every spider is connected to an output wire and each edge connecting spiders is a Hadamard edge.
- A MBQC-form diagram is a graph state diagram where each qubit is also allowed to be connected to an input wire, and can be connected to a measurement effect instead of its output.



Measurements in the ZX-calculus

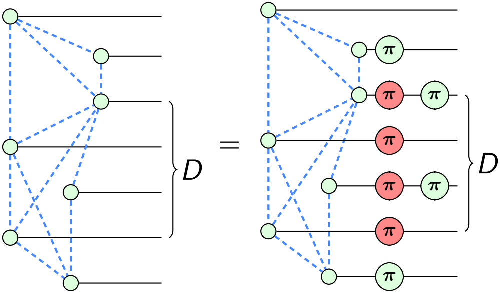
While we are able to measure qubits in arbitrary planes, we choose to restrict to measurements in the XY, XZ and YZ planes (along with Pauli measurements X, Y and Z).The ZX-diagrams corresponding to planar and Pauli measurement are given in the following tables.

operator	$\langle +_{XY,\alpha}\|$	$\langle +_{XZ,\alpha}\|$	$\langle +_{YZ,\alpha}\|$
diagram			

operator	$\langle +_{X,0}\|$	$\langle +_{Y,0}\|$	$\langle +_{Z,0}\|$
diagram			

Corrections in MBQC

Given some subset D of the vertices of a graph G, we denote the set of vertices which have an odd number of neighbours in D by $\mathrm{Odd}_G(D)$.
Corrections in MBQC rely on the following "fixed-point" property; given some graph state and any subset D of the vertices, applying $X_D Z_{\mathrm{Odd}_G(D)}$ leaves the state invariant.

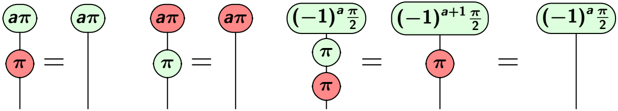


gflow

From the fixed-point property, we see that if we can find a partial order $\prec$ and a correction set function $g : V \setminus O \to 2^{V \setminus I}$ which assigns to each vertex v a set of vertices $g(v)$ (such that all of $g(v) \cup Odd_G(g(v)) \setminus \{v\}$ appear later in the partial order than v) which we can use to correct the error on v, then we can perform measurement-based computations deterministically. The pair $(g, \prec)$ is known as a gflow and the existence of a gflow is both sufficient and necessary for MBQC patterns with arbitrary planar measurements to be deterministic.

Pauli flow

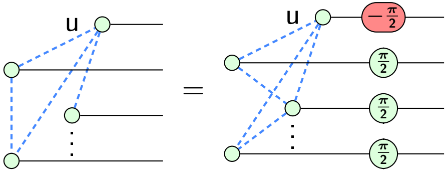
Pauli flow differs from gflow by allowing certain measurement angles to be fixed at integer multiples of $\frac{\pi}{2}$ (known as Pauli measurements). Note that the following holds for Pauli measurements;

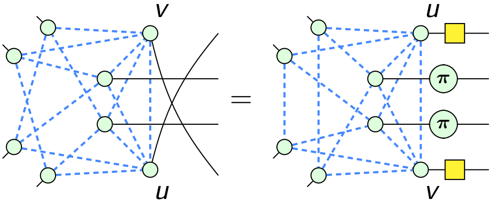


This means that we may use Pauli measured vertices to correct errors on qubits which appear later in the partial order in certain circumstances.

Local complementation and pivotting

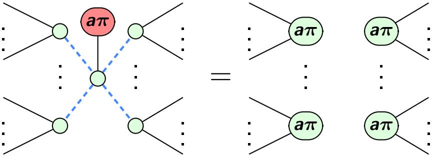
Local complementation and pivotting are two rules which have been shown to preserve the existence of gflow and Pauli flow, and have been used in circuit optimization [Duncan, 2020] [Staudacher, 2022].





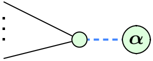
Z-deletion and insertion

[Simmons, 2021] proved that deleting Z-measured qubits preserves the existence of Pauli flow, while we [McElvanney, 2022] proved that inserting Z-measured qubits preserves Pauli flow in our submission to last year's QPL.

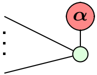


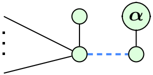
Phase gadgets in MBQC

In the ‘graph-like diagrams’ of [Duncan et al. , 2020] and [Kissinger et all. , 2020], phase-gadgets consist of a phase-free green spider connected to a 1-arity green spider via a Hadamard edge.



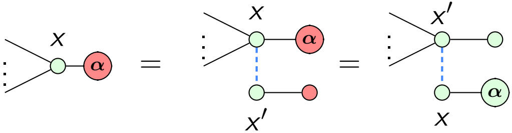
When converting from graph-like diagrams to our MBQC-form, it is unclear whether these phase-gadgets should be interpreted as a single YZ-measured vertex or as a Pauli-X measured vertex connected to a 1-arity XY-measured vertex.





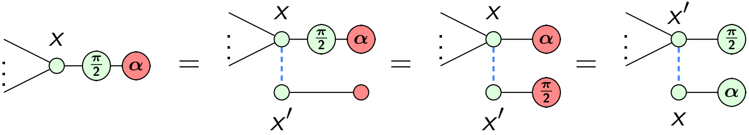
Phase gadgets in MBQC

Using a composition of the pivot rule along with Z-insertion and deletion we are able to convert between the two aforementioned interpretations of phase-gadgets in MBQC form while preserving Pauli flow as follows;



Converting XZ measurements into XY + Y

A similar sequence of rewrites allows us to represent XZ measurements in terms of just Pauli-Y measurements and XY measurements;



Every diagram is equivalent to a diagram with only XY, X and Y measurements

Using the rewrite rules introduced in the previous two slides, we can rewrite every MBQC-form diagram with Pauli flow into an equivalent diagram which only has XY, X and Y measurements while preserving the existence of Pauli flow.

- Z-delete all Pauli-Z measured vertices, leaving us with only X, Y, XY, XZ and YZ measured vertices.

Every diagram is equivalent to a diagram with only XY, X and Y measurements

Using the rewrite rules introduced in the previous two slides, we can rewrite every MBQC-form diagram with Pauli flow into an equivalent diagram which only has XY, X and Y measurements while preserving the existence of Pauli flow.

- Z-delete all Pauli-Z measured vertices, leaving us with only X, Y, XY, XZ and YZ measured vertices.
- Convert all YZ-measured vertices into X + XY using the phase-gadget rewriting rule.

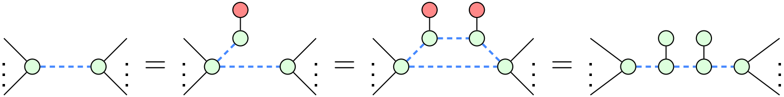
Every diagram is equivalent to a diagram with only XY, X and Y measurements

Using the rewrite rules introduced in the previous two slides, we can rewrite every MBQC-form diagram with Pauli flow into an equivalent diagram which only has XY, X and Y measurements while preserving the existence of Pauli flow.

- Z-delete all Pauli-Z measured vertices, leaving us with only X, Y, XY, XZ and YZ measured vertices.
- Convert all YZ-measured vertices into X + XY using the phase-gadget rewriting rule.
- Convert all XZ-measured vertices into Y+ XY using the rule from the previous slide.

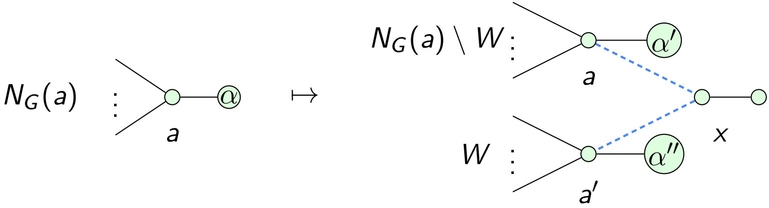
Subdividing an edge

In the blind quantum computing protocol of [Cao, 2022], rewrite rules which involve inserting new qubits while preserving flow are used for obfuscation. This paper used an unpublished rewrite rule by Backens of which we give the proof of flow-preservation below.



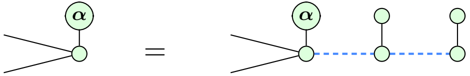
Splitting a vertex

Each of the previously mentioned flow-preserving rewrite rules are only able to change measurement angles by integer multiples of $\frac{\pi}{2}$. Here we introduce the first flow-preserving rewrite rule which allows us to change measurement angles arbitrarily.



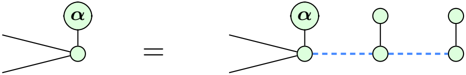
Corollaries of vertex splitting

Applying the vertex splitting rule with $W = \emptyset$ and $\alpha'' = 0$, we obtain the following rule which is also used in the blind quantum computing protocol of [Cao, 2022].

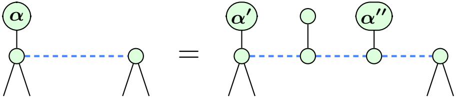


Corollaries of vertex splitting

Applying the vertex splitting rule with $W = \emptyset$ and $\alpha'' = 0$, we obtain the following rule which is also used in the blind quantum computing protocol of [Cao, 2022].



By applying vertex splitting with $|W| = 1$, we get that the following 'neighbour unfusion' rule of [Staudacher et al., 2022] preserves the existence of Pauli flow, where $\alpha = \alpha' + \alpha''$.

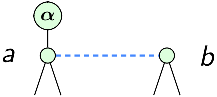


Neighbour unfusion and gflow

In [Staudacher et al., 2022], they found that neighbour unfusion preserved gflow when the two qubits involved are extracted onto the same qubit - we tried to find a more concrete characterisation of when gflow is preserved by neighbour unfusion.

A sufficient condition for neighbour unfusion to preserve gflow

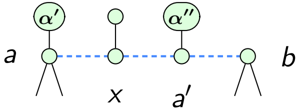
Suppose we have a diagram D with (focused) gflow $(g, \prec)$ which contains a subdiagram of the following form.



Then neighbour unfusion preserves the existence of gflow if $a \prec b$. $v > a$ implies $v > b$ and $v < b$ implies $v < a$ (or equivalently with the roles of a and b reversed).

Outline of proof of sufficiency

Consider the diagram D' with the following subdiagram obtained from applying neighbour unfusion to a and b in D.



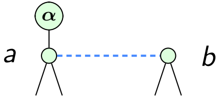
We can construct a gflow $(g', \prec')$ for D' as follows;

$$g'(v) = \begin{cases} g(v) \cup \{x, a'\} & \text{if } a \in g(v) \wedge b \in g(v) \\ g(v) \cup \{a'\} & \text{if } a \in g(v) \wedge b \notin g(v) \\ g(v) \cup \{x\} & \text{if } a \notin g(v) \wedge b \in g(v) \\ g(b) \cup \{a'\} & \text{if } v = x \\ g(a) & \text{if } v = a' \\ g(v) & \text{otherwise.} \end{cases}$$

Let the partial order $\prec'$ be the transitive closure of $\prec \cup \{(a, x), (x, a'), (a', b)\}$ - it is then simple to show that each of the gflow conditions is satisfied by $(g', \prec')$.

A necessary condition for neighbour unfusion to preserve gflow

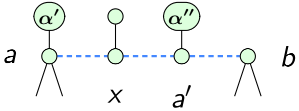
Suppose we have a diagram D satisfying $|I| = |O|$ with a focused gflow $(g, \prec)$ which contains a subdiagram of the following form.



Then $b \in g(a)$ is necessary for neighbour unfusion to preserve the existence of gflow (or equivalently $a \in g(b)$).

Outline of proof of necessity

Suppose we have a diagram D' with a focused gflow $(g', \prec')$ which has a subdiagram of the following form.



Then the diagram obtained from deleting all non-XY measured vertices from D' also has a focused gflow, and moreover this focused gflow is reversible in a strict sense [Backens et al, 2021]. Each 2-arity XY measured vertex must be in the correction set of one of its neighbours and its other neighbour must appear in its correction set. We can therefore assume WLOG that $x \in g'(a)$, $a' \in g'(x)$ and $b \in g'(a')$ - it is then somewhat simple to construct a focused Pauli flow for D where we have $b \in g(a)$, concluding the proof.

Closing remarks

Closing remarks

Thank you to the organisers and to all attending for the great conference!