Martin van IJcken

Merton College

University of Oxford

Generalized flow for hypergraph measurement patterns

A dissertation submitted for the degree of

Master of Science in Mathematical and Theoretical Physics

Trinity term 2024

Abstract

This dissertation explores the development of an analogue to generalized flow (gflow) for hypergraph measurement patterns using the ZH-calculus framework. The ZH-calculus, extends the ZX-calculus by efficiently describing Toffoli gates and multi-controlled Z gates. While the ZX-calculus and its gflow conditions are well-understood for graph states, no such analogue exists for hypergraphs. This dissertation establishes a necessary and sufficient condition for hypergraph measurement patterns to be deterministic. The new gflow criteria, analogous to those in graphs but adapted for hypergraphs, ensure that corrections are available for potential undesired measurements that may occur during execution of a measurement pattern.

Contents

1 Introduction

In 2007, Bob Coecke and Ross Duncan introduced the ZX-calculus as an elegant graphical way to allow matrix-free descriptions of quantum mechanical systems. Their work built upon earlier work using Tensor networks and Category theory [1], [2]. The benefit of the ZX-calculus was that it used only two families of tensors, with simple rewrite rules while still being able to describe any unitary linear map [3].

Additionally, the ZX-calculus comes equipped with a set of rewrite rules that are sufficient to prove equivalence between any Clifford+T diagrams [4]. Although not all unitaries can be written as a finite-sized Clifford+T diagram, it is possible to approximate any unitary up to arbitrary precision with a Clifford+T diagram [3].

The ZX-calculus is in some sense a higher level description of Quantum Mechanics than is the 'von Neumann' description [1] and has proven itself a valuable didactic tool, allowing high school students trained in the use of ZX-calculus to outperform undergraduate physics students in an undergraduate Quantum Physics exam [5]. The ZX-calculus has recently become widely used and is used extensively for the purposes of compilation, circuit optimisation [6], [7], error-correction, quantum natural language processing, quantum machine learning, and also many issues surrounding photonic quantum computing [8].

Specifically for circuit optimisation, the ZX-calculus is useful as it allows circuits to be written in ways that are not equal to a gate-based circuit but that are still somewhat similar to gates and from which gate-based circuits can be systematically extracted [7]. However, not all ZX diagrams can be turned into gate-based circuits.

In order to find which ZX diagrams can be turned into gate-based circuits, one can interpret the ZX diagrams as measurement-based quantum computing (MBQC) circuits. MBQC circuits work based on a highly-entangled resource state (which may differ per circuit), whose qubits are measured one-by-one. This is called a measuremetn pattern. As quantum measurements are inherently random, this measurement will not always give the same result and when an undesired result is measured a correction must be applied to not yet measured qubits. Such corrections are not always possible and so in 2006 the concept of flow was developed as a sufficient condition for when there will always be sufficient unmeasured qubits to correct an error that may occur [9].

Flow is a graph theoretical condition which can be applied when the resource state used by the measurement pattern is a graph-state. It essentially states that if there is always a way to apply a specific correction and an order to measure the qubits in so that that specific correction affects only not yet measured qubits, then the MBQC circuit can be deterministically ran [7], [9]. This concept of flow was then generalized to generalized flow (gflow) by Browne et al. in 2007, which is a necessary condition for measurement patterns based on graph-states to be deterministic [10].

In 2021 Miriam Backens et al. used the ZX-calculus to show that measurement patterns that have gflow can be turned into gate-based circuits using a procedure outlined in [7]. This procedure depends specifically on local complementation and pivoting operations which exist in the ZX-calculus.

The ZH-calculus was introduced in 2018 as a derivative of the ZX-calculus [11]. It is similar to the ZX-calculus in many ways but in addition to still being able to simply express any ZX-diagram as a ZH-diagram, the ZH-calculus is also able to describe Toffoli gates (CCZ) and other multi-controlled Z gates in far fewer operations than the ZX-calculus. Where ZX-diagrams can be rewritten to be in a graph-like form, ZH-diagrams can be rewritten into a hypergraph-like form. Edges between qubits being controlled Z gates and hyperedges being multi-controlled Z gates. The hypergraph form allows for a simpler description of certain 'really equally weighted' states [12], which are often the types of states used as resource states for MBQC.

Presently no analogue to gflow exists for hypergraph measurement patterns and it is not known which hypergraph measurement patterns even correspond to unitary operations (unitarity being a necessary condition on any deterministic quantum circuit). It is also not known which ZH diagrams can be turned into gate-based quantum circuits. The ZH-calculus does have rewrite rules analogous to local complementation and pivoting [13].

In this dissertation the ZH-calculus is used to develop an analogue to gflow for hypergraph measurement patterns, and it is proven that this new gflow on hypergraphs is necessary and sufficient for an hypergraph measurement pattern to be deterministic. Some minimalistic examples are investigated to see where direct applications of earlier circuit extraction algorithms break down when applied to hypergraph measurement patterns.

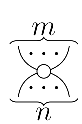
2 Preliminaries

This section introduces three preliminary topics. The ZH-calculus as a graphical language used throughout the dissertation, measurement-based quantum computation (MBQC) as a quantum computing framework which is the topic of much of the work done in this dissertation, and generalized flow as a concept from measurement-based computation which is further generalized upon in this dissertation.

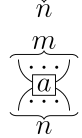
2.1 The ZH-calculus

The ZH-calculus is a diagrammatic description of complex linear maps. More well known than the ZH-calculus is the ZX-calculus which has been used extensively [8] and once familiar with either of these calculi, the other is easy to learn. Although the ZX-calculus is more frequently used, the ZH-calculus has certain benefits, such as its ability to neatly describe the Toffoli gate and other many-qubit entangling gates.

The calculus is based on two generators [11]



$$:= \underbrace{|0\dots0\rangle}_{n}\underbrace{\langle0\dots0|}_{m} + \underbrace{|1\dots1\rangle}_{n}\underbrace{\langle1\dots1|}_{m}$$



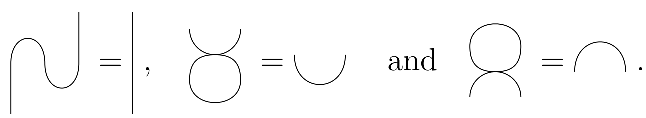
$$:= \sum a^{i_1\cdots i_n j_1\cdots j_m} |i_1\dots i_n\rangle \langle j_1\dots j_n| \,.$$

These generators are called the Z-spider and H-box respectively. The sum on the right hand side defining the H-box runs over all possible bit strings $i_1 \dots i_n \in \{0,1\}^n$, and $j_1 \dots j_m \in \{0,1\}^m$. By convention, the parameter a is omitted when it is equal to -1.

A ZH-diagram is made by taking a combination of these generators and connecting their wires. The generators can be interpreted as tensors [14], with the number of wires connecting to a generator corresponding to the rank of the tensor (in string diagrams the rank is called the 'arity') and connecting wires corresponding to contraction of the respective indices, exactly as it would in a tensor network [15]. When an input of one generator connects to the input of another, the wire that connects these two must necessarily make a shape similar to $\cup$, which is called a cup and in the tensor network language it corresponds to the tensor

$$\delta^{\mu\nu} = \begin{cases} 1 & \text{if } \mu = \nu \\ 0 & \text{otherwise} \end{cases} .$$

The equivalent for a $\cap$ is called a cap and corresponds to lowering an index. Due to their specific form, the $\cup$ and $\cap$ operator obey certain equations known as the Yanking-equations [3].

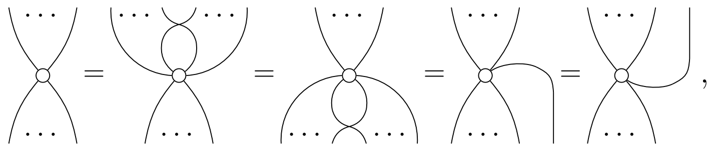


(1)

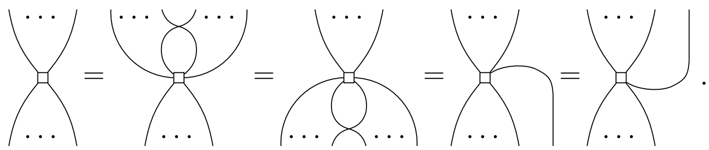
These three equations are common to all string diagrams, and not specific to the ZH-calculus.

For the specific tensors that generate the ZH-calculus, however, the distinction between inputs and outputs is only superficial as an input connected to a $\cup$ is equivalent to an output, and an output connected to a $\cap$ is equivalent to an input. The tensors are also invariant after permutation of the inputs and or outputs, as shown in Equations 2 and 3. For this reason, one may apply the paradigm of 'only connectivity matters' when discussing ZH-diagrams. The only thing that determines the structure of a ZH-diagram are the inputs, the outputs[1], the Z-spiders, the H-boxes, and the edges connecting these.

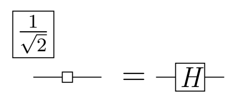
[1] That is, inputs and outputs of the diagram as a whole, not inputs and outputs of a specific tensor

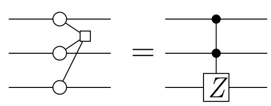
 (2)

and

 (3)

Example 1. *The ZH-calculus can be used to describe a Hadamard and a CCZ gate, using the diagrams shown in Equations 4 and 5 respectively.*

 (4)

 (5)

To confirm that these equalities hold, for the Hadamard it is sufficient to confirm that their matrices are the same. For the CCZ, it will be shown in Example 4. For the Hadamard, remember that the H-box's matrix is by definition equal to

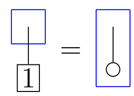
$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

and that this is equal to a Hadamard gate exactly up to the scalar $1/\sqrt{2}$.

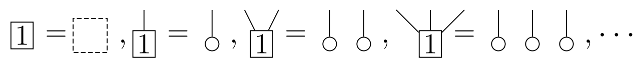
Additionally, the ZH-calculus has several rewrite rules but before those are shown it is necessary to introduce two pieces of notation. The first is the !-box (pronounced bang-box) notation. This notation is drawn as a blue box around some part of the diagram and it means that that part of the diagram is repeated a specific number of times; or potentially, when the number of repetitions is not specified, that it can be repeated an arbitrary number of times.

In equations, each !-box on one side of the equation should be paired with a corresponding !-box on the other side of that equation. Their labels can be omitted only if it is evident from context which !-box corresponds to which !-box.

Example 2. *An H-box whose parameter is equal to* 1 *is equal to a Z-spider connecting to each individual wire.*

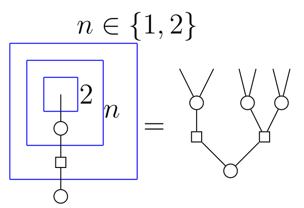


represents the family of equations



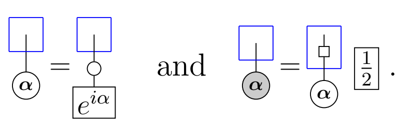
Here the !-boxes are not labelled because it is obvious from context which !-box on the LHS corresponds to which !-box on the RHS. The dashed box indicates an empty diagram.

Example 3. *It is possible for !-boxes to overlap, such as in equation 6*



(6)

The second new notation is an expansion to our set of available spiders. The previously mentioned Z-spider can be generalized to a Z-phase-spider, and can be used to derive an X-spider and X-phase-spider. The X-spiders are coloured gray and the phases are written as a real numbers α inside of the spider. When an X-spider has no phase written on it, its phase is assumed to be zero.



(7)

Using these notations, the rules of the ZH-calculus are as shown in Figure 1 and they can be used to proof the equivalence of any equivalent diagrams within the ZH-calculus [11].

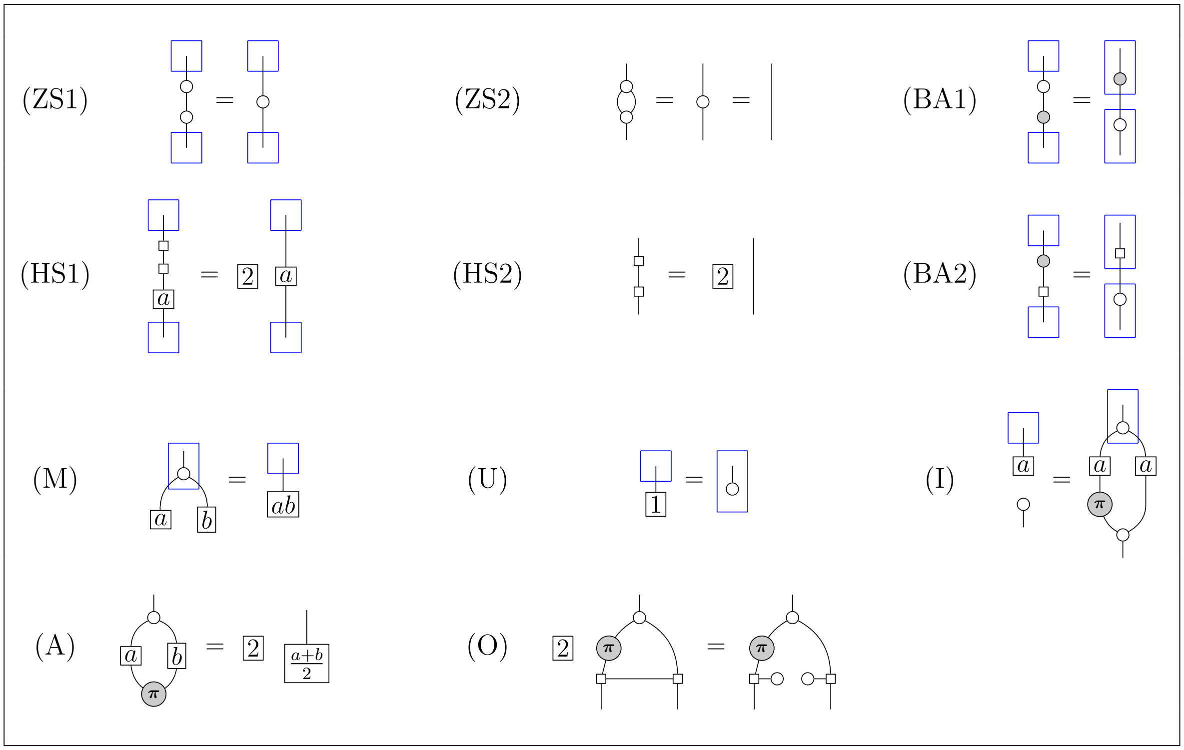
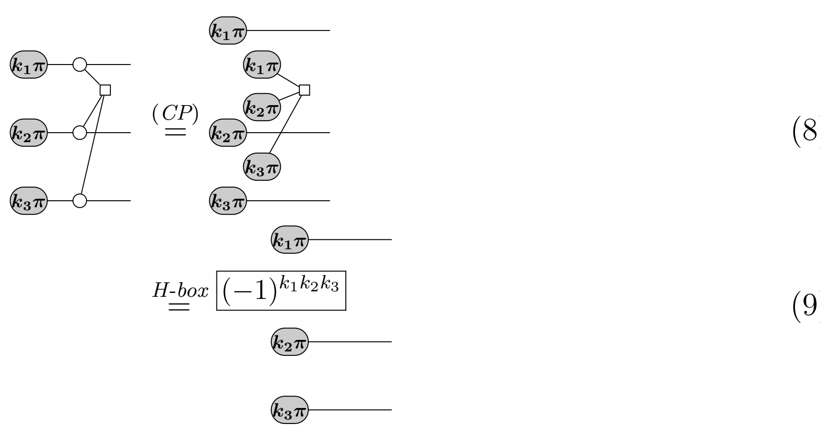


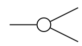
Figure 1: The rules of the ZH-calculus. Throughout, a, b are arbitrary complex numbers. These are the !-boxed versions of the rules as presented in [11]. This figure is taken from [13].

Example 4. *With this X-spider, it becomes possible to check the CCZ from Example 1. One can provide an arbitrary input from the Z-basis. Elements from the Z-basis are equal to gray phase spider with phase* 0 *for* $|0\rangle$ *and phase* π *for* $|1\rangle$.



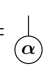
In the first step the Z-spider copies the input states. This can be proven in the ZX or ZH-

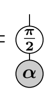
calculus [3] but for now it suffices to see this as a consequence of the definition

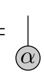
 $= |00\rangle \langle 0| + |11\rangle \langle 1|$

2.2 Measurement-based quantum computation

Multiple types of measurement-based quantum computing (MBQC) exist. This section introduces graph state MBQC, also known as the one-way model. Graph state MBQC is a potential architecture for quantum computing. The principle is to begin with a highly entangled resource state consisting of many qubits entangled with CZ gates and to then measure those qubits one-by-one. For each qubit, there is a desired result (denoted by $|+\rangle$) and an undesired result (denoted by $|-\rangle$). If the undesired result is measured, corrections must be applied to not-yet-measured qubits to bring the entire state back to the state it would have been in if the desired result had been measured. In the literature it is common to restrict the measurements to be at an arbitrary angle in one of three pre-determined planes.

$|+\rangle_{XY,\alpha} =$  (10)

$|+\rangle_{XZ,\alpha} =$  (11)

$|+\rangle_{YZ,\alpha} =$  (12)

and the undesired results denoted by $|-\rangle_{\lambda,\alpha}$ are the states orthogonal to these.

Definition 1 ([16]). *A graph measurement pattern is a tuple consisting of a register of qubits V, subsets $I, O \subset V$ of inputs and outputs, and a set of operations*

- *Preparations N_v which initialize a qubit $v \in V$ to the state $|+\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)$;*
- *Entangling operators E_{uv} which act as a controlled Z gate between the qubits $u, v \in V$;*

- *Destructive measurements $M_v^{\lambda,\alpha}$ which measure a qubit v in the λ plane at an angle α and project on either $|+\rangle_{\lambda,\alpha}$ or $|-\rangle_{\lambda,\alpha}$*

- *Corrections X_v and Z_u which act respectivly as the pauli X on the qubit $v \in V$ and the pauli Z gate on the qubit $u \in V$ and can be applied to qubits depending on the measurement outcomes of qubits measured earlier.*

A graph measurement pattern can be described as a labelled open graph, which is tuple (G, I, O, λ), where G is a graph in which each vertex represents a qubit and each edge represents a controlled Z-gate applied to its member vertices, I is the set of inputs to the circuit, O is the set of outputs to the circuit, and $\lambda : O^c \to \{XY, XZ, YZ\}$ determines the measurement plane for each non-output vertex. Additionally one needs a function $\alpha : O^c \to [0, 2\pi)$ which determines the desired measurement angle for each non-output vertex. Figure 2 shows an example (unlabelled) resource state.

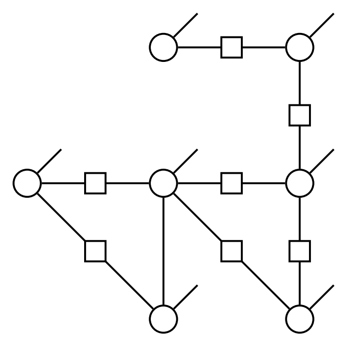


Figure 2: In this graph (state), Z-spiders take on the role of vertices (qubits), and arity-2 H-boxes function as edges (controlled-Z gates).

The MBQC framework is often extended to include local clifford gates applied to the outputs of the circuit to form an MBQC+LC graph. These local cliffords will occasionally show up in example circuits.

Example 5. *A CNOT between two states in the YZ plane can be implemented using the labelled graph-state in Figure 3.*

The implementation is done by measuring qubits at the indicated angles in order from left to right. Whenever an error occurs, this should be corrected for by appling a specific

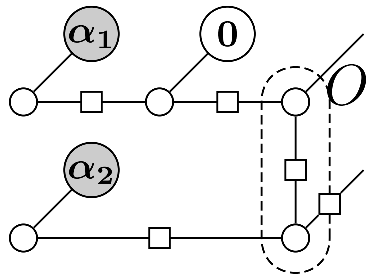
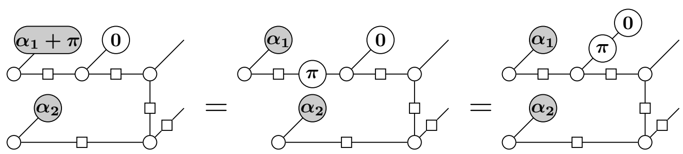


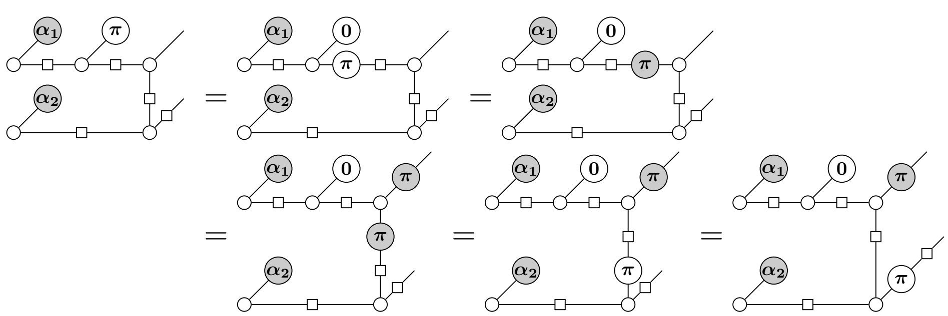
Figure 3: *This is a graph state, along with the desired measurement effects for a CNOT gate (when the inputs are in the YZ-plane). The outputs are the two vertices on the right. The Hadamard on the bottom output is a local Clifford, which must be applied after all measurements and corrections have been performed. Note that the input set for this graph-state is emtpy as all qubits are prepared, instead of measuring the left to qubits in the YZ-plane, they could be connected to inputs and measured as* 0 *in the XY-plane for a more general CNOT that take in arbitrary inputs.*

gate to one of the not-yet-measured qubits. For instance, when the top input is measured as the undesired result this can be fixed by applying a Z gate to the middle qubit, as shown in Equation 13.



(13)

Alternatively the middle qubit is measured as the undesired result, this is fixed by applying an X gate to the top output and a Z gate to the bottom output, as shown in Equation 14.



(14)

Each time a measurement is executed, different errors may occur. Different errors may cause the pattern to give different outputs each time it is ran.

Definition 2. *The specific linear map implemented by a measurement pattern for a specific set of errors is called a branch of that measurement pattern.*

Not all measurement patterns can be implemented in practice. To distinguish those that can from those that cannot, Definition 3 is used.

Definition 3 ([7]). *A measurement pattern is called runnable if*

- *No corrections depend on a qubit that has not been measured yet;*
- *No command acts on a qubit that has already been measured*
- *no input qubit is prepared;*
- *all non-input qubits are prepared exactly once;*
- *all non-output qubits are measured;*
- *a command C that is not a preparation acts only on qubits that are either inputs or that have been prepared.*

A specific branch of a measurement pattern can be written as

$$\prod_{v\in O^c}^{\prec}\left(C_v^{s_v}M_v^{\lambda(v),\alpha(v),s_v}\alpha(v)\right)E_G N_{I^c}, \tag{15}$$

where N_{I^c} prepares all qubits in I^c in the state $|+\rangle_{XY,0}$, E_G performs controlled Z gates for all edges in G, $s_v = 1$ if an error occurs at qubit v, and 0 otherwise,

$$M_v^{\lambda(v),\alpha(v),s_v} = \begin{cases} |+\rangle_{v,\lambda(v),\alpha} & \text{if} s_v = 0 \\ |-\rangle_{v,\lambda(v),\alpha} & \text{if} s_v = 1 \end{cases}$$

is a measurement of qubit v in the plane $\lambda(v)$ at the angle $\alpha(v)$, and C_v is a unitary correction for a potential error at v i.e. $C_v \langle -|_{v,\lambda(v),\alpha(v)} E_G N_{I^c} \propto \langle -|_{v,\lambda(v),\alpha(v)} E_G N_{I^c}$, C_v acts like identity when restricted to those vertices that are not successors of v in the partial order $\prec$ (except v itself), and $\prod^{\prec}$ is a product of the terms in an order that respects the partial order $\prec$ so that $v_1 \prec v_2$ implies that v_1 will be measured before v_2.

Definition 4. *A measurement pattern is called deterministic if all branches differ only up to a global scalar. It is strongly deterministic if all branches differ only up to a global phase, it is uniformly deterministic if it is deterministic for any choice of measurement angles, and it is stepwise deterministic if it remains deterministic when arbitrary qubits are turned into outputs.*

There exist measurement patterns which are not deterministic.

Example 6 (Page 16 of [7])**.** *The measurement pattern associated to the labelled open graph state shown in Figure 4 is not deterministic. An undesired result on the first input measured can be corrected with an X gate to one of the outputs and a Z gate to the other input, but an undesired result on the second input to be measured cannot be corrected, as it would require a Z gate on the first input, which has already been measured. It can also be confirmed that this*

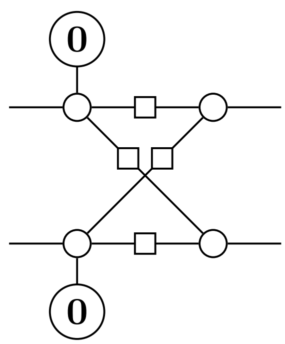


Figure 4: *This resource state, with the left qubits as inputs measured in the XY plane and the right as outputs cannot be executed deterministically. The left two vertices are inputs and the right two are outputs.*

measurement pattern should not be deterministic as it would correspond to the linear map

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

which is not unitary.

2.3 Generalized flow for graph measurement patterns

The corrections from the previous section applied after undesired results are measured such as in Example 5 can be formalised using a concept called generalized-flow (gflow). Before introducing generalized flow it is useful to consider (causal) flow. Causal flow is defined when all qubits are measured in the XY-plane.

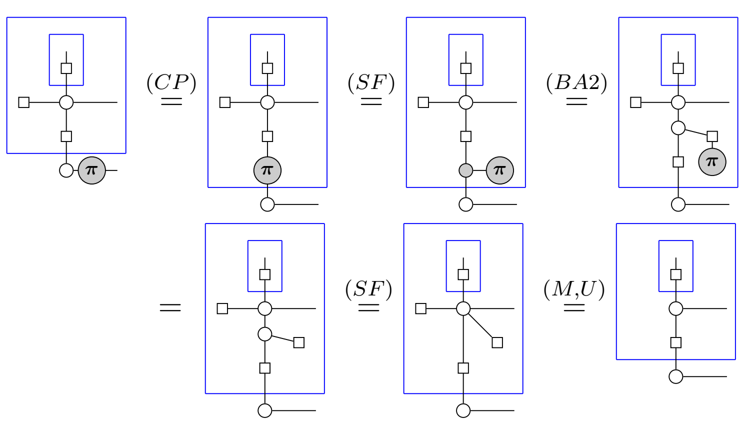
Definition 5. *Let (G, I, O) be an open graph. We say G has a (causal) flow if there exists a function $f : O^c \to I^c$ from non-output qubits to non-input qubits and a strict partial order $\prec$ over the vertices of the graph so that for each vertex v of G*

1. *v is connected to $f(v)$;*
2. *$v \prec f(v)$;*
3. *$v \prec u$ for each $u \neq v$ connected to $f(v)$.*

If a graph has a causal flow, it is uniformly, strongly and stepwise deterministic [7]. The way to correct for errors depends on the stabilizer presented in Lemma 1.

Lemma 1 (Graph stabilizer lemma). *For a graph $G = (V, E)$, the only state stabilized by the group generated by $\left\{X_v Z_{\mathrm{Odd}(\{v\})}\right\}_{v\in V}$ is the associated graph state.*

Proof. The graph state is indeed stabilized by this group



(16)

There are $|V|$ generators of the gruop, and they are independent Pauli gates as the X gates cannot be canceled out. So $\langle \left\{X_v Z_{\mathrm{Odd}(\{v\})}\right\}_{v\in V} \rangle$ stabilizes the graph-state [17]. $\square$

This lemma allows for corrections to be aplied retroactively to already measured qubits. After a qubit v is measured as the undesired result, applying an X gate to $f(v)$ and a Z gate to all neighbors of $f(v)$ except v is effectively the same as applying a Z gate to v itself. Thus, as long as $f(v)$ and the neighbors of $f(v)$ except for v have not been measured yet, it is possible to retroactively 'correct' the measurement. This also shows why $f(v)$ is not allowed to take on values in I, as the above stabiliser would not function when the Z-spider in question has an extra wire connecting to an input of the open graph.

If the measurements of the qubits are performed in an order that obeys the partial ordering $\prec$, it is always guaranteed that at the moment that an error occurs on a qubit v, $f(v)$ and its neighbors (except v itself) have not been measured yet, so that the error can always be corrected.

This construction immediately suggests another more general construction. If a Z gate was applied to all neighbors of v, that would correspond to applying an X gate to v, which could correct undesired results in the YZ basis, one might say $f(v) = v$ for vertices in the XY basis. To correct also for undesired results in the XZ basis it is necessary to allow the correction to involve additional qubits.

Definition 6. *The odd neighborhood* $\mathrm{Odd}(U)$ *of some subset* U *of the vertices in a graph* G *is defined to be those vertices which have an odd number of connections to vertices in* U*. Let* E *be the edges of* G *and* V *the vertices of* G*,then*

$$\mathrm{Odd}(U) := \Big\{ v \in V \mid \# \big\{ u \in U \mid u \neq v \wedge \{v, u\} \in E \big\} \textit{ is odd} \Big\}$$

The odd neigbborhood is important because when the stabilizer from Lemma 1 is applied to multiple qubits at once, with their X gates affecting qubits in some set U, the Z gates applied to neighbors of the qubits in U may cancel eachother so that the collective effect is a Z gate applied only to those qubits that are in $\mathrm{Odd}(U)$.

Definition 7. *Let* (G, I, O, λ) *be a labelled open graph with specified inputs, outputs, and* *measurement planes. We say G has a generalized flow if there exists a function $g : O^c \to \mathcal{P}(I^c)$ from non-output vertices to subsets of non-input vertices, and a strict partial order $\prec$ over the vertices of the graph so that for each vertex v of G*

1. *If $u \in g(v)$ and $u \neq v$ then $v \prec u$;*

2. *If $u \in \mathrm{Odd}(g(v))$ and $u \neq v$ then $v \prec u$;*

3. *If $\lambda(v) = XY$ then $v \in \mathrm{Odd}(g(v))$ and $v \notin g(v)$;*

4. *If $\lambda(v) = XZ$ then $v \notin \mathrm{Odd}(g(v))$ and $v \in g(v)$;*

5. *If $\lambda(v) = YZ$ then $v \in \mathrm{Odd}(g(v))$ and $v \in g(v)$.*

Definition 8. *A graph measurement pattern is said to have gflow if the associated labelled open graph has gflow.*

Theorem 1. *A graph measurement pattern is uniformly, strongly, and stepwise deterministic if and only if it has gflow.*

Proof. This is theorem 7.9.7 of [18]. □

3 Hypergraph Measurement patterns

Definition 9. *A hypergraph H is a pair of vertices and hyperedges (V, E), where the hyperedge set is a subset of the powerset of the vertices $E \subset \mathcal{P}(V)$, and each hyperedge contains at least two elements $|e| \geq 2 \forall e \in E$.*

The definition for a hypergraph measurement pattern is the same as that for a graph measurement pattern except the entangling operators E_{uv} are replaced by entangling operators E_S which act as multi-controlled Z gates on all qubits in the set $S \subset V$.

Definition 10 ([16])**.** *A hypergraph measurement pattern is a tuple consisting of a register of qubits V, subsets $I, O \subset V$ of inputs and outputs, and a set of operations*

- *Preparations N_v which initialize a qubit $v \in V$ to the state $|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$;*

- *Entangling operators E_S which act as a multi-controlled Z gate on the qubits $s \in S \subset V$;*

- *Destructive measurements $M_v^{\lambda,\alpha}$ which measure a qubit v in the λ plane at an angle α and project on either $|+\rangle_{\lambda,\alpha}$ or $|-\rangle_{\lambda,\alpha}$*

- *Corrections X_v and Z_S which act respectively as the pauli X gate on the qubit u and a multi-controlled Z gate on the set of qubits $S \subset U$, and can be applied to qubits depending on the measurement outcomes of qubits measured earlier.*

These new multi-controlled Z gates can be interpreted as hyper edges, making the new resource state a hypergraph-state. Branches of a hypergraph measurement pattern (H, I, O, λ) with measurement angles α are given by

$$\prod_{v \in O^c}^{\prec} \left(C_v^{s_v} M_v^{\lambda(v),\alpha(v),s_v} \alpha(v) \right) E_H N_{I^c},$$

where all symbols are the same as in 15, except E_H contains not only controlled Z gates for edges in H but also multi-controlled Z gates for hyperedges in H.

A hypergraph measurement is runnable and/or deterministic depended on the same coniditions as for a graph measurement pattern. These conditions were laid out in definitions 3 and 4

Definition 11. *The odd neighborhood* $\mathrm{Odd}(U)$ *of a subset U of the vertices V of a hypergraph H is defined to be the set of hyperedges that can be obtained in an odd number of ways by removing a single vertex from U from a hyperedge in H.*

$$\mathrm{Odd}(U) := \left\{ S \in \mathcal{P}(V) \mid \# \left\{ u \in U \mid \exists e \in E : u \in e \wedge e \setminus \{u\} = S \right\} \textit{ is odd} \right\},$$

where V is the vertex set of H and E is the hyperedge set of H.

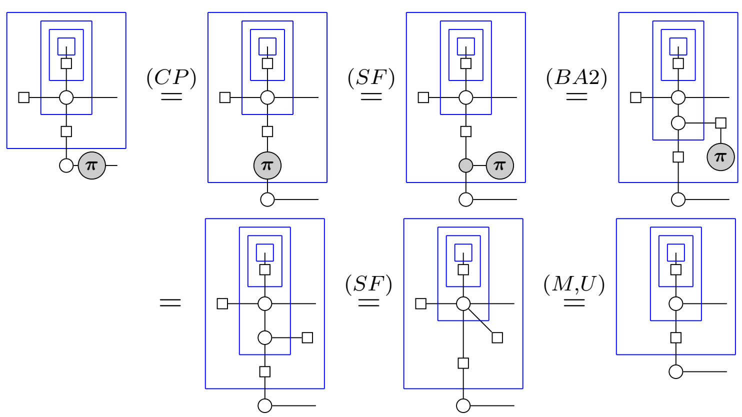
For the special case of graphs, this definition is the same as the Odd neighborhood on graphs, except that every neighbor is now a singleton containing that neighbor. This is because the condition $u \in e \wedge e \setminus \{u\} = S$ simplifies to $e = (u, s) \wedge S = \{s\}$.

3.1 Stabilisers of hypergraph states

The concept of generalized flow can be extended to hypergraph measurement patterns by first generalizing Lemma 1. The generalized version is presented in 2

Lemma 2 (Hypergraph stabilizer lemma)**.** *For a hypergraph $H = (V, E)$, the only state stabilized by the group G generated by $\left\{X_v Z_{\mathrm{Odd}(\{v\})}\right\}_{v \in V}$ is the associated hypergraph state.*

Proof. The hypergraph-state is indeed stabilized by this group

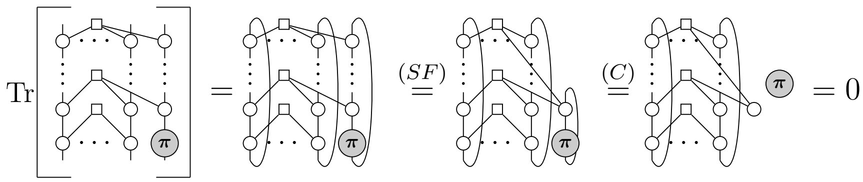


(17)

The projector P on the $+1$ eigenspace of all these stabilizers is given by [17]

$$\frac{1}{2}(\mathbb{1} + G_1) \cdots \frac{1}{2}(\mathbb{1} + G_{|V|}) = \frac{1}{2^{|V|}}(S_1 + \cdots + S_{2^{|V|}}),$$

where $G_1, \ldots, G_{|V|}$ are the generators of G and $S_1, \ldots, S_{2^{|V|}}$ are the elements of G. The equality between these two expressions holds for any Abelian group in which all generators square to identity. Consider an arbitrary element of G that is not the identity element. This element is made using generators and so there is at least one spider to which a X gate is applied. This element will disconnect using complementarity of X and Z spiders, resulting in a trace of zero.



The trace of the identity operator is the dimension of the system, in this case that is $2^{|V|}$.

$$\operatorname{Tr} P = \frac{1}{2^{|V|}} \operatorname{Tr} \mathbb{1} = 2^{|V|-|V|} = 1,$$

so up to a constant there is only a single element that is in the $+1$ eigenspaces of all operators in this group. $\square$

This Lemma is also hinted at in [12] although not explicitly proven. That same paper does show explicitly that the group from Lemma 2 is Abelian.

We say the hypergraph stabilizer is applied centered on the qubit v. When multiple stabilizers are applied centered on each qubit in a set U, the result is an X gate on each qubit in U and a controlled Z gate on each set of vertices in $\mathrm{Odd}(U)$. This hypergraph stabilizer lemma allows for generalized flow to be generalized to ZH-diagrams.

Although these stabilizers are not made of just single qubit gates, like the stabilizers of graph states were, they are still made of 'smaller' gates than are needed to generate the resource state itself. If the resource state contains hyperedges of order at most n, the stabilizers will contain at multi-controlled gates that involve at most $n-1$ qubits.

3.2 Generalized flow for hypergraph measurement patterns

The new stabilizers on hypergraph states allow the concept of generalized flow to be easily extended to hypergraph measurement patterns.

Definition 12. *Let (H, I, O, λ) be an open labelled hypergraph with specified inputs, outputs, and measurement planes. We say H has a generalized flow if there exists a function $g : O^c \to \mathcal{P}(V)$ from non-output vertices to subsets of non-input vertices, and a strict partial order $\prec$ over the vertices of the graph so that for each vertex v of G.*

1. *If $u \in g(v)$ and $u \neq v$ then $v \prec u$;*
2. *If $S \in \mathrm{Odd}(g(v))$ and $S \neq \{v\}$ then $\{v\} \prec S$;*
3. *If $\lambda(v) = XY$ then $\{v\} \in \mathrm{Odd}(g(v))$ and $v \notin g(v)$;*
4. *If $\lambda(v) = XZ$ then $\{v\} \notin \mathrm{Odd}(g(v))$ and $v \in g(v)$;*
5. *If $\lambda(v) = YZ$ then $\{v\} \in \mathrm{Odd}(g(v))$ and $v \in g(v)$.*

Where the partial order between sets is defined as $S_1 \prec S_2$ iff $s_1 \prec s_2$ for all $s_1 \in S_1$ and $s_2 \in S_2$.

When the hypergraph-state contains only arity-2 H-boxes, so that it is also a graph-state, this definition is the same as the definition of gflow for graph states except that all vertices in the definition have been replaced by singletons containing that single vertex.

Theorem 2. *A hypergraph measurement pattern is uniformly, strongly, and stepwise deterministic if and only if the associated labelled open hypergraph (H, I, O, λ) has gflow $(g, \prec)$.*

Proof. The proof is similar to the proof of 7.9.7 of [18].

($\Longleftarrow$) When measuring vertices in an order that respects the partial ordering $\prec$, whenever an error occurs at a qubit v it is possible to apply the hypergraph stabilizer centered on each qubit in $g(v)$, excluding those gates that would be a single qubit gate affecting v itself. Due to conditions 1 and 2 of gflow, these corrections will only affect qubits that have not been measured yet, and due to conditions 3, 4, and 5, their application will result in the error being corrected.

($\Longrightarrow$) This proof goes by induction. If $|O^c| = 0$, there is trivially a gflow.

Now for any $n \in \mathbb{N}$ assume that all uniformly, strongly, and stepwise deterministic measurement patterns with $|O^c| = n-1$ have gflow, and consider an arbitrary uniformly, strongly, and stepwise deterministic measurement pattern $\Gamma = (H, I, O, \lambda)$ for which $|O^c| = n$. Since the pattern is stepwise deterministic, it is still uniformly, strongly, and stepwise deterministic when one of the vertices is added to O. Call this vertex n and choose it so that C_n acts only on qubits in O. At least one such vertex exists, as there must be a last qubit to be measured. The new measurement pattern is $\Gamma' = (H, I, O', \lambda')$ where $O' = O \cup \{n\}$ and λ' is the restriction of λ to O'^c. Now, $|O'^c| = n-1$ so Γ' has gflow $(g', \prec')$ and Γ' realizes the unitary map

$$U' := \prod_{v \in O'^c} \langle + |_{v, \lambda(v), \alpha(v)} E_H N_{I^c}.$$

Then Γ realized the unitary map

$$\begin{cases} \langle + |_{n,\lambda,\alpha} U' & \text{if } s_n = 0 \\ \langle - |_{n,\lambda,\alpha} C_n U' & \text{if } s_n = 1 \end{cases}$$

and since Γ is strongly deterministic, these two are equal up to a phase $e^{i\phi}$ which may be taken to be zero by varying the correction C_n. Then by Lemma 7.9.8 of [18]

$$E_H N_{I^c} = C_n T E_H N_{I^c},$$

where T is chosen so that $\langle - |_{n,\lambda,\alpha} T = \langle + |_{n,\lambda,\alpha}$ so

$$T = \begin{cases} Z_n & \text{if } \lambda(n) = XY \\ Z_n X_n & \text{if } \lambda(n) = XZ \\ X_n & \text{if } \lambda(n) = YZ \end{cases}.$$

Thus $C_n T$ is a stabiliser of $E_H N_{I^c}$. From Lemma 2, it must then be the case that $C_n T =$

$\prod_{v \in S} X_v Z_{\mathrm{Odd}(\{v\})} = X_S Z_{\mathrm{Odd}(S)}$. As C_n only acts on qubits in O, $S \setminus \{v\}$ and $\mathrm{Odd}(S) \setminus \{\{v\}\}$ must contain only vertices that are in O, which is a successor to $\{v\}$. Thus setting

$$g(v) = \begin{cases} S & \text{if } v = n \\ g'(v) & \text{otherwise} \end{cases},$$

and $a \prec b \iff b = n \vee a \prec' b$, a gflow for H is obtained. □

3.3 Examples of generalized flow in hypergraphs

It is interesting to look at some examples of what generalized flow looks like on simple hypergraph-states. This section provides two minimal examples which are especially instructive as they both do not permit circuit extraction according to the algorithm from [7] although they have very different reasons for not permitting this.

Example 7. *Consider the hypergraph in Figure 5, with assigned measurement planes $\lambda(a) = \lambda(b) = XY$, $\lambda(d) = YZ$. The partial ordering $a \prec d \prec b \prec c, e$ along with the successor map $g(a) = \{b\}, g(b) = \{c, e\}, g(d) = \{d\}$ provide a generalized flow on this hypergraph-state. It satisfies the conditions from Definition 12 as*

$$\mathrm{Odd}(g(a) = \{\{a\}, \{c\}, \{c, d\}, \{d, e\}\}, \mathrm{Odd}(g(b)) = \{\{b\}\} \quad \textit{and} \quad \mathrm{Odd}(g(d)) = \{\{b, c\}, \{b, e\}\}.$$

One could attempt to extract a gate-based circuit from a hypergraph measurement pattern using the algorithm presented in Section 5.1 of [7], which works up to a point as gflow on hypergraphs and gflow on graphs are similar in form but for the present measurement pattern this attempt would get stuck at the point shown in Figure 6 when the algorithm from [7] requires a pivot to be performed about an edge connecting d to a frontier qubit. However in this hypergraph the only such edge is a hyperedge and the generalisation of pivots to hypergraphs still only allows pivoting about regular edges [13].

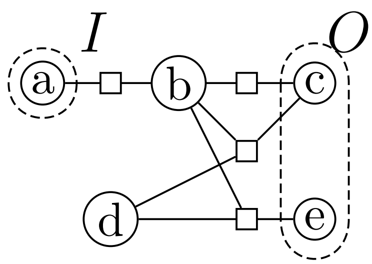


Figure 5: An example hypergraph. To turn this into a hypergraph state, all Z-spiders should be given one additional leg that acts as an output of the state. The spiders are labelled with letters and the input and output sets are enclosed in a dashed line and labelled with I and O respectively.

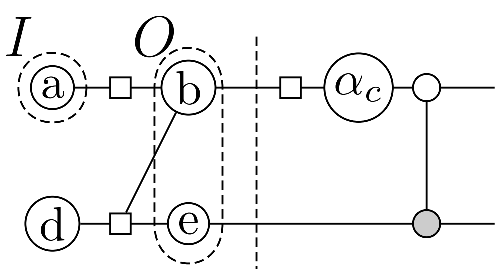


Figure 6: The measurement pattern from Figure 5 has been partially extracted and now contains one fewer qubit but is followed by post-processing which includes a CNOT gate, a Hadamard gate and a rotation by α_c which is the desired measurement angle of the original qubit c. A dashed line signifies the transition point from MBQC to gate-based circuit.

Example 8. *Even when all qubits are measured in the XY plane, it is not necessarily possible to straightforwardly apply the algorithm from [7]. For example, consider the hypergraph in Figure 7, with measurement plane $\lambda(a)$ =XY, and gflow $g(a) = \{b, c, d\}$, $a \prec b, c, d$. It satisfies the conditions from Definition 12 as*

$$\mathrm{Odd}(g(a)) = \big\{\{a\}\big\}$$

Attempting to extract a gate-based circuit from this measurement pattern fails, because one

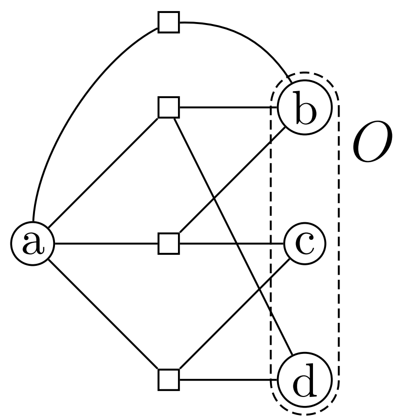
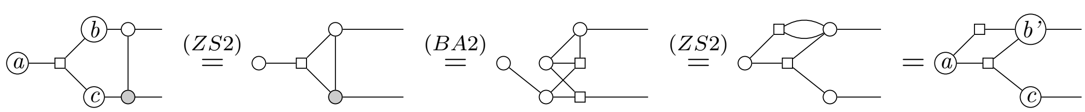


Figure 7: An example hypergraph. The qubits are labelled with lowercase letters and the outputs are enclosed in a dashed line labelled O.

cannot add the hyperedges of two qubits together when the target and source of the addition share a hyperedge.



Where the last step is Lemma 5.1 in [19]. The labels on the vertices serve only to identify these vertices.

Notice $\mathrm{Odd}(\{b'\}) = \big\{\{a\}, \{a, c\}\big\} \neq \big\{\{a, b\}, \{a, c\}\big\} = \mathrm{Odd}(\{b, c\})$, *a property that did hold for graph-states and the absence of this property means that here too the algorithm from [7] cannot be used to extract a gate-based circuit.*

4 Conclusion

Generalized flow (gflow) exists for measurement based quantum computing (MBQC) which use graph-states as their resource state. When applied to graph measurement, the existence of gflow is equivalent to that measurement pattern being uniformly, strongly, and stepwise deterministic. For graph measurement patterns, the existence of gflow also implies that it is possible to extract a gate-based circuit which achieves the same linear map as the measurement pattern did [7].

The concept of generalized flow can be extended to apply to a hypergraph measurement pattern, which uses a hypergraph-state as their resource state. The new definition of generalized flow looks similar to the generalized flow on graph measurement pattens. Both generalized flow on graph and on hypergraphs can be derived by starting with a group of stabilisers which uniquely fix the specific (hyper)graph-state. The stabilisers for a hypergraph being a generalized version of the stabilisers for a graph. The gflow condition on measurement patterns can be understood as a condition for when qubits can be measured in such an order that these stabilisers can be used to correct any possible error that might show up.

It is proven that a hypergraph measurement pattern is uniformly, strongly, and stepwise deterministic if and only if the hypergraph-state has gflow. However, no algorithm was found to create a gate-based quantum circuit that implements the same linear map as such a measurement pattern does.

Future work may investigate the possibilities to extract gate-based circuits from hypergraph measurement patterns, potentially using the algorithm from [7] as a base but needing to at least circumvent the inability to pivot about arbitrary hyperedges as well as the problem that shows up in Example 8.

References

[1] B. Coecke, *Kindergarten quantum mechanics*, 2005. DOI: 10.48550/ARXIV.QUANT-PH/0510032. [Online]. Available: https://arxiv.org/abs/quant-ph/0510032.

[2] A. Joyal and R. Street, "The geometry of tensor calculus, i," *Advances in Mathematics*, vol. 88, no. 1, pp. 55–112, Jul. 1991, ISSN: 0001-8708. DOI: 10.1016/0001-8708(91)90003-p. [Online]. Available: http://dx.doi.org/10.1016/0001-8708(91)90003-P.

[3] B. Coecke and A. Kissinger, *Picturing quantum processes*. Cambridge, England: Cambridge University Press, Mar. 2017.

[4] E. Jeandel, S. Perdrix, and R. Vilmart, "Completeness of the zx-calculus," 2019. DOI: 10.23638/LMCS-16(2:11)2020. eprint: arXiv:1903.06035.

[5] S. Dündar-Coecke, L. Yeh, C. Puca, *et al.*, "Quantum picturalism: Learning quantum theory in high school," 2023. DOI: 10.1109/QCE57702.2023.20321. eprint: arXiv:2312.03653.

[6] C. Holker, *Causal flow preserving optimisation of quantum circuits in the zx-calculus*, 2023. DOI: 10.48550/ARXIV.2312.02793. [Online]. Available: https://arxiv.org/abs/2312.02793.

[7] M. Backens, H. Miller-Bakewell, G. de Felice, L. Lobski, and J. van de Wetering, "There and back again: A circuit extraction tale," 2020. DOI: 10.22331/q-2021-03-25-421. eprint: arXiv:2003.01664.

[8] B. Coecke, *Basic zx-calculus for students and professionals*, 2023. eprint: arXiv:2303.03163.

[9] V. Danos and E. Kashefi, "Determinism in the one-way model," *Physical Review A*, vol. 74, no. 5, Nov. 2006, ISSN: 1094-1622. DOI: 10.1103/physreva.74.052310. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.74.052310.

[10] D. E. Browne, E. Kashefi, M. Mhalla, and S. Perdrix, "Generalized flow and determinism in measurement-based quantum computation," *New Journal of Physics*, vol. 9, no. 8, pp. 250–250, Aug. 2007, ISSN: 1367-2630. DOI: 10.1088/1367-2630/9/8/250. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/9/8/250.

[11] M. Backens and A. Kissinger, "Zh: A complete graphical calculus for quantum computations involving classical non-linearity," 2018. DOI: 10.4204/EPTCS.287.2. eprint: arXiv:1805.02175.

[12] M. Rossi, M. Huber, D. Bruß, and C. Macchiavello, "Quantum hypergraph states," 2012. DOI: 10.1088/1367-2630/15/11/113022. eprint: arXiv:1211.5554.

[13] L. Lemonnier, J. van de Wetering, and A. Kissinger, "Hypergraph simplification: Linking the path-sum approach to the zh-calculus," 2020. DOI: 10.4204/EPTCS.340.10. eprint: arXiv:2003.13564.

[14] J. van de Wetering, *Zx-calculus for the working quantum computer scientist*, 2020. DOI: 10.48550/ARXIV.2012.13966. [Online]. Available: https://arxiv.org/abs/2012.13966.

[15] R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," *Annals of Physics*, vol. 349, pp. 117–158, Oct. 2014, ISSN: 0003-4916. DOI: 10.1016/j.aop.2014.06.013. [Online]. Available: http://dx.doi.org/10.1016/j.aop.2014.06.013.

[16] V. Danos, E. Kashefi, and P. Panangaden, "The measurement calculus," *Journal of the ACM*, vol. 54, no. 2, p. 8, Apr. 2007, ISSN: 1557-735X. DOI: 10.1145/1219092.1219096. [Online]. Available: http://dx.doi.org/10.1145/1219092.1219096.

[17] A. Ekert, T. Hosgood, A. Kay, and C. Macchiavello. "Introduction to Quantum Information Science." (May 7, 2024), [Online]. Available: https://qubit.guide.

[18] V. Danos, E. Kashefi, P. Panangaden, and S. Perdrix, "Extended measurement calculus," in *Semantic Techniques in Quantum Computation*. Cambridge University Press, Nov. 2009, pp. 235–310. DOI: 10.1017/cbo9781139193313.008. [Online]. Available: http://dx.doi.org/10.1017/CBO9781139193313.008.

[19] M. Backens, A. Kissinger, H. Miller-Bakewell, J. van de Wetering, and S. Wolffs, "Completeness of the zh-calculus," 2021. DOI: 10.32408/compositionality-5-5. eprint: arXiv:2103.06610.