The circuit model, the one-way model, and the ZX-calculus

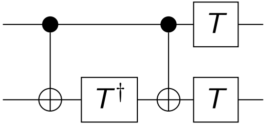
Miriam Backens (they/them)
MOCQUA, Inria & Loria, Nancy
miriam.backens@inria.fr

Graphix workshop, 18 November 2024

Two models of quantum computation

Two models of quantum computation

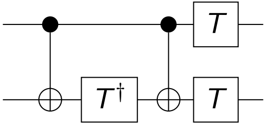
quantum circuit model



[Deutsch 1989]

Two models of quantum computation

quantum circuit model

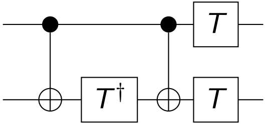


[Deutsch 1989]

- initialise state $|0\dots0\rangle$

Two models of quantum computation

quantum circuit model

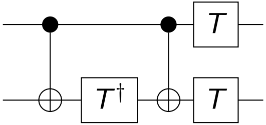


[Deutsch 1989]

- initialise state $|0\ldots0\rangle$
- computation driven by unitary gates

Two models of quantum computation

quantum circuit model

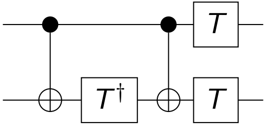


[Deutsch 1989]

- initialise state $|0\ldots0\rangle$
- computation driven by unitary gates
- read out data using computational basis measurements

Two models of quantum computation

quantum circuit model



[Deutsch 1989]

- initialise state $|0\dots0\rangle$
- computation driven by unitary gates
- read out data using computational basis measurements

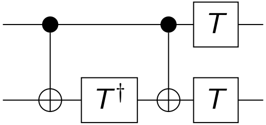
one-way model (MBQC)

$$X_3^{s_2} M_2^{XY,\beta} Z_3^{s_1} X_2^{s_1} M_1^{XY,\alpha} E_{23} E_{12} N_3 N_2$$

[Raussendorf & Briegel 2001]

Two models of quantum computation

quantum circuit model



[Deutsch 1989]

- initialise state $|0\dots0\rangle$
- computation driven by unitary gates
- read out data using computational basis measurements

one-way model (MBQC)

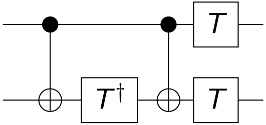
$$X_3^{s_2} M_2^{XY,\beta} Z_3^{s_1} X_2^{s_1} M_1^{XY,\alpha} E_{23} E_{12} N_3 N_2$$

[Raussendorf & Briegel 2001]

- initialise entangled resource state (can be independent of computation)

Two models of quantum computation

quantum circuit model



[Deutsch 1989]

- initialise state $|0 \ldots 0\rangle$
- computation driven by unitary gates
- read out data using computational basis measurements

one-way model (MBQC)

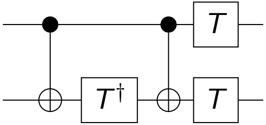
$$X_3^{s_2} M_2^{XY,\beta} Z_3^{s_1} X_2^{s_1} M_1^{XY,\alpha} E_{23} E_{12} N_3 N_2$$

[Raussendorf & Briegel 2001]

- initialise entangled resource state (can be independent of computation)
- computation driven by successive adaptive single-qubit measurements

Two models of quantum computation

quantum circuit model



[Deutsch 1989]

- initialise state $|0\ldots0\rangle$
- computation driven by unitary gates
- read out data using computational basis measurements

one-way model (MBQC)

$$X_3^{s_2} M_2^{XY,\beta} Z_3^{s_1} X_2^{s_1} M_1^{XY,\alpha} E_{23} E_{12} N_3 N_2$$

[Raussendorf & Briegel 2001]

- initialise entangled resource state (can be independent of computation)
- computation driven by successive adaptive single-qubit measurements
- if goal is state preparation, need Pauli gates as correction at end

Motivation

Translation between models

- circuit to MBQC is straightforward

Motivation

Translation between models

- circuit to MBQC is straightforward
- MBQC to circuit with ancillas is also straightforward

Motivation

Translation between models

- circuit to MBQC is straightforward
- MBQC to circuit with ancillas is also straightforward

Motivation

Translation between models

- circuit to MBQC is straightforward
- MBQC to circuit with ancillas is also straightforward

Question 1: How to translate any (suitably deterministic) MBQC into an ancilla-free circuit?

Motivation

Translation between models

- circuit to MBQC is straightforward
- MBQC to circuit with ancillas is also straightforward

Question 1: How to translate any (suitably deterministic) MBQC into an ancilla-free circuit?

Rewriting MBQCs

- optimisation (of circuits and of MBQC themselves), e.g. computational depth, number of operations,...

Motivation

Translation between models

- circuit to MBQC is straightforward
- MBQC to circuit with ancillas is also straightforward

Question 1: How to translate any (suitably deterministic) MBQC into an ancilla-free circuit?

Rewriting MBQCs

- optimisation (of circuits and of MBQC themselves), e.g. computational depth, number of operations,...
- verification

Motivation

Translation between models

- circuit to MBQC is straightforward
- MBQC to circuit with ancillas is also straightforward

Question 1: How to translate any (suitably deterministic) MBQC into an ancilla-free circuit?

Rewriting MBQCs

- optimisation (of circuits and of MBQC themselves), e.g. computational depth, number of operations,...
- verification
- blind quantum computation

Motivation

Translation between models

- circuit to MBQC is straightforward
- MBQC to circuit with ancillas is also straightforward

Question 1: How to translate any (suitably deterministic) MBQC into an ancilla-free circuit?

Rewriting MBQCs

- optimisation (of circuits and of MBQC themselves), e.g. computational depth, number of operations,...
- verification
- blind quantum computation

Motivation

Translation between models

- circuit to MBQC is straightforward
- MBQC to circuit with ancillas is also straightforward

Question 1: How to translate any (suitably deterministic) MBQC into an ancilla-free circuit?

Rewriting MBQCs

- optimisation (of circuits and of MBQC themselves), e.g. computational depth, number of operations,...
- verification
- blind quantum computation

Question 2: What rewrite rules would be useful?

Outline

The one-way model of measurement-based quantum computing

A common formalism for circuits and MBQC

Translation and rewriting

Conclusions

Outline

The one-way model of measurement-based quantum computing

A common formalism for circuits and MBQC

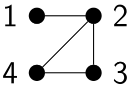
Translation and rewriting

Conclusions

Resource states for the one-way model

The resource states are **graph states** defined by simple graphs $G = (V, E)$:

- For each vertex in V, a qubit prepared in state $|+\rangle$.

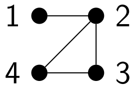


$\rightsquigarrow$ $|++++\rangle$

Resource states for the one-way model

The resource states are **graph states** defined by simple graphs $G = (V, E)$:

- For each vertex in V, a qubit prepared in state $|+\rangle$.
- For each edge in E, a CZ gate.

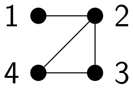


$$\rightsquigarrow \qquad CZ_{12}CZ_{23}CZ_{24}CZ_{34}\,|{+}{+}{+}{+}\rangle$$

Resource states for the one-way model

The resource states are **graph states** defined by simple graphs $G = (V, E)$:

- For each vertex in V, a qubit prepared in state $|+\rangle$.
- For each edge in E, a CZ gate.

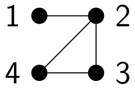


$$\rightsquigarrow \qquad CZ_{12}\, CZ_{23}\, CZ_{24}\, CZ_{34}\, |{+}{+}{+}{+}\rangle$$

Resource states for the one-way model

The resource states are **graph states** defined by simple graphs $G = (V, E)$:

- For each vertex in V, a qubit prepared in state $|+\rangle$.
- For each edge in E, a CZ gate.



$$\rightsquigarrow \qquad CZ_{12}\,CZ_{23}\,CZ_{24}\,CZ_{34}\,|{+}{+}{+}{+}\rangle$$

All graph states are **stabiliser states**: eigenstates of certain tensor products of Pauli matrices.

Measurements and corrections

Measurements are constrained to three planes of the Bloch sphere spanned by two of the Pauli operators: XY, XZ and YZ.

- Each measurement has an associated angle α.

Measurements and corrections

Measurements are constrained to three planes of the Bloch sphere spanned by two of the Pauli operators: XY, XZ and YZ.

- Each measurement has an associated angle α.
- Pauli measurements may be treated separately.

Measurements and corrections

Measurements are constrained to three planes of the Bloch sphere spanned by two of the Pauli operators: XY, XZ and YZ.

- Each measurement has an associated angle α.
- Pauli measurements may be treated separately.

Measurements and corrections

Measurements are constrained to three planes of the Bloch sphere spanned by two of the Pauli operators: XY, XZ and YZ.

- Each measurement has an associated angle α.
- Pauli measurements may be treated separately.

Desired and **undesired** measurement outcomes are related by Pauli matrices, e.g.

$$|+_{XY,\alpha}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\alpha}|1\rangle) \qquad |-_{XY,\alpha}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - e^{i\alpha}|1\rangle) = Z|+_{XY,\alpha}\rangle$$

Measurements and corrections

Measurements are constrained to three planes of the Bloch sphere spanned by two of the Pauli operators: XY, XZ and YZ.

- Each measurement has an associated angle α.
- Pauli measurements may be treated separately.

Desired and **undesired** measurement outcomes are related by Pauli matrices, e.g.

$$|+_{XY,\alpha}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\alpha}|1\rangle) \qquad |-_{XY,\alpha}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - e^{i\alpha}|1\rangle) = Z|+_{XY,\alpha}\rangle$$

Correction strategy: use **stabiliser property** of graph state to turn undesired outcome into desired one by applying Paulis to other qubits.

- Must have trivial effects on all qubits that are already measured.

Determinism and flow properties

Theorem [Browne et al. 2007, Mhalla et al. 2022]

An MBQC has a robustly deterministic implementation if and only if the underlying labelled open graph has flow.

Determinism and flow properties

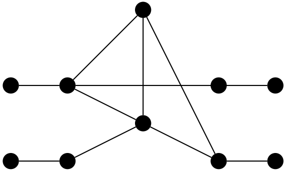
Theorem [Browne et al. 2007, Mhalla et al. 2022]

An MBQC has a robustly deterministic implementation if and only if the underlying labelled open graph has flow.

Determinism and flow properties

Theorem [Browne et al. 2007, Mhalla et al. 2022]

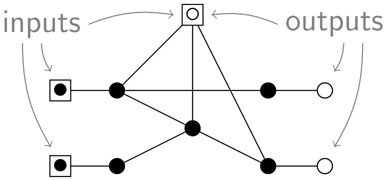
An MBQC has a robustly deterministic implementation if and only if the underlying labelled open graph has flow.



Determinism and flow properties

Theorem [Browne et al. 2007, Mhalla et al. 2022]

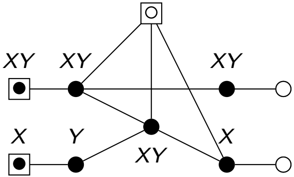
An MBQC has a robustly deterministic implementation if and only if the underlying labelled open graph has flow.



Determinism and flow properties

Theorem [Browne et al. 2007, Mhalla et al. 2022]

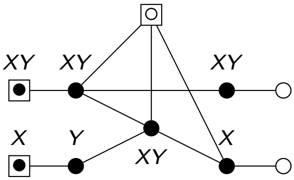
An MBQC has a robustly deterministic implementation if and only if the underlying labelled open graph has flow.



Determinism and flow properties

Theorem [Browne et al. 2007, Mhalla et al. 2022]

An MBQC has a robustly deterministic implementation if and only if the underlying labelled open graph has flow.



A **flow** consists of

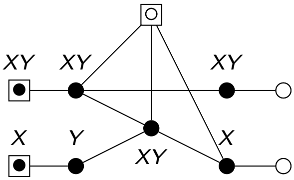
- a partial order over the vertices,
- and a correction function,

satisfying certain compatibility conditions.

Determinism and flow properties

Theorem [Browne et al. 2007, Mhalla et al. 2022]

An MBQC has a robustly deterministic implementation if and only if the underlying labelled open graph has flow.



A **flow** consists of

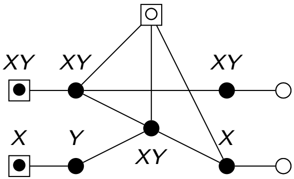
- a partial order over the vertices,
- and a correction function,

satisfying certain compatibility conditions.

Determinism and flow properties

Theorem [Browne et al. 2007, Mhalla et al. 2022]

An MBQC has a robustly deterministic implementation if and only if the underlying labelled open graph has flow.



A **flow** consists of

- a partial order over the vertices,
- and a correction function,

satisfying certain compatibility conditions.

Theorem [de Beaudrap 2008; Mhalla & Perdrix 2008; B. et al. 2021; Simmons 2021]

Flows can be found in polynomial time.

Outline

The one-way model of measurement-based quantum computing

A common formalism for circuits and MBQC

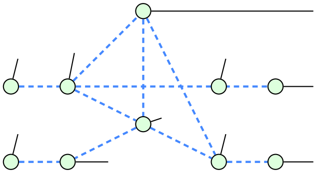
Translation and rewriting

Conclusions

The one-way model in ZX-notation

An MBQC is represented in the graphical ZX-notation as follows:

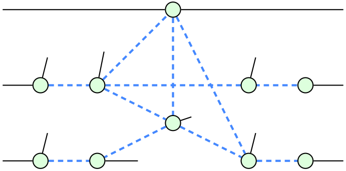
- green vertices with one outgoing wire each, connected by blue dashed edges, for the graph state,



The one-way model in ZX-notation

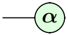
An MBQC is represented in the graphical ZX-notation as follows:

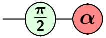
- green vertices with one outgoing wire each, connected by blue dashed edges, for the graph state,
- additional wires for the inputs, and

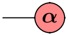


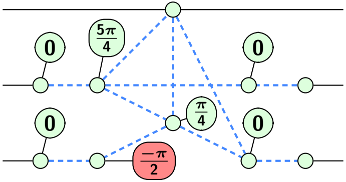
The one-way model in ZX-notation

An MBQC is represented in the graphical ZX-notation as follows:

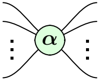
- green vertices with one outgoing wire each, connected by blue dashed edges, for the graph state,
- additional wires for the inputs, and
- additional vertices connected to some of the outgoing wires for the measurements: $XY \rightsquigarrow$ 

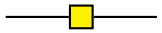
$XZ \rightsquigarrow$ 

$YZ \rightsquigarrow$ 



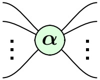
ZX generators and circuits in ZX-notation

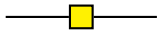


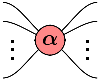


$$\begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{i\alpha} \end{pmatrix} \qquad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

ZX generators and circuits in ZX-notation

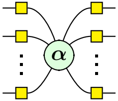






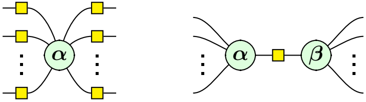
$$\begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{i\alpha} \end{pmatrix}$$

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

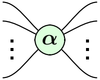


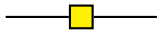
ZX generators and circuits in ZX-notation

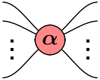
$$\begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{i\alpha} \end{pmatrix} \qquad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

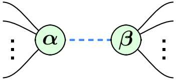


ZX generators and circuits in ZX-notation



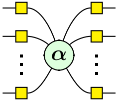


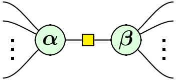


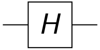


$$\begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{i\alpha} \end{pmatrix}$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



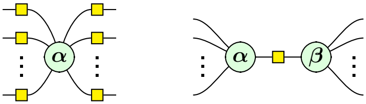


Gate 

ZX equivalent 

ZX generators and circuits in ZX-notation

$$\begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{i\alpha} \end{pmatrix} \qquad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



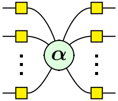
Gate

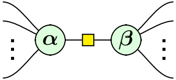
ZX equivalent

ZX generators and circuits in ZX-notation

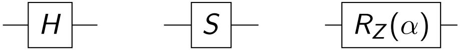
$$\begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{i\alpha} \end{pmatrix}$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

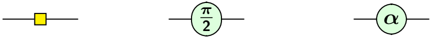




Gate

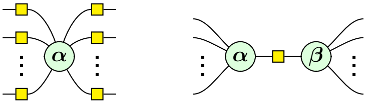


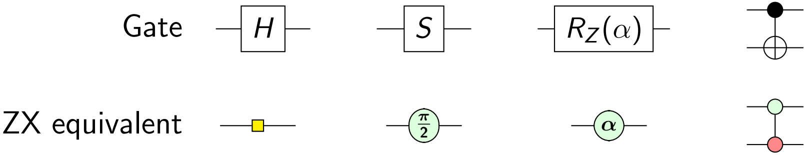
ZX equivalent



ZX generators and circuits in ZX-notation

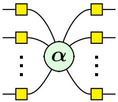
$$\begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{i\alpha} \end{pmatrix} \qquad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

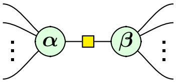


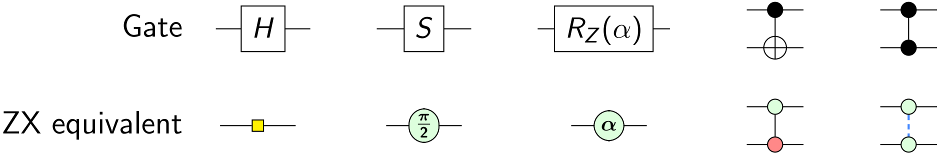


ZX generators and circuits in ZX-notation

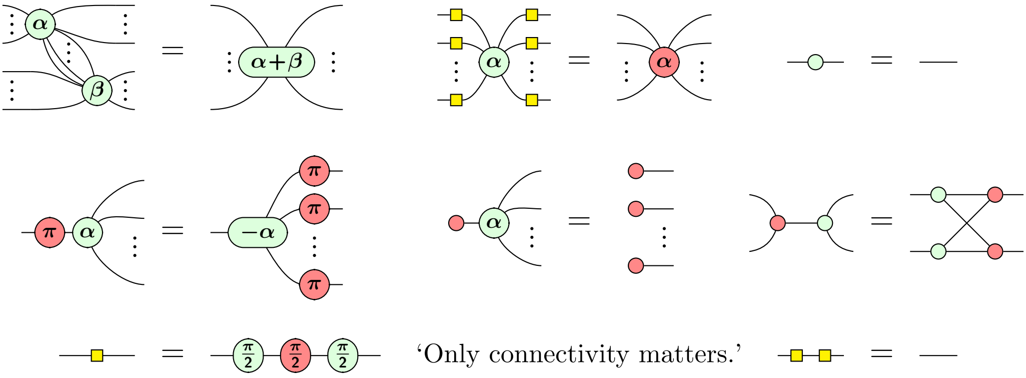
$$\begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{i\alpha} \end{pmatrix} \qquad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$







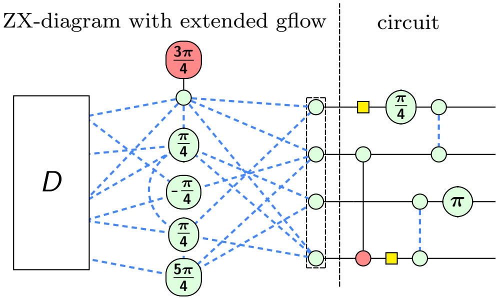
Sound and complete ZX-calculus rewrite rules (up to scalars)



This set is complete for the stabiliser ZX-calculus [B. 2014], can find overview over different complete rule sets in [van de Wetering 2021].

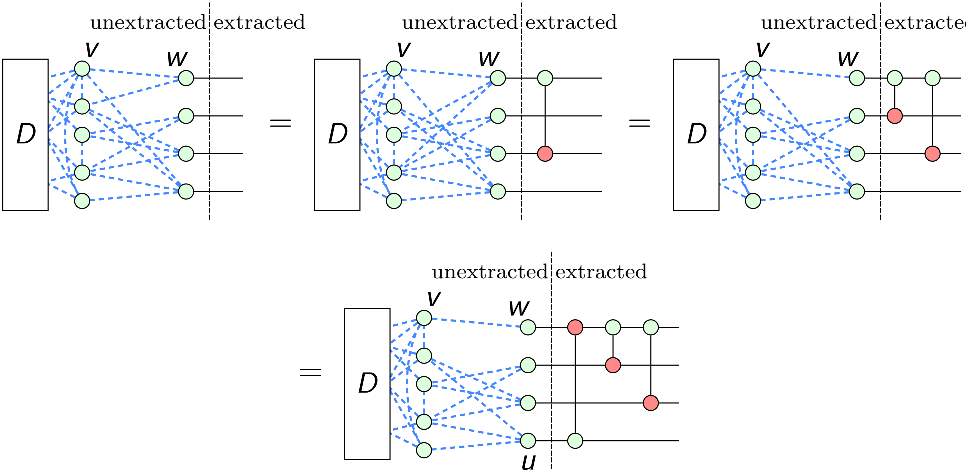
Outline

Ancilla-free circuit extraction: overview

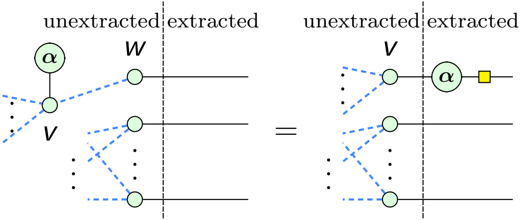


[Duncan et al. 2020; B., Miller-Bakewell, Felice, Lobski, van de Wetering 2021; Staudacher 2023]

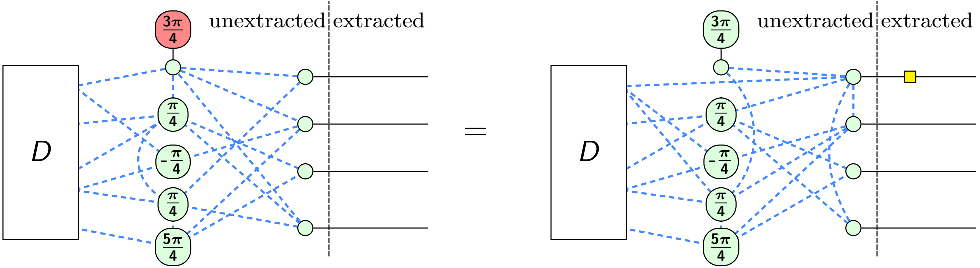
Simplify connections between frontier and unextracted layer



Extract a maximal vertex



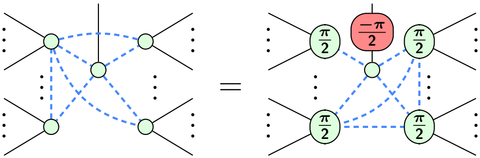
If needed, change measurement type



Flow-preserving rewrite rules for the stabiliser ZX-calculus

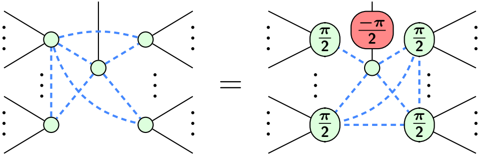
Flow-preserving rewrite rules for the stabiliser ZX-calculus

Local complementation



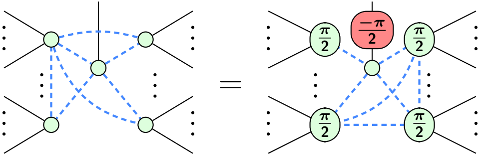
Flow-preserving rewrite rules for the stabiliser ZX-calculus

Local complementation

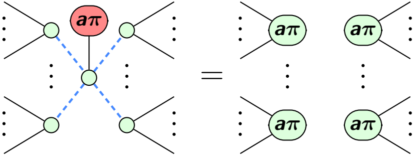


Flow-preserving rewrite rules for the stabiliser ZX-calculus

Local complementation

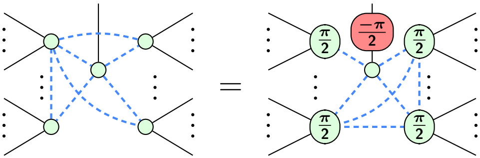


Z-deletion/insertion

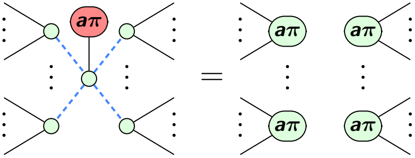


Flow-preserving rewrite rules for the stabiliser ZX-calculus

Local complementation

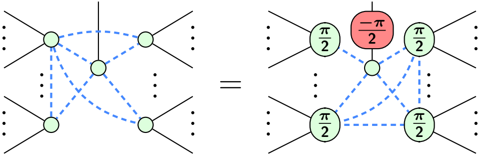


Z-deletion/insertion

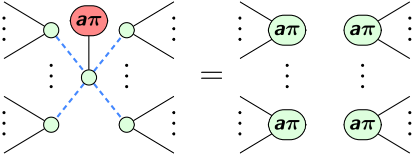


Flow-preserving rewrite rules for the stabiliser ZX-calculus

Local complementation



Z-deletion/insertion



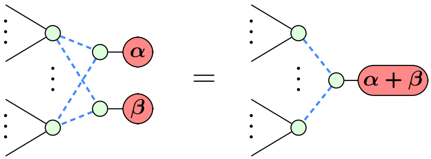
Theorem [McElvanney & B. 2023]

Suppose D and D' are two stabiliser ZX-diagrams with flow that both represent the same linear map. Then one can be rewritten into the other using local complementation, Z-insertion, and Z-deletion.

Further flow-preserving rewrite rules

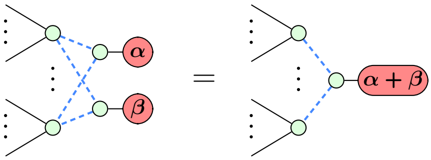
Further flow-preserving rewrite rules

Phase gadget fusion/splitting



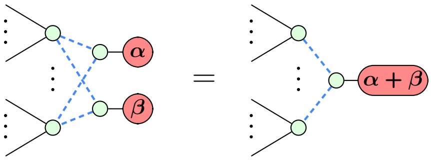
Further flow-preserving rewrite rules

Phase gadget fusion/splitting

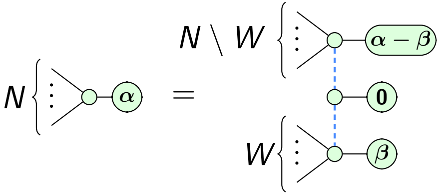


Further flow-preserving rewrite rules

Phase gadget fusion/splitting



Vertex splitting/fusion



Applications of flow-preserving rewriting

Optimisation

- T-count [Duncan et al. 2020]

Applications of flow-preserving rewriting

Optimisation

- T-count [Duncan et al. 2020]
- two-qubit gate count [Staudacher et al. 2022]

Applications of flow-preserving rewriting

Optimisation

- T-count [Duncan et al. 2020]
- two-qubit gate count [Staudacher et al. 2022]

Applications of flow-preserving rewriting

Optimisation

- T-count [Duncan et al. 2020]
- two-qubit gate count [Staudacher et al. 2022]

Blind quantum computing

- more resource-efficient protocol for scenario with two non-communicating quantum servers [Cao 2023]

Applications of flow-preserving rewriting

Optimisation

- T-count [Duncan et al. 2020]
- two-qubit gate count [Staudacher et al. 2022]

Blind quantum computing

- more resource-efficient protocol for scenario with two non-communicating quantum servers [Cao 2023]

Applications of flow-preserving rewriting

Optimisation

- T-count [Duncan et al. 2020]
- two-qubit gate count [Staudacher et al. 2022]

Blind quantum computing

- more resource-efficient protocol for scenario with two non-communicating quantum servers [Cao 2023]

Computational complexity of reasoning with ZX-calculus

- circuit extraction from arbitrary ZX diagram is #P-hard [de Beaudrap et al. 2022]

Applications of flow-preserving rewriting

Optimisation

- T-count [Duncan et al. 2020]
- two-qubit gate count [Staudacher et al. 2022]

Blind quantum computing

- more resource-efficient protocol for scenario with two non-communicating quantum servers [Cao 2023]

Computational complexity of reasoning with ZX-calculus

- circuit extraction from arbitrary ZX diagram is #P-hard [de Beaudrap et al. 2022]
- circuit extraction from ZX diagrams with flow is in P

Outline

The one-way model of measurement-based quantum computing

A common formalism for circuits and MBQC

Translation and rewriting

Conclusions

Summary and outlook

It is useful to:

- translate between one-way measurement patterns and circuits

Summary and outlook

It is useful to:

- translate between one-way measurement patterns and circuits
- rewrite measurement patterns (while preserving interpretation & property of having flow)

Summary and outlook

It is useful to:

- translate between one-way measurement patterns and circuits
- rewrite measurement patterns (while preserving interpretation & property of having flow)
- employ ZX-calculus as a common language and to visualise graph structure

Summary and outlook

It is useful to:

- translate between one-way measurement patterns and circuits
- rewrite measurement patterns (while preserving interpretation & property of having flow)
- employ ZX-calculus as a common language and to visualise graph structure

Summary and outlook

It is useful to:

- translate between one-way measurement patterns and circuits
- rewrite measurement patterns (while preserving interpretation & property of having flow)
- employ ZX-calculus as a common language and to visualise graph structure

Further work

- complete set of flow-preserving rewrite rules

Summary and outlook

It is useful to:

- translate between one-way measurement patterns and circuits
- rewrite measurement patterns (while preserving interpretation & property of having flow)
- employ ZX-calculus as a common language and to visualise graph structure

Further work

- complete set of flow-preserving rewrite rules
- extra conditions for some rules for different types of flow

Summary and outlook

It is useful to:

- translate between one-way measurement patterns and circuits
- rewrite measurement patterns (while preserving interpretation & property of having flow)
- employ ZX-calculus as a common language and to visualise graph structure

Further work

- complete set of flow-preserving rewrite rules
- extra conditions for some rules for different types of flow
- going beyond the current framework of flow

Summary and outlook

It is useful to:

- translate between one-way measurement patterns and circuits
- rewrite measurement patterns (while preserving interpretation & property of having flow)
- employ ZX-calculus as a common language and to visualise graph structure

Further work

- complete set of flow-preserving rewrite rules
- extra conditions for some rules for different types of flow
- going beyond the current framework of flow

Summary and outlook

It is useful to:

- translate between one-way measurement patterns and circuits
- rewrite measurement patterns (while preserving interpretation & property of having flow)
- employ ZX-calculus as a common language and to visualise graph structure

Further work

- complete set of flow-preserving rewrite rules
- extra conditions for some rules for different types of flow
- going beyond the current framework of flow

Thank you!