Flow-preserving rewriting in the ZX-calculus

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

SYCO 12, University of Birmingham, 15 April 2024

Outline

Quantum computing

The ZX-calculus

The one-way model of measurement-based quantum computing

Flow-preserving rewriting

Conclusions

Quantum computing in a nutshell

Describing the (pure state) quantum mechanics of qubits:

states: vectors (in a complex Hilbert space of dimension 2^n)

Quantum computing in a nutshell

Describing the (pure state) quantum mechanics of qubits:

states: vectors (in a complex Hilbert space of dimension 2^n)

transformations: linear maps

Quantum computing in a nutshell

Describing the (pure state) quantum mechanics of qubits:

states: vectors (in a complex Hilbert space of dimension 2^n)

transformations: linear maps

measurements: projections (at least for our purposes), probabilistic

Quantum computing in a nutshell

Describing the (pure state) quantum mechanics of qubits:

states: vectors (in a complex Hilbert space of dimension 2^n)

transformations: linear maps

measurements: projections (at least for our purposes), probabilistic

Quantum computing in a nutshell

Describing the (pure state) quantum mechanics of qubits:

states: vectors (in a complex Hilbert space of dimension 2^n)

transformations: linear maps

measurements: projections (at least for our purposes), probabilistic

Fix a basis, then this gives a dagger compact PROP where morphisms $n \to m$ are complex matrices of size $2^n \times 2^m$.

Quantum computing in a nutshell

Describing the (pure state) quantum mechanics of qubits:

- states: vectors (in a complex Hilbert space of dimension 2^n)
- transformations: linear maps
- measurements: projections (at least for our purposes), probabilistic

Fix a basis, then this gives a dagger compact PROP where morphisms $n \to m$ are complex matrices of size $2^n \times 2^m$.

These arbitrary exponentially large matrices

- are difficult to work with on the theory side

Quantum computing in a nutshell

Describing the (pure state) quantum mechanics of qubits:

- states: vectors (in a complex Hilbert space of dimension 2^n)
- transformations: linear maps
- measurements: projections (at least for our purposes), probabilistic

Fix a basis, then this gives a dagger compact PROP where morphisms $n \to m$ are complex matrices of size $2^n \times 2^m$.

These arbitrary exponentially large matrices

- are difficult to work with on the theory side
- and cannot be implemented directly on the experimental side

Quantum computing in a nutshell

Describing the (pure state) quantum mechanics of qubits:

states:	vectors (in a complex Hilbert space of dimension 2^n)
transformations:	linear maps
measurements:	projections (at least for our purposes), probabilistic

Fix a basis, then this gives a dagger compact PROP where morphisms $n \to m$ are complex matrices of size $2^n \times 2^m$.

These arbitrary exponentially large matrices

- are difficult to work with on the theory side
- and cannot be implemented directly on the experimental side

Quantum computing in a nutshell

Describing the (pure state) quantum mechanics of qubits:

states:	vectors (in a complex Hilbert space of dimension 2^n)
transformations:	linear maps
measurements:	projections (at least for our purposes), probabilistic

Fix a basis, then this gives a dagger compact PROP where morphisms $n \to m$ are complex matrices of size $2^n \times 2^m$.

These arbitrary exponentially large matrices

- are difficult to work with on the theory side
- and cannot be implemented directly on the experimental side

so represent everything in terms of a chosen set of generators.

The quantum circuit model

- State preparation and measurements only in the standard basis.

The quantum circuit model

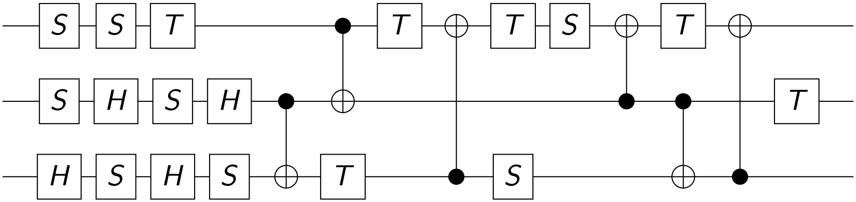
- State preparation and measurements only in the standard basis.
- Computation is driven by unitary transformations, generated by a set of gates (which usually act on 1 or 2 qubits).

The quantum circuit model

- State preparation and measurements only in the standard basis.
- Computation is driven by unitary transformations, generated by a set of gates (which usually act on 1 or 2 qubits).

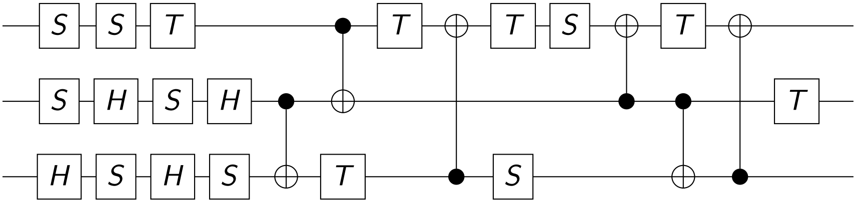
The quantum circuit model

- State preparation and measurements only in the standard basis.
- Computation is driven by unitary transformations, generated by a set of gates (which usually act on 1 or 2 qubits).



The quantum circuit model

- State preparation and measurements only in the standard basis.
- Computation is driven by unitary transformations, generated by a set of gates (which usually act on 1 or 2 qubits).



Quantum circuits act as a sort of 'machine language' for quantum computers.

Questions arising from limited quantum computing resources

Currently, and near-term future: noisy intermediate-scale quantum (NISQ) era

- Processors have up to a few hundred qubits

Questions arising from limited quantum computing resources

Currently, and near-term future: noisy intermediate-scale quantum (NISQ) era

- Processors have up to a few hundred qubits
- Cannot implement error-correction yet, so have to tolerate noise

Questions arising from limited quantum computing resources

Currently, and near-term future: noisy intermediate-scale quantum (NISQ) era

- Processors have up to a few hundred qubits
- Cannot implement error-correction yet, so have to tolerate noise
- Generally limited connectivity for two-/multi-qubit gates

Questions arising from limited quantum computing resources

Currently, and near-term future: noisy intermediate-scale quantum (NISQ) era

- Processors have up to a few hundred qubits
- Cannot implement error-correction yet, so have to tolerate noise
- Generally limited connectivity for two-/multi-qubit gates

Questions arising from limited quantum computing resources

Currently, and near-term future: noisy intermediate-scale quantum (NISQ) era

- Processors have up to a few hundred qubits
- Cannot implement error-correction yet, so have to tolerate noise
- Generally limited connectivity for two-/multi-qubit gates

Longer term: large-scale computation with error correction

- Error-correction schemes generally do not allow any universal gate set to be directly implemented in a fault-tolerant way

Questions arising from limited quantum computing resources

Currently, and near-term future: noisy intermediate-scale quantum (NISQ) era

- Processors have up to a few hundred qubits
- Cannot implement error-correction yet, so have to tolerate noise
- Generally limited connectivity for two-/multi-qubit gates

Longer term: large-scale computation with error correction

- Error-correction schemes generally do not allow any universal gate set to be directly implemented in a fault-tolerant way
- Secure delegation to a quantum server in the cloud

Questions arising from limited quantum computing resources

Currently, and near-term future: noisy intermediate-scale quantum (NISQ) era

- Processors have up to a few hundred qubits
- Cannot implement error-correction yet, so have to tolerate noise
- Generally limited connectivity for two-/multi-qubit gates

Longer term: large-scale computation with error correction

- Error-correction schemes generally do not allow any universal gate set to be directly implemented in a fault-tolerant way
- Secure delegation to a quantum server in the cloud

Questions arising from limited quantum computing resources

Currently, and near-term future: noisy intermediate-scale quantum (NISQ) era

- Processors have up to a few hundred qubits
- Cannot implement error-correction yet, so have to tolerate noise
- Generally limited connectivity for two-/multi-qubit gates

Longer term: large-scale computation with error correction

- Error-correction schemes generally do not allow any universal gate set to be directly implemented in a fault-tolerant way
- Secure delegation to a quantum server in the cloud

Questions: compilation, optimisation, routing

Outline

Quantum computing

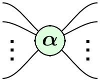
The ZX-calculus

The one-way model of measurement-based quantum computing

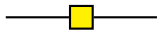
Flow-preserving rewriting

Conclusions

ZX-calculus generators

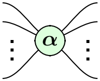


$$\rightsquigarrow \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & e^{i\alpha} \end{pmatrix}$$

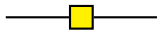


$$\rightsquigarrow \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

ZX-calculus generators

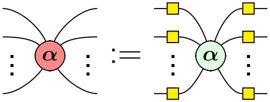


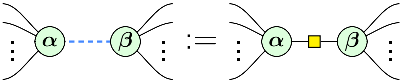
$$\rightsquigarrow \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & e^{i\alpha} \end{pmatrix}$$



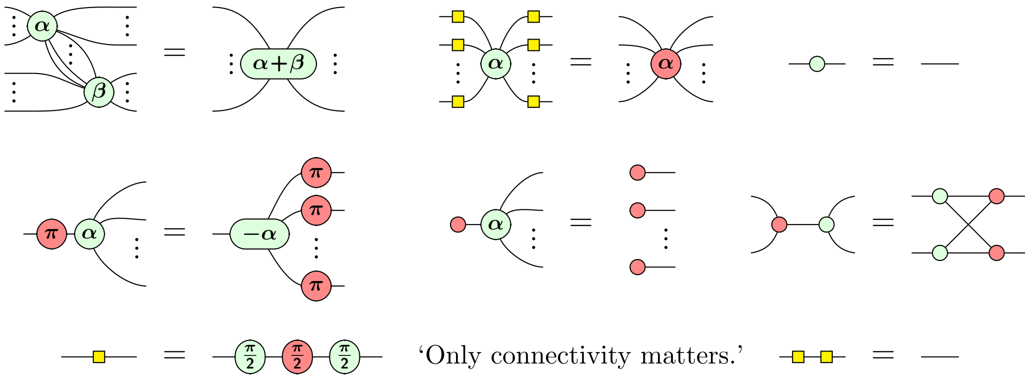
$$\rightsquigarrow \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Common syntactic sugars:



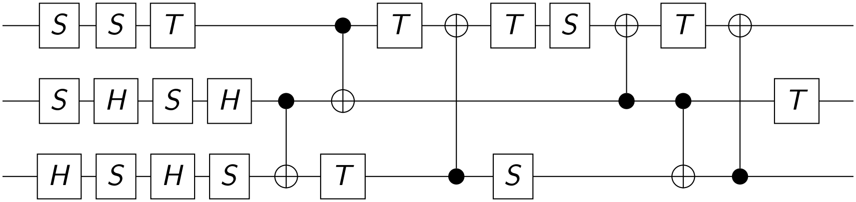


(Some) ZX-calculus rewrite rules

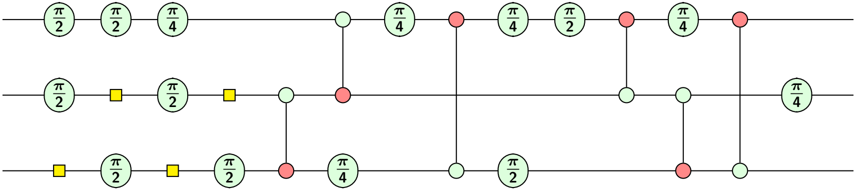


These rules are complete for the 'stabiliser fragment' (ignoring scalars) [B. 2014].
The universal calculus can be made complete [Jeandel et al. 2017; Ng & Wang 2017].

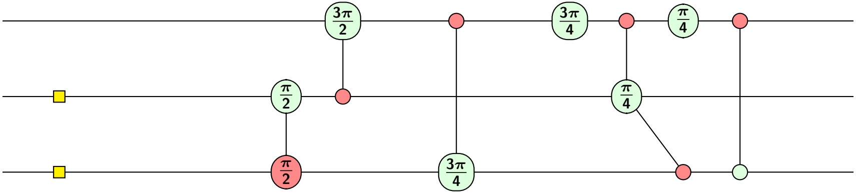
Optimising quantum computations using the ZX-calculus



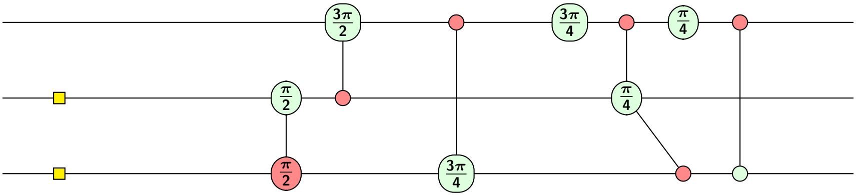
Optimising quantum computations using the ZX-calculus

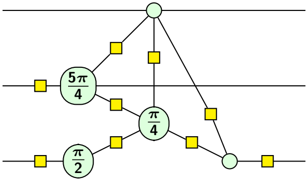


Optimising quantum computations using the ZX-calculus

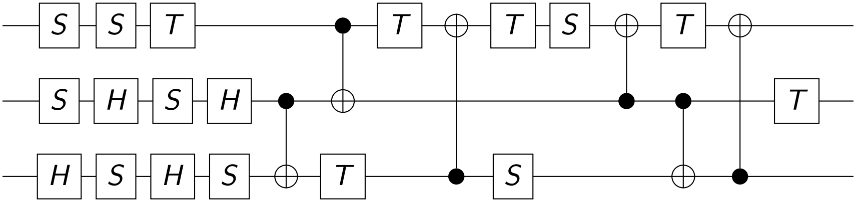


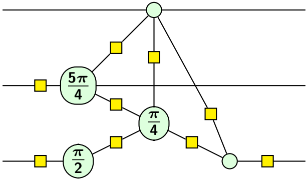
Optimising quantum computations using the ZX-calculus





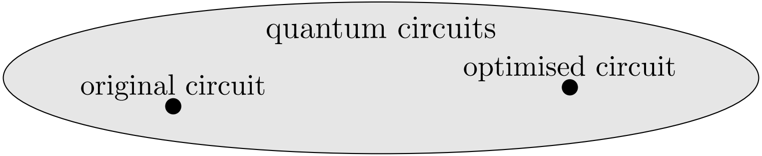
Optimising quantum computations using the ZX-calculus



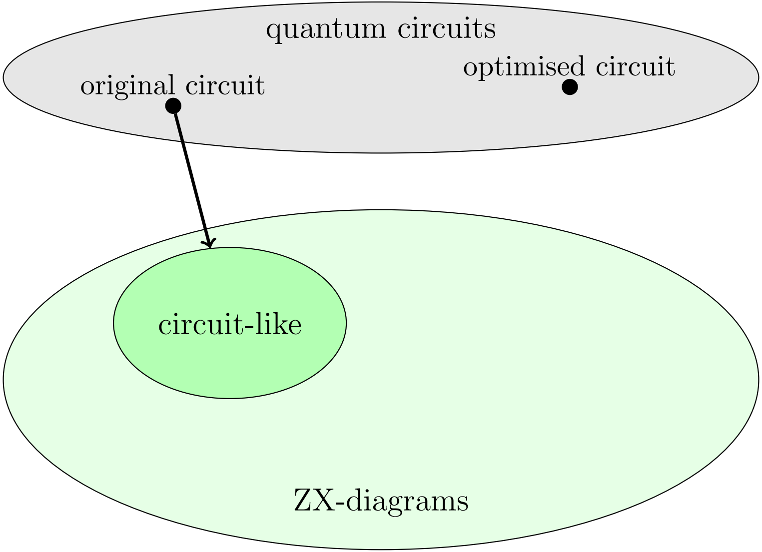


Problem: translating ZX-diagrams to circuits is #P-hard [de Beaudrap et al. 2022]

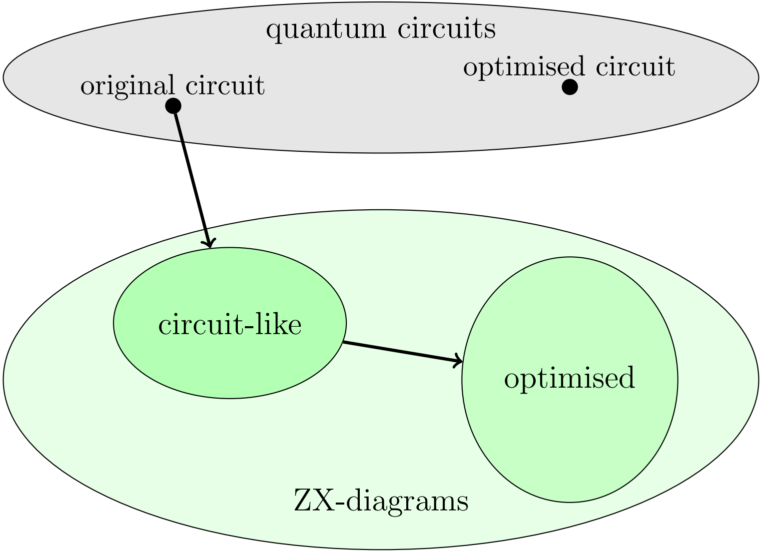
Flow permits efficient circuit extraction



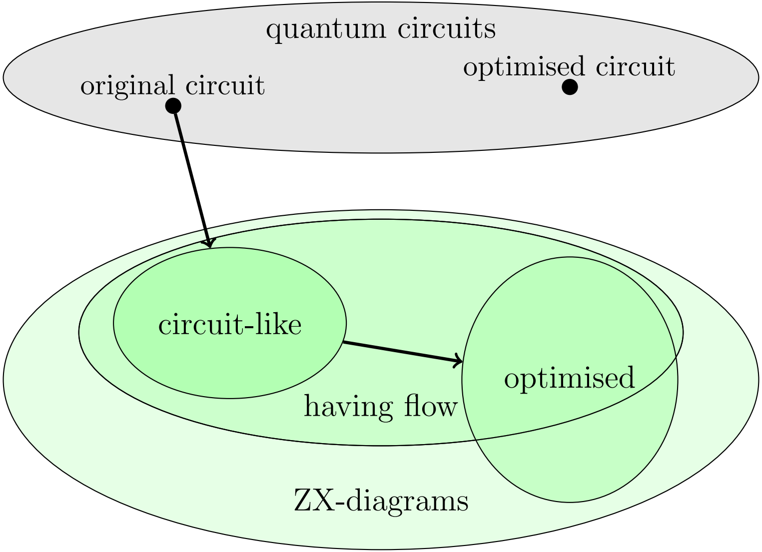
Flow permits efficient circuit extraction



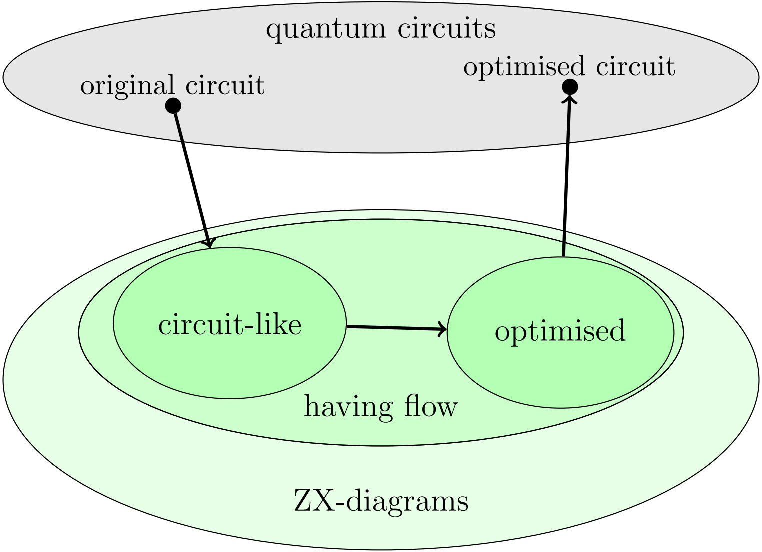
Flow permits efficient circuit extraction



Flow permits efficient circuit extraction



Flow permits efficient circuit extraction



Outline

Quantum computing

The ZX-calculus

The one-way model of measurement-based quantum computing

Flow-preserving rewriting

Conclusions

Two models of quantum computation

Two models of quantum computation

quantum circuit model

Two models of quantum computation

quantum circuit model

- initialise classical state $0 \ldots 0$

Two models of quantum computation

quantum circuit model

- initialise classical state $0 \dots 0$
- computation driven by (reversible) unitary gates

Two models of quantum computation

quantum circuit model

- initialise classical state $0 \dots 0$
- computation driven by (reversible) unitary gates
- simple measurements read out data at end

Two models of quantum computation

quantum circuit model

- initialise classical state $0 \ldots 0$
- computation driven by (reversible) unitary gates
- simple measurements read out data at end

one-way model

Two models of quantum computation

quantum circuit model

- initialise classical state $0 \dots 0$
- computation driven by (reversible) unitary gates
- simple measurements read out data at end

one-way model

- initialise entangled 'graph state' (can be made independent of computation)

Two models of quantum computation

quantum circuit model

- initialise classical state $0 \dots 0$
- computation driven by (reversible) unitary gates
- simple measurements read out data at end

one-way model

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

Two models of quantum computation

quantum circuit model

- initialise classical state $0 \dots 0$
- computation driven by (reversible) unitary gates
- simple measurements read out data at end

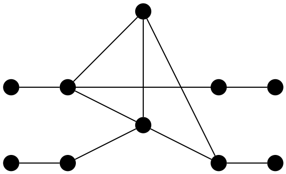
one-way model

- initialise entangled 'graph state' (can be made independent of computation)
- computation driven by successive adaptive single-qubit measurements
- if goal is state preparation, may need very simple unitary gates as correction at the end

Deterministic computation from probabilistic measurements

Deterministic computation from probabilistic measurements

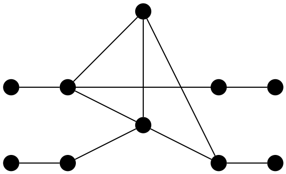
A labelled open graph Γ consists of



Deterministic computation from probabilistic measurements

A labelled open graph Γ consists of

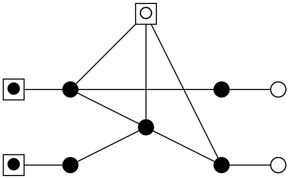
- a finite simple graph $G = (V, E)$,



Deterministic computation from probabilistic measurements

A labelled open graph Γ consists of

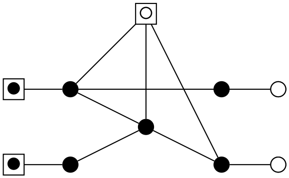
- a finite simple graph $G = (V, E)$,
- subsets $I, O \subseteq V$ called inputs and outputs, and



Deterministic computation from probabilistic measurements

A labelled open graph Γ consists of

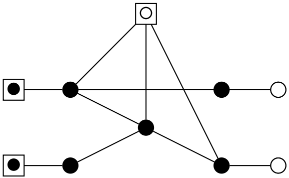
- a finite simple graph $G = (V, E)$,
- subsets $I, O \subseteq V$ called inputs and outputs, and
- one of six 'measurement labels' for each $v \in \overline{O}$.



Deterministic computation from probabilistic measurements

A labelled open graph Γ consists of

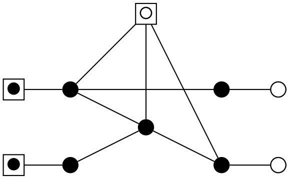
- a finite simple graph $G = (V, E)$,
- subsets $I, O \subseteq V$ called inputs and outputs, and
- one of six 'measurement labels' for each $v \in \overline{O}$.



Deterministic computation from probabilistic measurements

A labelled open graph Γ consists of

- a finite simple graph $G = (V, E)$,
- subsets $I, O \subseteq V$ called inputs and outputs, and
- one of six 'measurement labels' for each $v \in \overline{O}$.



Deterministic computation from probabilistic measurements

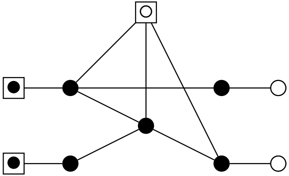
A labelled open graph Γ consists of

- a finite simple graph $G = (V, E)$,
- subsets $I, O \subseteq V$ called inputs and outputs, and
- one of six 'measurement labels' for each $v \in \overline{O}$.

A flow on a labelled open graph Γ consists of

- a partial order $\prec$ over V, and

which satisfy certain compatibility conditions.



Deterministic computation from probabilistic measurements

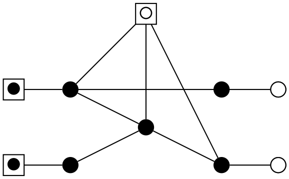
A labelled open graph Γ consists of

- a finite simple graph $G = (V, E)$,
- subsets $I, O \subseteq V$ called inputs and outputs, and
- one of six 'measurement labels' for each $v \in \overline{O}$.

A flow on a labelled open graph Γ consists of

- a partial order $\prec$ over V, and
- a 'correction function' $f : \overline{O} \to \mathcal{P}\left(\overline{I}\right)$,

which satisfy certain compatibility conditions.



Deterministic computation from probabilistic measurements

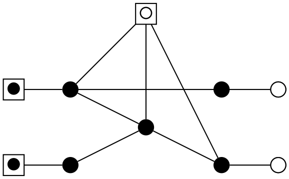
A labelled open graph Γ consists of

- a finite simple graph $G = (V, E)$,
- subsets $I, O \subseteq V$ called inputs and outputs, and
- one of six 'measurement labels' for each $v \in \overline{O}$.

A flow on a labelled open graph Γ consists of

- a partial order $\prec$ over V, and
- a 'correction function' $f : \overline{O} \to \mathcal{P}\left(\overline{I}\right)$,

which satisfy certain compatibility conditions.



Deterministic computation from probabilistic measurements

A labelled open graph Γ consists of

- a finite simple graph $G = (V, E)$,
- subsets $I, O \subseteq V$ called inputs and outputs, and
- one of six 'measurement labels' for each $v \in \overline{O}$.

A flow on a labelled open graph Γ consists of

- a partial order $\prec$ over V, and
- a 'correction function' $f : \overline{O} \to \mathcal{P}\left(\overline{I}\right)$,

which satisfy certain compatibility conditions.

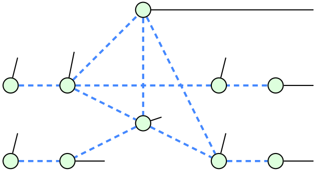
Theorem [Browne et al. 2007, Mhalla et al. 2022]

A one-way-model computation has a robustly deterministic implementation if and only if the underlying labelled open graph has flow.

MBQC-form ZX-diagrams

An MBQC-form ZX-diagram consists of:

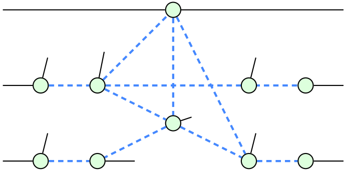
- a graph state diagram



MBQC-form ZX-diagrams

An MBQC-form ZX-diagram consists of:

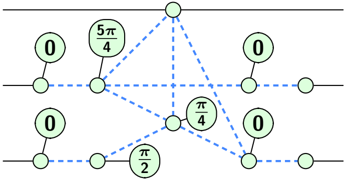
- a graph state diagram
- with some input wires, and



MBQC-form ZX-diagrams

An MBQC-form ZX-diagram consists of:

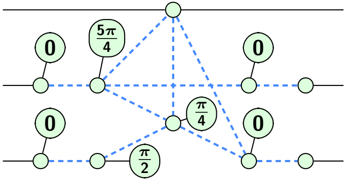
- a graph state diagram
- with some input wires, and
- measurement effects on some outputs.



MBQC-form ZX-diagrams

An MBQC-form ZX-diagram consists of:

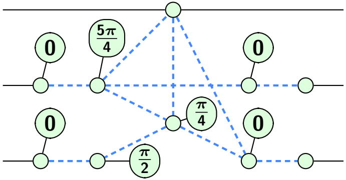
- a graph state diagram
- with some input wires, and
- measurement effects on some outputs.



MBQC-form ZX-diagrams

An MBQC-form ZX-diagram consists of:

- a graph state diagram
- with some input wires, and
- measurement effects on some outputs.



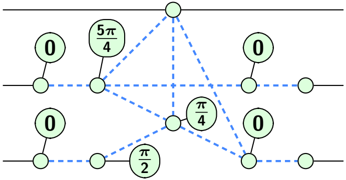
Lemma

Any ZX-diagram can be brought into MBQC form efficiently.

MBQC-form ZX-diagrams

An MBQC-form ZX-diagram consists of:

- a graph state diagram
- with some input wires, and
- measurement effects on some outputs.



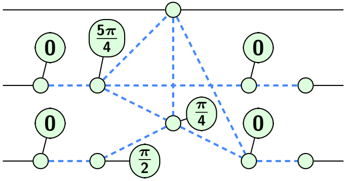
Lemma

Any ZX-diagram can be brought into MBQC form efficiently.

MBQC-form ZX-diagrams

An MBQC-form ZX-diagram consists of:

- a graph state diagram
- with some input wires, and
- measurement effects on some outputs.



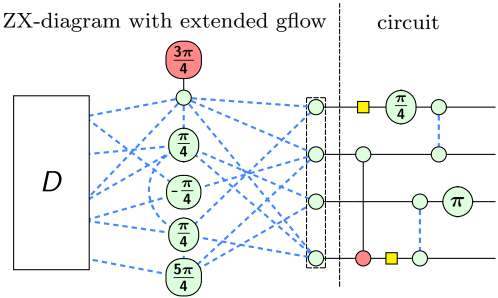
Lemma

Any ZX-diagram can be brought into MBQC form efficiently.

Definition

An MBQC-form ZX-diagram has flow if its underlying labelled open graph has flow.

Efficient circuit extraction from MBQC-form diagrams with flow



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

Outline

Quantum computing

The ZX-calculus

The one-way model of measurement-based quantum computing

Flow-preserving rewriting

Conclusions

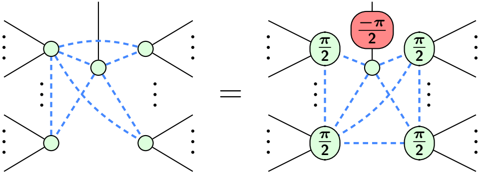
Flow-preserving rewrite rules for the stabiliser ZX-calculus

(all phase labels are integer multiples of $\pi/2$)

Flow-preserving rewrite rules for the stabiliser ZX-calculus

(all phase labels are integer multiples of $\pi/2$)

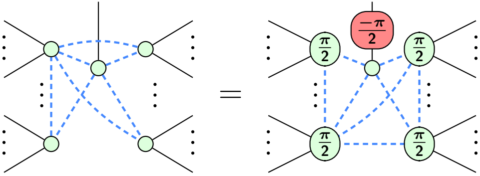
Local complementation



Flow-preserving rewrite rules for the stabiliser ZX-calculus

(all phase labels are integer multiples of $\pi/2$)

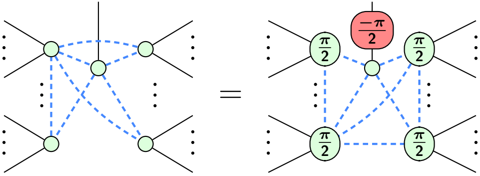
Local complementation



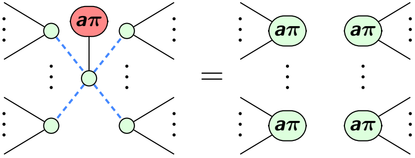
Flow-preserving rewrite rules for the stabiliser ZX-calculus

(all phase labels are integer multiples of $\pi/2$)

Local complementation



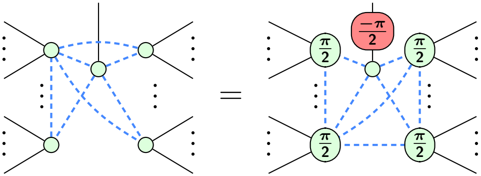
Z-deletion/insertion



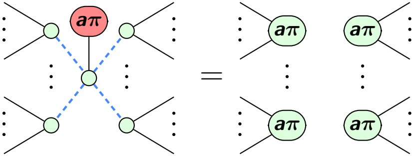
Flow-preserving rewrite rules for the stabiliser ZX-calculus

(all phase labels are integer multiples of $\pi/2$)

Local complementation



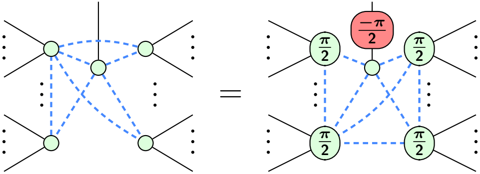
Z-deletion/insertion



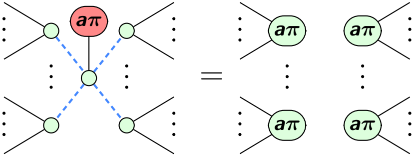
Flow-preserving rewrite rules for the stabiliser ZX-calculus

(all phase labels are integer multiples of $\pi/2$)

Local complementation



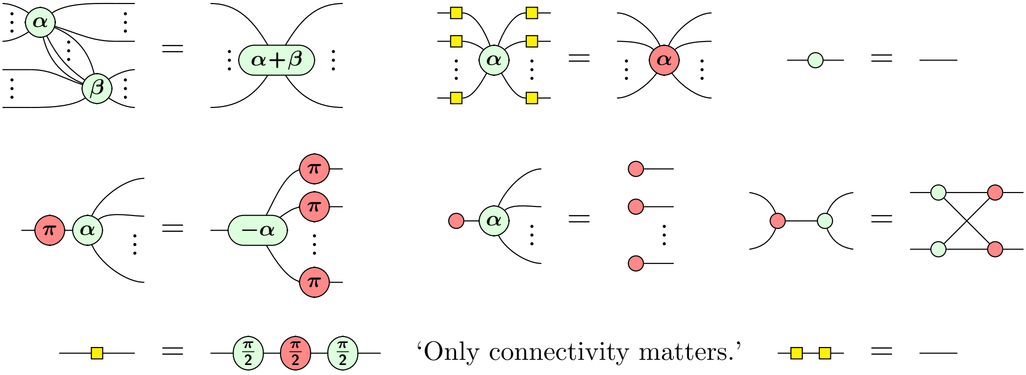
Z-deletion/insertion



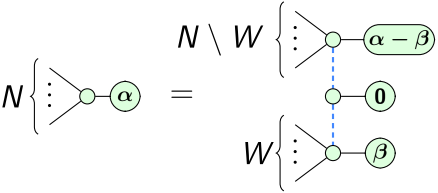
Theorem [McElvanney & B. 2023]

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.

The standard stabiliser ZX-calculus rewrite rules again



A further flow-preserving rewrite rule: vertex splitting



[McElvanney & B. 2023]

Applications

Reduce number of non-stabiliser operations

- Non-stabiliser operations tend to be very expensive in error-correcting codes

Applications

Reduce number of non-stabiliser operations

- Non-stabiliser operations tend to be very expensive in error-correcting codes
- Stabiliser rewrite rules are largely sufficient for this [van de Wetering et al. 2024]

Applications

Reduce number of non-stabiliser operations

- Non-stabiliser operations tend to be very expensive in error-correcting codes
- Stabiliser rewrite rules are largely sufficient for this [van de Wetering et al. 2024]

Applications

Reduce number of non-stabiliser operations

- Non-stabiliser operations tend to be very expensive in error-correcting codes
- Stabiliser rewrite rules are largely sufficient for this [van de Wetering et al. 2024]

Reduce number of two-qubit gates

- Two-qubit gates tend to have larger errors in NISQ devices

Applications

Reduce number of non-stabiliser operations

- Non-stabiliser operations tend to be very expensive in error-correcting codes
- Stabiliser rewrite rules are largely sufficient for this [van de Wetering et al. 2024]

Reduce number of two-qubit gates

- Two-qubit gates tend to have larger errors in NISQ devices
- Vertex splitting rule is useful for reducing counts [Staudacher et al. 2022]

Applications

Reduce number of non-stabiliser operations

- Non-stabiliser operations tend to be very expensive in error-correcting codes
- Stabiliser rewrite rules are largely sufficient for this [van de Wetering et al. 2024]

Reduce number of two-qubit gates

- Two-qubit gates tend to have larger errors in NISQ devices
- Vertex splitting rule is useful for reducing counts [Staudacher et al. 2022]

Applications

Reduce number of non-stabiliser operations

- Non-stabiliser operations tend to be very expensive in error-correcting codes
- Stabiliser rewrite rules are largely sufficient for this [van de Wetering et al. 2024]

Reduce number of two-qubit gates

- Two-qubit gates tend to have larger errors in NISQ devices
- Vertex splitting rule is useful for reducing counts [Staudacher et al. 2022]

More efficient secure delegated computation

- Arbitrary secure delegated quantum computation is possible

Applications

Reduce number of non-stabiliser operations

- Non-stabiliser operations tend to be very expensive in error-correcting codes
- Stabiliser rewrite rules are largely sufficient for this [van de Wetering et al. 2024]

Reduce number of two-qubit gates

- Two-qubit gates tend to have larger errors in NISQ devices
- Vertex splitting rule is useful for reducing counts [Staudacher et al. 2022]

More efficient secure delegated computation

- Arbitrary secure delegated quantum computation is possible
- Traditional schemes use a lot of redundant resources, can reduce that using flow-preserving ZX-calculus rewriting [Cao 2023]

Outline

Quantum computing

The ZX-calculus

The one-way model of measurement-based quantum computing

Flow-preserving rewriting

Conclusions

Summary and outlook

- ZX-calculus is useful for optimising quantum computations

Summary and outlook

- ZX-calculus is useful for optimising quantum computations
- Translating arbitrary ZX-diagrams into quantum circuits is #P-hard

Summary and outlook

- ZX-calculus is useful for optimising quantum computations
- Translating arbitrary ZX-diagrams into quantum circuits is #P-hard
- Translating MBQC-form ZX-diagrams with flow is efficient

Summary and outlook

- ZX-calculus is useful for optimising quantum computations
- Translating arbitrary ZX-diagrams into quantum circuits is #P-hard
- Translating MBQC-form ZX-diagrams with flow is efficient
- New rewrite rules that preserve these properties

Summary and outlook

- ZX-calculus is useful for optimising quantum computations
- Translating arbitrary ZX-diagrams into quantum circuits is #P-hard
- Translating MBQC-form ZX-diagrams with flow is efficient
- New rewrite rules that preserve these properties
- Known applications in optimisation and obfuscation

Summary and outlook

- ZX-calculus is useful for optimising quantum computations
- Translating arbitrary ZX-diagrams into quantum circuits is #P-hard
- Translating MBQC-form ZX-diagrams with flow is efficient
- New rewrite rules that preserve these properties
- Known applications in optimisation and obfuscation

Summary and outlook

- ZX-calculus is useful for optimising quantum computations
- Translating arbitrary ZX-diagrams into quantum circuits is #P-hard
- Translating MBQC-form ZX-diagrams with flow is efficient
- New rewrite rules that preserve these properties
- Known applications in optimisation and obfuscation

Further work

- More work to be done around different types of flow

Summary and outlook

- ZX-calculus is useful for optimising quantum computations
- Translating arbitrary ZX-diagrams into quantum circuits is #P-hard
- Translating MBQC-form ZX-diagrams with flow is efficient
- New rewrite rules that preserve these properties
- Known applications in optimisation and obfuscation

Further work

- More work to be done around different types of flow
- Are there similar properties for ZW-calculus and ZH-calculus?

Summary and outlook

- ZX-calculus is useful for optimising quantum computations
- Translating arbitrary ZX-diagrams into quantum circuits is #P-hard
- Translating MBQC-form ZX-diagrams with flow is efficient
- New rewrite rules that preserve these properties
- Known applications in optimisation and obfuscation

Further work

- More work to be done around different types of flow
- Are there similar properties for ZW-calculus and ZH-calculus?

Summary and outlook

- ZX-calculus is useful for optimising quantum computations
- Translating arbitrary ZX-diagrams into quantum circuits is #P-hard
- Translating MBQC-form ZX-diagrams with flow is efficient
- New rewrite rules that preserve these properties
- Known applications in optimisation and obfuscation

Further work

- More work to be done around different types of flow
- Are there similar properties for ZW-calculus and ZH-calculus?

Thank you!