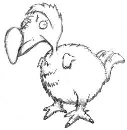
Dario Trinchero
20854714@sun.ac.za

Computing by Collapsing

…or, Measurement-Based Quantum Computing
& the stunning efficiency of graphical calculi

Stellenbosch University
April 2022



Inspiration

Universal MBQC with generalised parity-phase interactions and Pauli measurements

Aleks Kissinger and John van de Wetering

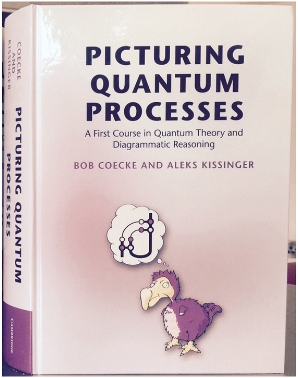
Radboud University Nijmegen
April 17, 2019

We introduce a new family of models for measurement-based quantum computation which are deterministic and approximately universal. The resource states which play the role of graph states are prepared via 2-qubit gates of the form $\exp(-i\frac{\pi}{2^n} Z \otimes Z)$. When $n = 2$, these are equivalent, up to local Clifford unitaries, to graph states. However, when $n > 2$, their behaviour di-

Physics, Topology, Logic and Computation: A Rosetta Stone

John C. Baez
Department of Mathematics, University of California
Riverside, California 92521, USA

Mike Stay



Preliminaries

Crash Course in Quantum Computing

1. QUBIT state $\longrightarrow$ $\boldsymbol{v} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2, \quad |\alpha|^2 + |\beta|^2 = 1$

Crash Course in Quantum Computing

1. **Qubit** state $\longrightarrow$ $\boldsymbol{v} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2, \quad |\alpha|^2 + |\beta|^2 = 1$
2. Single-qubit **gate** $\longrightarrow$ $M \equiv \begin{bmatrix} \kappa & \lambda \\ \mu & \nu \end{bmatrix} \in \mathrm{U}(2), \quad \text{i.e.} \quad M^\dagger = M$

Crash Course in Quantum Computing

1. QUBIT state $\longrightarrow$ $\boldsymbol{v} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2, \quad |\alpha|^2 + |\beta|^2 = 1$
2. Single-qubit GATE $\longrightarrow$ $M \equiv \begin{bmatrix} \kappa & \lambda \\ \mu & \nu \end{bmatrix} \in \mathrm{U}(2), \quad \text{i.e.} \quad M^\dagger = M$

e.g. $$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Crash Course in Quantum Computing

1. Qubit state $\longrightarrow$ $\boldsymbol{v} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2, \quad |\alpha|^2 + |\beta|^2 = 1$

2. Single-qubit gate $\longrightarrow$ $M \equiv \begin{bmatrix} \kappa & \lambda \\ \mu & \nu \end{bmatrix} \in \mathrm{U}(2), \quad \text{i.e.} \quad M^\dagger = M$

   e.g. $Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

3. N-qubit state/gate $\longrightarrow$ $\boldsymbol{v} \in (\mathbb{C}^2)^{\otimes N} \cong \mathbb{C}^{2^N}, \quad M \in \mathrm{U}(2^N)$

Crash Course in Quantum Computing

1. QUBIT state $\longrightarrow$ $\boldsymbol{v} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbb{C}^2, \quad |\alpha|^2 + |\beta|^2 = 1$

2. Single-qubit GATE $\longrightarrow$ $M \equiv \begin{bmatrix} \kappa & \lambda \\ \mu & \nu \end{bmatrix} \in \mathrm{U}(2), \quad \text{i.e.} \quad M^\dagger = M$

   e.g. $$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

3. N-QUBIT state/gate $\longrightarrow$ $\boldsymbol{v} \in (\mathbb{C}^2)^{\otimes N} \cong \mathbb{C}^{2^N}, \quad M \in \mathrm{U}(2^N)$

   e.g. $$CZ := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad CX := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Crash Course in Quantum Computing

Sidenote (Tensor product)

- $$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \otimes \begin{bmatrix}\gamma\\ \delta\end{bmatrix} = \begin{bmatrix}\alpha\begin{bmatrix}\gamma\\ \delta\end{bmatrix}\\ \beta\begin{bmatrix}\gamma\\ \delta\end{bmatrix}\end{bmatrix} = \begin{bmatrix}\alpha\gamma\\ \alpha\delta\\ \beta\gamma\\ \beta\delta\end{bmatrix}$$
- $$\begin{bmatrix}u_{11} & u_{12}\\ u_{21} & u_{22}\end{bmatrix} \otimes \begin{bmatrix}v_{11} & v_{12}\\ v_{21} & v_{22}\end{bmatrix} = \begin{bmatrix}u_{11}\begin{bmatrix}v_{11} & v_{12}\\ v_{21} & v_{22}\end{bmatrix} & u_{12}\begin{bmatrix}v_{11} & v_{12}\\ v_{21} & v_{22}\end{bmatrix}\\ u_{21}\begin{bmatrix}v_{11} & v_{12}\\ v_{21} & v_{22}\end{bmatrix} & u_{22}\begin{bmatrix}v_{11} & v_{12}\\ v_{21} & v_{22}\end{bmatrix}\end{bmatrix}$$
- Hence, $M_1 \otimes M_2(\boldsymbol{u} \otimes \boldsymbol{v}) = (M_1\boldsymbol{u}) \otimes (M_2\boldsymbol{v})$
- $$(\mathbb{C}^2)^{\otimes N} := \mathrm{span}_{\mathbb{C}}\left\{\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n \;\middle|\; v_i = \begin{bmatrix}1\\ 0\end{bmatrix}, \begin{bmatrix}0\\ 1\end{bmatrix}\right\} \cong \mathbb{C}^{2^N}$$

Crash Course in Quantum Computing

4 Measurement of qubit $\boldsymbol{v}$ in basis $\{\boldsymbol{e}_0, \boldsymbol{e}_1\}$:

Crash Course in Quantum Computing

4 MEASUREMENT of qubit $\boldsymbol{v}$ in basis $\{\boldsymbol{e}_0, \boldsymbol{e}_1\}$:

Possible outcomes: $b \in \{0, 1\}$

Crash Course in Quantum Computing

4 MEASUREMENT of qubit $\boldsymbol{v}$ in basis $\{\boldsymbol{e}_0, \boldsymbol{e}_1\}$:

Possible outcomes: $b \in \{0, 1\}$
Possible bases:

$$\left\{\boldsymbol{z}_0 \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{z}_1 \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right\} \qquad \text{“measure } Z\text{”}$$

$$\left\{\boldsymbol{x}_0 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{x}_1 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right\} \qquad \text{“measure } X\text{”}$$

Crash Course in Quantum Computing

4 MEASUREMENT of qubit $\boldsymbol{v}$ in basis $\{\boldsymbol{e}_0, \boldsymbol{e}_1\}$:

Possible outcomes: $b \in \{0, 1\}$
Possible bases:

$$\left\{ \boldsymbol{z}_0 \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{z}_1 \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \qquad \text{“measure } Z\text{”}$$

$$\left\{ \boldsymbol{x}_0 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{x}_1 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\} \qquad \text{“measure } X\text{”}$$

- $\boldsymbol{v} = \alpha\, \boldsymbol{e}_0 + \beta\, \boldsymbol{e}_1 \implies p(b) = \begin{cases} |\alpha|^2 & b = 0 \\ |\beta|^2 & b = 1 \end{cases}$

Crash Course in Quantum Computing

4. MEASUREMENT of qubit $\boldsymbol{v}$ in basis $\{\boldsymbol{e}_0, \boldsymbol{e}_1\}$:

Possible outcomes: $b \in \{0, 1\}$

Possible bases:

$$\left\{ \boldsymbol{z}_0 \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{z}_1 \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \qquad \text{“measure } Z\text{”}$$

$$\left\{ \boldsymbol{x}_0 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{x}_1 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\} \qquad \text{“measure } X\text{”}$$

- $\boldsymbol{v} = \alpha\, \boldsymbol{e}_0 + \beta\, \boldsymbol{e}_1 \implies p(b) = \begin{cases} |\alpha|^2 & b = 0 \\ |\beta|^2 & b = 1 \end{cases}$
- Outcome $b = \begin{cases} 0 \\ 1 \end{cases} \implies$ COLLAPSE to $\begin{cases} \boldsymbol{e}_0 \\ \boldsymbol{e}_1 \end{cases}$

4 MEASUREMENT of qubit $\boldsymbol{v}$ in basis $\{\boldsymbol{e}_0, \boldsymbol{e}_1\}$:

Possible outcomes: $b \in \{0, 1\}$
Possible bases:

$$\left\{ \boldsymbol{z}_0 \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{z}_1 \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \qquad \text{“measure } Z\text{”}$$

$$\left\{ \boldsymbol{x}_0 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{x}_1 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\} \qquad \text{“measure } X\text{”}$$

More generally:

Crash Course in Quantum Computing

4 MEASUREMENT of qubit $\boldsymbol{v}$ in basis $\{\boldsymbol{e}_0, \boldsymbol{e}_1\}$:

Possible outcomes: $b \in \{0, 1\}$

Possible bases:

$$\left\{ \boldsymbol{z}_0 \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{z}_1 \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \qquad \text{“measure } Z\text{”}$$

$$\left\{ \boldsymbol{x}_0 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{x}_1 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\} \qquad \text{“measure } X\text{”}$$

More generally:

- $p(b) = \boldsymbol{v}^\dagger P_b \boldsymbol{v}, \quad P_b := \boldsymbol{e}_b \boldsymbol{e}_b^\dagger,$

Crash Course in Quantum Computing

4 MEASUREMENT of qubit $\boldsymbol{v}$ in basis $\{\boldsymbol{e}_0, \boldsymbol{e}_1\}$:

Possible outcomes: $b \in \{0, 1\}$

Possible bases:

$$\left\{ \boldsymbol{z}_0 \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{z}_1 \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \qquad \text{“measure } Z\text{”}$$

$$\left\{ \boldsymbol{x}_0 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{x}_1 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\} \qquad \text{“measure } X\text{”}$$

More generally:

- $p(b) = \boldsymbol{v}^\dagger P_b \boldsymbol{v}, \quad P_b := \boldsymbol{e}_b \boldsymbol{e}_b^\dagger,$
- Outcome $b \implies$ collapse to $\frac{1}{\sqrt{p(b)}} P_b \boldsymbol{v}$

Crash Course in Quantum Computing

4 MEASUREMENT of qubit $\boldsymbol{v}$ in basis $\{\boldsymbol{e}_0, \boldsymbol{e}_1\}$:

Possible outcomes: $b \in \{0, 1\}$

Possible bases:

$$\left\{ \boldsymbol{z}_0 \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{z}_1 \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \qquad \text{“measure } Z\text{”}$$

$$\left\{ \boldsymbol{x}_0 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{x}_1 \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\} \qquad \text{“measure } X\text{”}$$

More generally:

- $p(b) = \boldsymbol{v}^\dagger P_b \boldsymbol{v}, \quad P_b := \boldsymbol{e}_b \boldsymbol{e}_b^\dagger,$
- Outcome $b \implies$ collapse to $\frac{1}{\sqrt{p(b)}} P_b \boldsymbol{v}$
- QUBIT m in N-QUBIT state $\longrightarrow P_b^{(m)} := I^{\otimes m-1} \otimes P_b \otimes I^{\otimes N-m-1}$

Universal Computation

Theorem (Universal gates)

The following gates are UNIVERSAL*:*

$$H := \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad CZ := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad T := \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$$

Measurement-Based Quantum Computing

Measurement-Based Quantum Computing

The central idea

Computation in MBQC:

1. Start with fixed N-qubit RESOURCE STATE
2. Perform sequence of SINGLE-QUBIT measurements

Computation in MBQC:

1. Start with fixed N-qubit RESOURCE STATE
2. Perform sequence of SINGLE-QUBIT measurements

The ALGORITHM comprises:

1. ORDER of measurements
2. BASIS (Z or X) for each measurement

Measurement-Based Quantum Computing

The central idea

Computation in MBQC:

1. Start with fixed N-qubit RESOURCE STATE
2. Perform sequence of SINGLE-QUBIT measurements

The ALGORITHM comprises:

1. ORDER of measurements
2. BASIS (Z or X) for each measurement

Question: Measurements are destructive & non-deterministic.
Can we realize deterministic quantum algorithms?

Measurement-Based Quantum Computing

The central idea

Computation in MBQC:

1. Start with fixed N-qubit RESOURCE STATE
2. Perform sequence of SINGLE-QUBIT measurements

The ALGORITHM comprises:

1. ORDER of measurements
2. BASIS (Z or X) for each measurement

Question: Measurements are destructive & non-deterministic.
Can we realize deterministic quantum algorithms?
Yes, with FEED-FORWARD.

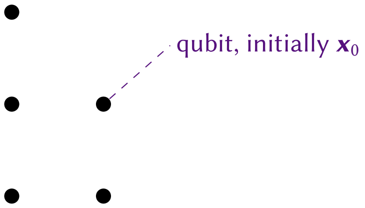
The Resource State

- PARITY-PHASE gate $\longrightarrow$ $P(\alpha) := \exp\left[-i\frac{\alpha}{2}\, Z \otimes Z\right]$

The Resource State

- Parity-phase gate $\longrightarrow$ $P(\alpha) := \exp\left[-i\frac{\alpha}{2} Z \otimes Z\right]$
- Resource state $\longrightarrow$ multigraph

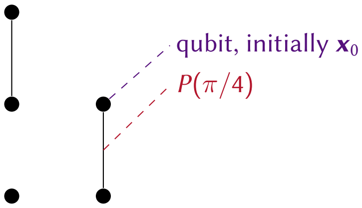
e.g.



The Resource State

- Parity-phase gate $\longrightarrow$ $P(\alpha) := \exp\left[-i\frac{\alpha}{2} Z \otimes Z\right]$
- Resource state $\longrightarrow$ multigraph

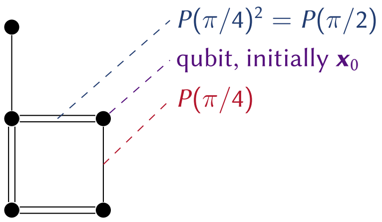
e.g.



The Resource State

- PARITY-PHASE gate $\longrightarrow$ $P(\alpha) := \exp\left[-i\frac{\alpha}{2}\, Z \otimes Z\right]$
- RESOURCE STATE $\longrightarrow$ multigraph

e.g.



Representation

Label vertices with expressions

$$a_{n+1} \leftarrow \phi(a_1, \ldots, a_n),$$

for

- outcomes $a_i \in \{0, 1\}$
- classical function ϕ

Representation

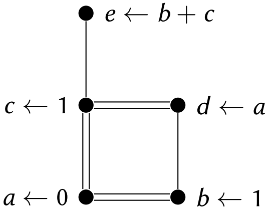
Label vertices with expressions

$$a_{n+1} \leftarrow \phi(a_1, \ldots, a_n),$$

for

- outcomes $a_i \in \{0, 1\}$
- classical function ϕ

e.g.



Representation

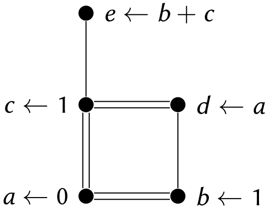
Label vertices with expressions

$$a_{n+1} \leftarrow \phi(a_1, \ldots, a_n),$$

for

- outcomes $a_i \in \{0, 1\}$
- classical function ϕ

e.g.



Execution

1. Topological-sort dependencies

Representation

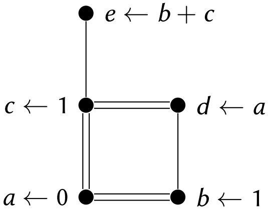
Label vertices with expressions

$$a_{n+1} \leftarrow \phi(a_1, \ldots, a_n),$$

for

- outcomes $a_i \in \{0, 1\}$
- classical function ϕ

e.g.



Execution

1. TOPOLOGICAL-SORT dependencies
2. For each qubit, measure

$$\begin{cases} X & \phi(a_1, \ldots, a_n) = 0 \\ Z & \text{otherwise} \end{cases}$$

This is FEED-FORWARD

Representation

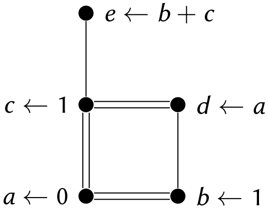
Label vertices with expressions

$$a_{n+1} \leftarrow \phi(a_1, \ldots, a_n),$$

for

- outcomes $a_i \in \{0, 1\}$
- classical function ϕ

e.g.



Execution

1. TOPOLOGICAL-SORT dependencies
2. For each qubit, measure

$$\begin{cases} X & \phi(a_1, \ldots, a_n) = 0 \\ Z & \text{otherwise} \end{cases}$$

   This is FEED-FORWARD

3. Store outcome in a_{n+1}

Representation

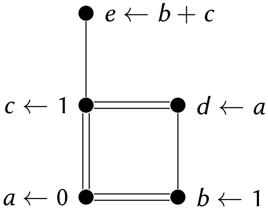
Label vertices with expressions

$$a_{n+1} \leftarrow \phi(a_1, \ldots, a_n),$$

for

- outcomes $a_i \in \{0, 1\}$
- classical function ϕ

e.g.



Execution

1. TOPOLOGICAL-SORT dependencies
2. For each qubit, measure

$$\begin{cases} X & \phi(a_1, \ldots, a_n) = 0 \\ Z & \text{otherwise} \end{cases}$$

   This is FEED-FORWARD

3. Store outcome in a_{n+1}

Question: Can we decompose an algorithm into “gates”?

The "Gates"

a.k.a. Pattern fragments

A PATTERN FRAGMENT implementing gate G has:

The “Gates”

a.k.a. Pattern fragments

A PATTERN FRAGMENT implementing gate G has:

Inputs

- label: $(x, z) \leftarrow *$
- INITIAL STATE: $X^x Z^z \boldsymbol{v}$

The "Gates"

a.k.a. Pattern fragments

A PATTERN FRAGMENT implementing gate G has:

Inputs

- label: $(x, z) \leftarrow *$
- INITIAL STATE: $X^x Z^z \boldsymbol{v}$

Outputs

- label: $* \leftarrow (\xi, \zeta)$
- FINAL STATE: $X^\xi Z^\zeta G\boldsymbol{v}$

incoming/outgoing errors

The “Gates”

a.k.a. Pattern fragments

A PATTERN FRAGMENT implementing gate G has:

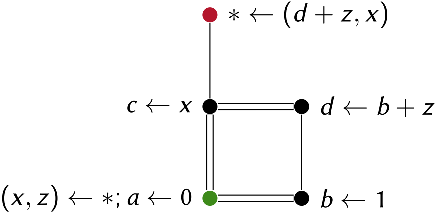
Inputs

- label: $(x, z) \leftarrow *$
- INITIAL STATE: $X^x Z^z \boldsymbol{v}$

Outputs

- label: $* \leftarrow (\xi, \zeta)$
- FINAL STATE: $X^\xi Z^\zeta G\boldsymbol{v}$

e.g.



The "Gates"

a.k.a. Pattern fragments

A PATTERN FRAGMENT implementing gate G has:

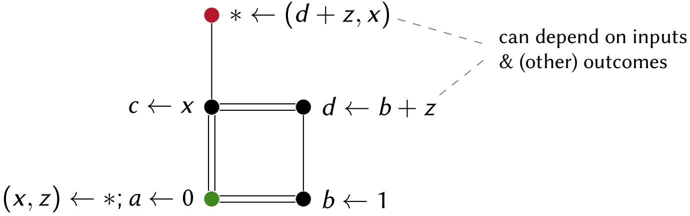
Inputs

- label: $(x, z) \leftarrow *$
- INITIAL STATE: $X^x Z^z \boldsymbol{v}$

Outputs

- label: $* \leftarrow (\xi, \zeta)$
- FINAL STATE: $X^\xi Z^\zeta G\boldsymbol{v}$

e.g.



Universality of MBQC

Claim

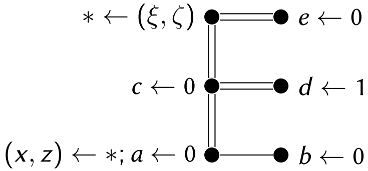
There are pattern fragments for $\{H, T, CZ\}$:

Universality of MBQC

Claim

There are pattern fragments for $\{H, T, \mathrm{C}Z\}$:

$H \equiv$



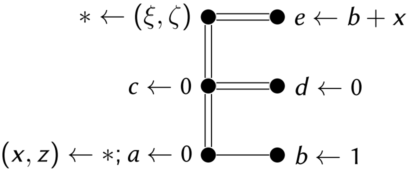
$$\text{where } \begin{cases} \xi = a + b + c + d + z + 1 \\ \zeta = c + d + e + x + 1 \end{cases}$$

Universality of MBQC

Claim

There are pattern fragments for $\{H, T, \mathrm{C}Z\}$:

$T \equiv$



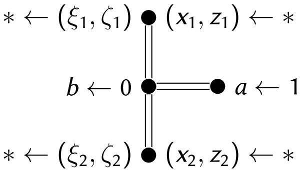
$$\text{where } \begin{cases} \xi = c + d + z + 1 \\ \zeta = a + c + d + e + x + \xi \cdot (b + z) \end{cases}$$

Universality of MBQC

Claim

There are pattern fragments for $\{H, T, CZ\}$:

$CZ \equiv$



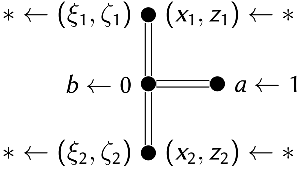
$$\text{where } \begin{cases} \xi_i = x_i \\ \zeta_1 = a + b + 1 + z_1 + x_2 \\ \zeta_2 = a + b + 1 + z_2 + x_1 \end{cases}$$

Universality of MBQC

Claim

There are pattern fragments for $\{H, T, CZ\}$:

$CZ \equiv$



$$\text{where } \begin{cases} \xi_i = x_i \\ \zeta_1 = a + b + 1 + z_1 + x_2 \\ \zeta_2 = a + b + 1 + z_2 + x_1 \end{cases}$$

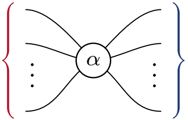
We will now prove this.

ZX-Calculus

Introducing ZX-notation

Meet the spiders

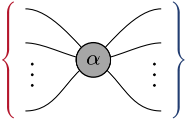
$Z_n^m(\alpha) := m$



n

$$= \boldsymbol{z}_0^{\otimes n}(\boldsymbol{z}_0^\dagger)^{\otimes m} + e^{i\alpha}\, \boldsymbol{z}_1^{\otimes n}(\boldsymbol{z}_1^\dagger)^{\otimes m},$$

$X_n^m(\alpha) := m$

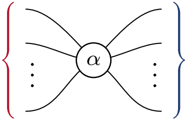


n

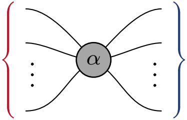
$$= \boldsymbol{x}_0^{\otimes n}(\boldsymbol{x}_0^\dagger)^{\otimes m} + e^{i\alpha}\, \boldsymbol{x}_1^{\otimes n}(\boldsymbol{x}_1^\dagger)^{\otimes m}$$

Introducing ZX-notation

Meet the spiders

$Z_n^m(\alpha) :=$ m  n

$$= \boldsymbol{z}_0^{\otimes n}(\boldsymbol{z}_0^\dagger)^{\otimes m} + e^{i\alpha}\, \boldsymbol{z}_1^{\otimes n}(\boldsymbol{z}_1^\dagger)^{\otimes m},$$

$X_n^m(\alpha) :=$ m  n

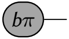
$$= \boldsymbol{x}_0^{\otimes n}(\boldsymbol{x}_0^\dagger)^{\otimes m} + e^{i\alpha}\, \boldsymbol{x}_1^{\otimes n}(\boldsymbol{x}_1^\dagger)^{\otimes m}$$

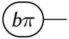
They compose HORIZONTALLY & VERTICALLY.

States & Gates in ZX-notation

. . . and the ambiguous flow of time

Ignoring constants,

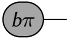
- STATES: $\boldsymbol{z}_b =$ 

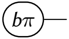
, $\boldsymbol{x}_b =$ 

States & Gates in ZX-notation

…and the ambiguous flow of time

Ignoring constants,

- STATES: $\boldsymbol{z}_b =$ 

, $\boldsymbol{x}_b =$ 

- GATES:

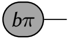
$Z^b =$ 

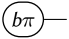
, $X^b =$ 

States & Gates in ZX-notation

…and the ambiguous flow of time

Ignoring constants,

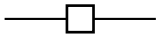
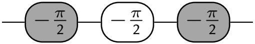
- STATES: $\boldsymbol{z}_b =$ 

 , $\boldsymbol{x}_b =$ 

- GATES:

$Z^b =$ 

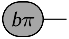
 , $X^b =$ 

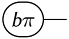
$H \equiv$  $=$ 

States & Gates in ZX-notation

…and the ambiguous flow of time

Ignoring constants,

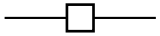
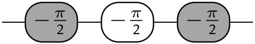
- STATES: $\boldsymbol{z}_b =$ 

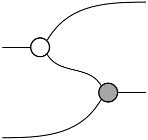
, $\boldsymbol{x}_b =$ 

- GATES:

$Z^b =$ 

, $X^b =$ 

$H \equiv$  $=$ 

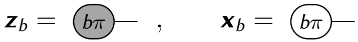
$CX =$ 

States & Gates in ZX-notation

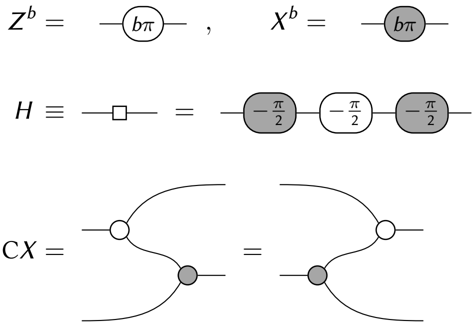
...and the ambiguous flow of time

Ignoring constants,

- STATES:



- GATES:

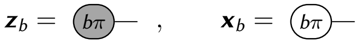


States & Gates in ZX-notation

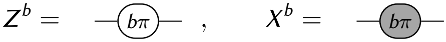
. . . and the ambiguous flow of time

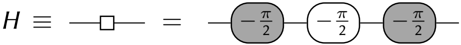
Ignoring constants,

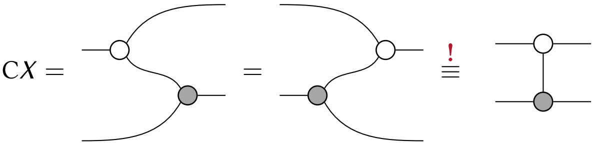
- STATES:



- GATES:

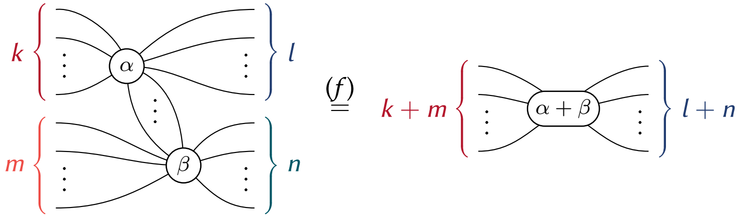






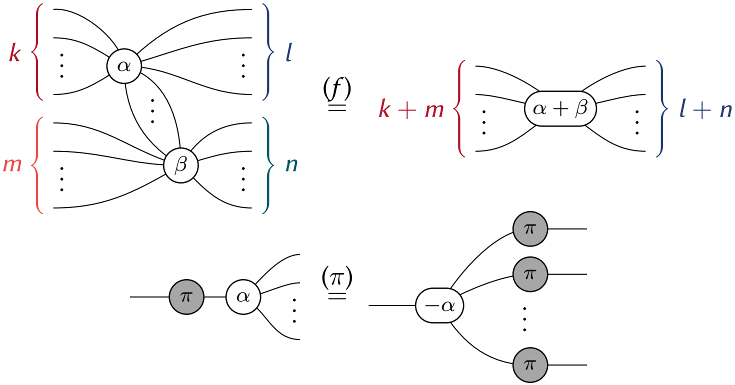
Introducing ZX-Calculus

Dance of the spiders



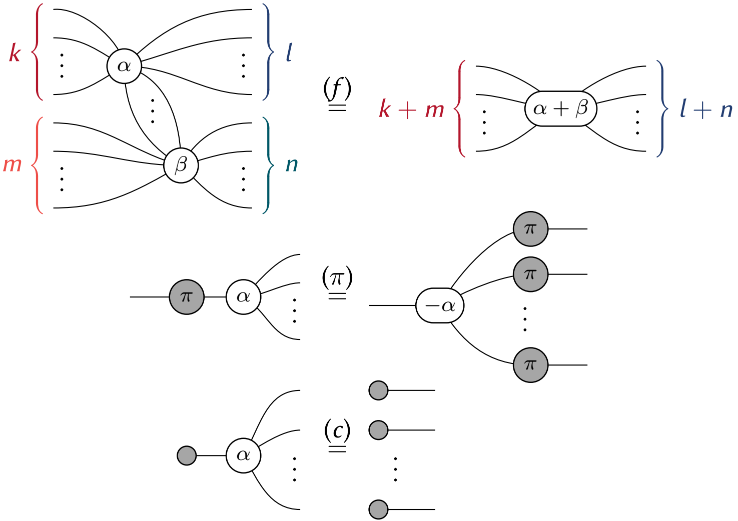
Introducing ZX-Calculus

Dance of the spiders



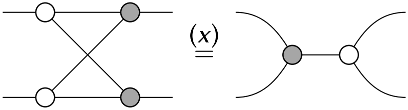
Introducing ZX-Calculus

Dance of the spiders



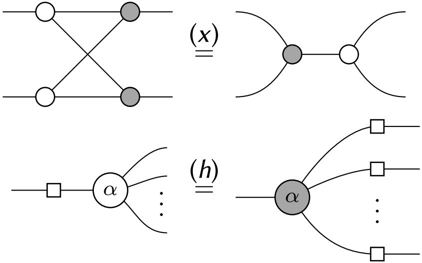
Introducing ZX-Calculus

Dance of the spiders



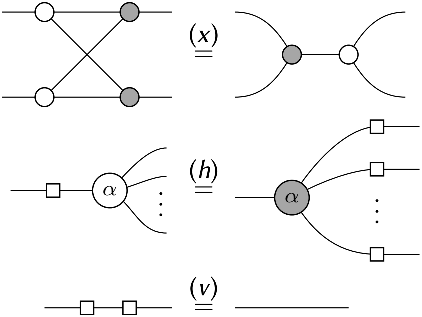
Introducing ZX-Calculus

Dance of the spiders



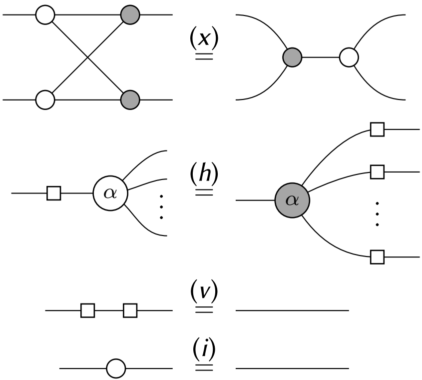
Introducing ZX-Calculus

Dance of the spiders



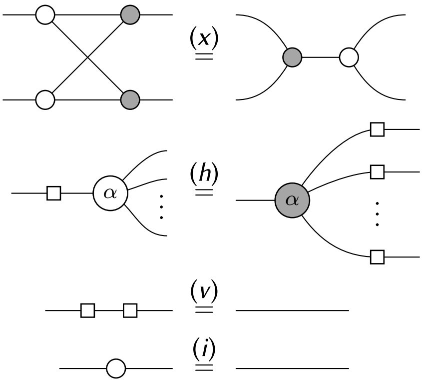
Introducing ZX-Calculus

Dance of the spiders



Introducing ZX-Calculus

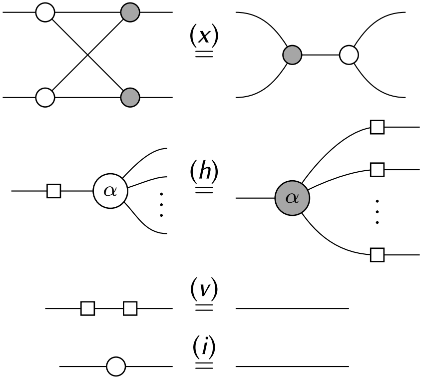
Dance of the spiders



- (h) & (v) $\implies$ COLOUR-INVERSIONS of all rules

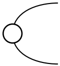
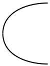
Introducing ZX-Calculus

Dance of the spiders

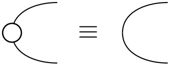


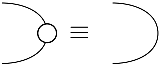
- (h) & (v) $\implies$ COLOUR-INVERSIONS of all rules
- ONLY TOPOLOGY MATTERS

"Only Topology Matters"

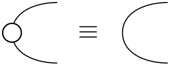
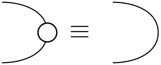
- **Bell state**:  $\equiv$  $= \boldsymbol{z}_0^{\otimes 2} + \boldsymbol{z}_1^{\otimes 2}$ "cup"

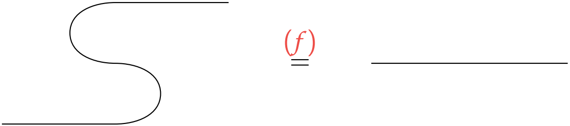
“Only Topology Matters”

- Bell state:  $= z_0^{\otimes 2} + z_1^{\otimes 2}$ “cup”

- Bell effect:  $= (z_0^\dagger)^{\otimes 2} + (z_1^\dagger)^{\otimes 2}$ “cap”

"Only Topology Matters"

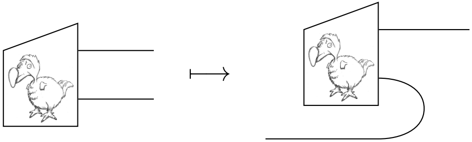
- **Bell state:**  $= \boldsymbol{z}_0^{\otimes 2} + \boldsymbol{z}_1^{\otimes 2}$ "cup"
- **Bell effect:**  $= (\boldsymbol{z}_0^{\dagger})^{\otimes 2} + (\boldsymbol{z}_1^{\dagger})^{\otimes 2}$ "cap"
- **Yanking identity:**



"Only Topology Matters"

Cups/caps can:

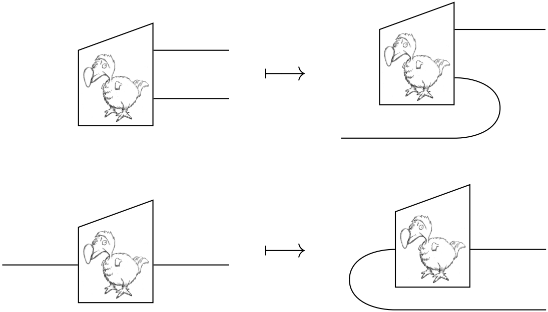
1. convert INPUTS $\longleftrightarrow$ OUTPUTS:



"Only Topology Matters"

Cups/caps can:

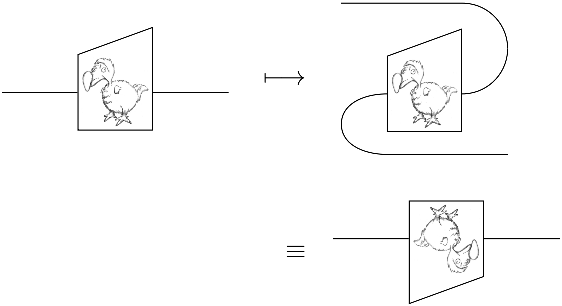
1. convert INPUTS $\longleftrightarrow$ OUTPUTS:



"Only Topology Matters"

Cups/caps can:

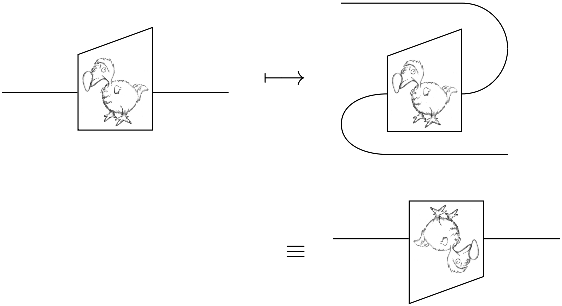
2 define ROTATIONS:



“Only Topology Matters”

Cups/caps can:

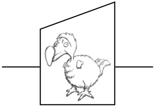
2 define ROTATIONS:



a.k.a. TRANSPOSE

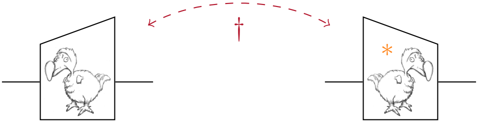
“Only Topology Matters”

Geometric operations:



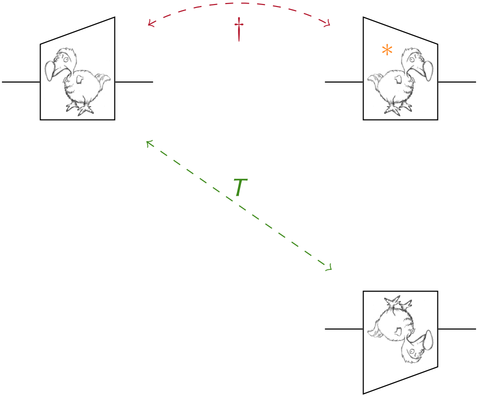
“Only Topology Matters”

Geometric operations:



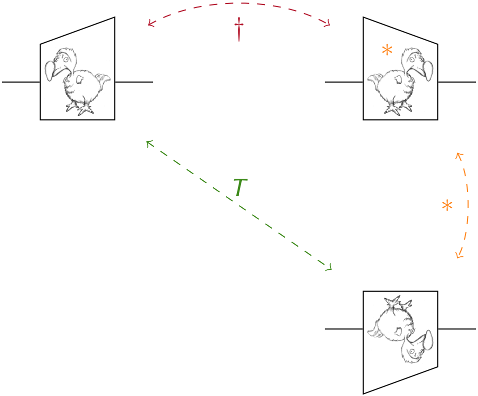
“Only Topology Matters”

Geometric operations:



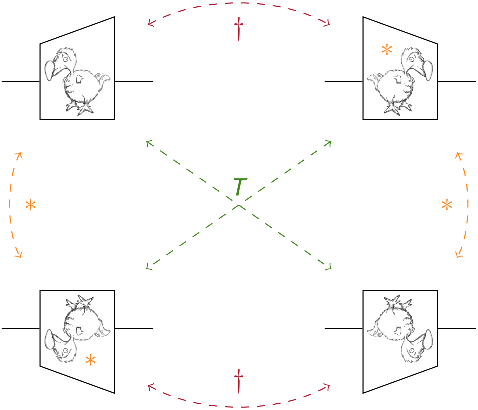
"Only Topology Matters"

Geometric operations:



“Only Topology Matters”

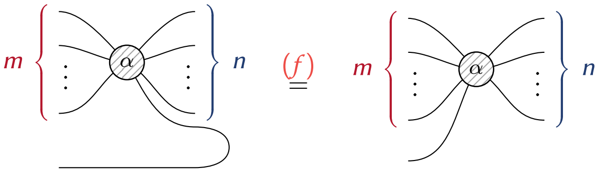
Geometric operations:



"Only Topology Matters"

Cups/caps can:

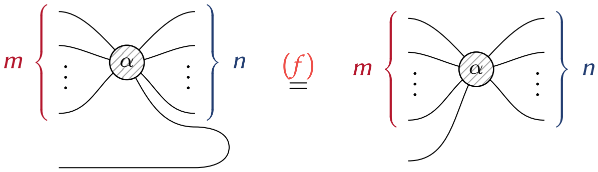
3. BEND spider legs:



“Only Topology Matters”

Cups/caps can:

3 BEND spider legs:



$\Longrightarrow$ we can “REORDER” connected spiders!

The Power of Pictures

Can we do everything in ZX-Calculus?

Theorem (Universality of ZX-notation)

All $\left(\begin{array}{c}\textit{pure states}\\ \textit{quantum gates}\\ \textit{measurements}^{*}\end{array}\right)$ *can be expressed in ZX-notation.*

Indeed, $\left\{\begin{array}{c}\textit{quantum}\\ \textit{circuits}\end{array}\right\} \subset \left\{\textit{ZX-diagrams}\right\}$

Theorem (Universality of ZX-notation)

$$\textit{All} \begin{pmatrix} \textit{pure states} \\ \textit{quantum gates} \\ \textit{measurements}^{*} \end{pmatrix} \textit{can be expressed in ZX-notation.}$$

$$\textit{Indeed,} \begin{Bmatrix} \textit{quantum} \\ \textit{circuits} \end{Bmatrix} \subset \{\textit{ZX-diagrams}\}$$

Theorem (Soundness of ZX-Calculus)

$$\begin{pmatrix} \textit{diagram } A \\ \overset{\textit{ZX-calculus}}{=} \\ \textit{diagram } B \end{pmatrix} \Longrightarrow \begin{pmatrix} \textit{matrix } A \\ = \\ \textit{matrix } B \end{pmatrix}$$

Theorem (Universality of ZX-notation)

All $\begin{pmatrix} \textit{pure states} \\ \textit{quantum gates} \\ \textit{measurements}^* \end{pmatrix}$ *can be expressed in ZX-notation.*

Indeed, $\begin{Bmatrix} \textit{quantum} \\ \textit{circuits} \end{Bmatrix} \subset \{\textit{ZX-diagrams}\}$

Theorem (Soundness & Completeness of ZX-Calculus)

$$\begin{pmatrix} \textit{diagram } A \\ \overset{\textit{ZX-calculus}^*}{=} \\ \textit{diagram } B \end{pmatrix} \iff \begin{pmatrix} \textit{matrix } A \\ = \\ \textit{matrix } B \end{pmatrix}$$

Proving Universality of MBQC

Parity-Phase Gate in ZX-notation

$$P(\alpha) := \exp\left[-i\frac{\alpha}{2} Z \otimes Z\right] = e^{-i\frac{\alpha}{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\alpha} & 0 & 0 \\ 0 & 0 & e^{i\alpha} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Parity-Phase Gate in ZX-notation

$$P(\alpha) := \exp\left[-i\frac{\alpha}{2} Z \otimes Z\right] = e^{-i\frac{\alpha}{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\alpha} & 0 & 0 \\ 0 & 0 & e^{i\alpha} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

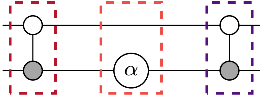
$$= \mathrm{C}X \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\alpha} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\alpha} \end{bmatrix} \mathrm{C}X = \mathrm{C}X \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix} \mathrm{C}X$$

Parity-Phase Gate in ZX-notation

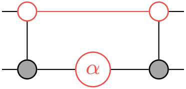
$$P(\alpha) := \exp\left[-i\frac{\alpha}{2} Z \otimes Z\right] = e^{-i\frac{\alpha}{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\alpha} & 0 & 0 \\ 0 & 0 & e^{i\alpha} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= CX \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\alpha} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\alpha} \end{bmatrix} CX = CX \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix} CX$$

$$= CX \ (I \otimes Z_1^1(\alpha)) \ CX$$

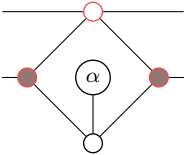
$=$ 

Parity-Phase Gate in ZX-notation

$P(\alpha) =$ 

Coming up: (f)

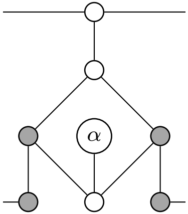
Parity-Phase Gate in ZX-notation

$P(\alpha) =$ 

Coming up: (f)

Parity-Phase Gate in ZX-notation

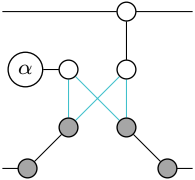
$P(\alpha) =$



Coming up: rearrange

Parity-Phase Gate in ZX-notation

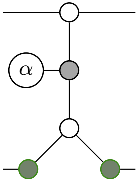
$P(\alpha) =$



Coming up: (x)

Parity-Phase Gate in ZX-notation

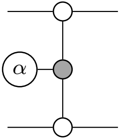
$P(\alpha) =$



Coming up: (i)

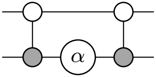
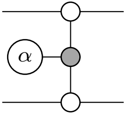
Parity-Phase Gate in ZX-notation

$P(\alpha) =$



Parity-Phase Gate in ZX-notation

$$P(\alpha) = \mathrm{C}X \; (I \otimes Z_1^1(\alpha)) \; \mathrm{C}X$$

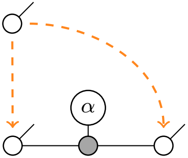
$=$  $=$  .

Pattern Fragments in ZX-notation

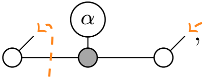
- VERTICES $\longrightarrow$ $\boldsymbol{x}_0 =$ 

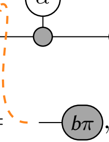
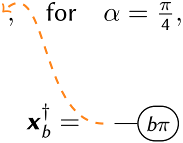
Pattern Fragments in ZX-notation

- VERTICES $\longrightarrow$ $\mathbf{x}_0 =$
- EDGES $\longrightarrow$ $P(\alpha) =$, for $\alpha = \frac{\pi}{4}, \frac{\pi}{2}$

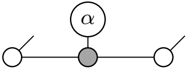


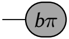
Pattern Fragments in ZX-notation

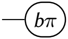
- VERTICES $\longrightarrow$ $\boldsymbol{x}_0 =$ 
- EDGES $\longrightarrow$ $P(\alpha) =$  , for $\alpha = \frac{\pi}{4}, \frac{\pi}{2}$

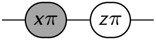
- MEASUREMENTS $\longrightarrow$ $\boldsymbol{z}_b^\dagger =$ , $\boldsymbol{x}_b^\dagger =$ 

Pattern Fragments in ZX-notation

- VERTICES $\longrightarrow$ $\boldsymbol{x}_0 =$ 
- EDGES $\longrightarrow$ $P(\alpha) =$ 

, for $\alpha = \frac{\pi}{4}, \frac{\pi}{2}$
- MEASUREMENTS $\longrightarrow$ $\boldsymbol{z}_b^\dagger =$ 

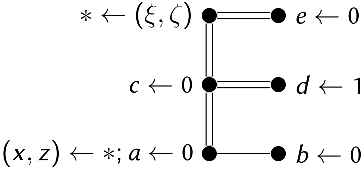
, $\boldsymbol{x}_b^\dagger =$ 

- ERRORS $\longrightarrow$ $X^x Z^z =$ 

Algorithms in ZX-notation

Worked example

$H \equiv$

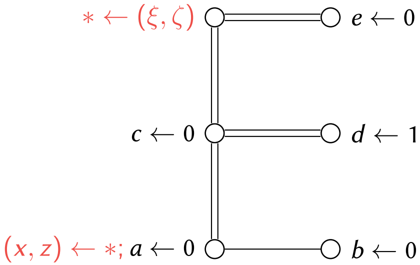


$$\text{where } \begin{cases} \xi = a + b + c + d + z + 1 \\ \zeta = c + d + e + x + 1 \end{cases}$$

Algorithms in ZX-notation

Worked example

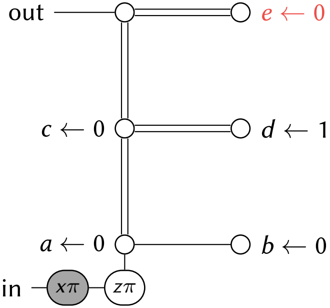
$H \equiv$



Algorithms in ZX-notation

Worked example

$H \equiv$

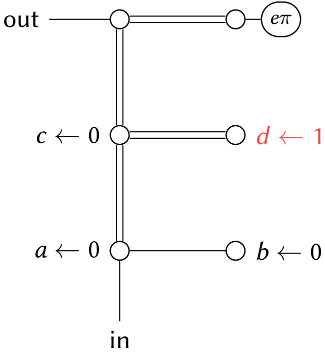


hide incoming errors for now…

Algorithms in ZX-notation

Worked example

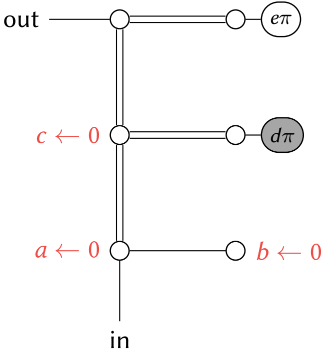
$H \equiv$



Algorithms in ZX-notation

Worked example

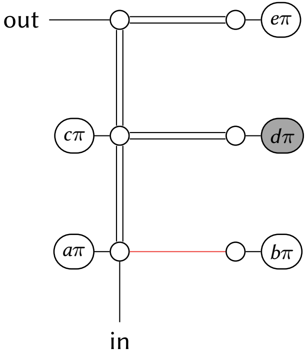
$H \equiv$



Algorithms in ZX-notation

Worked example

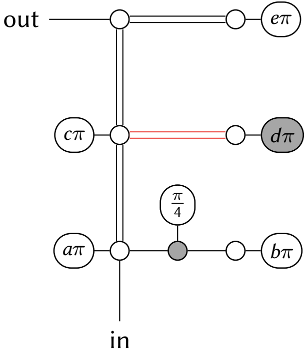
$H \equiv$



Algorithms in ZX-notation

Worked example

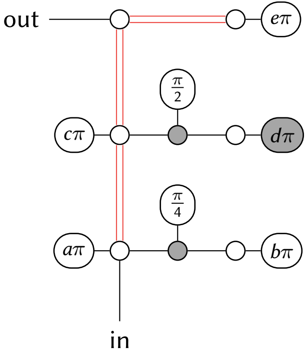
$H \equiv$



Algorithms in ZX-notation

Worked example

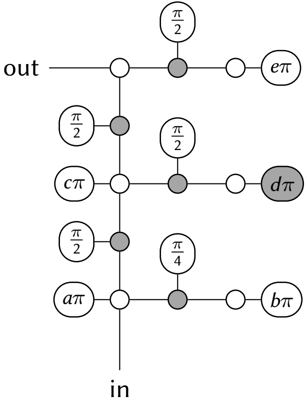
$H \equiv$



Algorithms in ZX-notation

Worked example

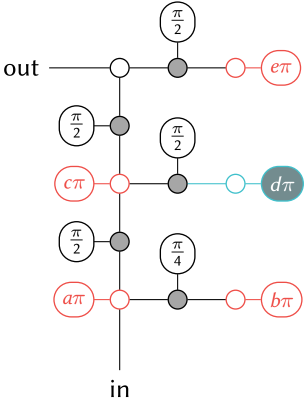
$H \equiv$



Checking Algorithms with ZX-Calculus

Worked example

$H \overset{?}{=}$

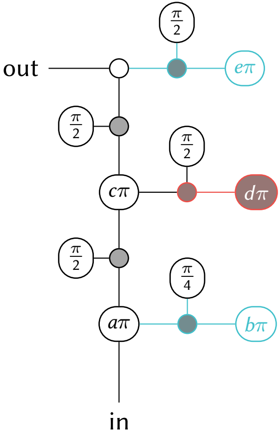


Coming up: (f), $(c)^*$

Checking Algorithms with ZX-Calculus

Worked example

$H \stackrel{?}{=}$

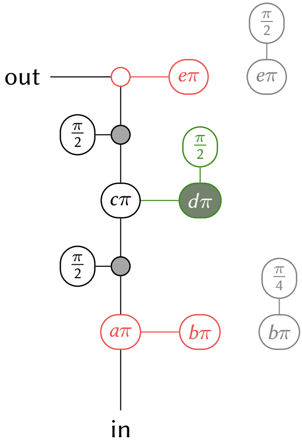


Coming up: (f), $(c)^*$

Checking Algorithms with ZX-Calculus

Worked example

$H \stackrel{?}{=}$

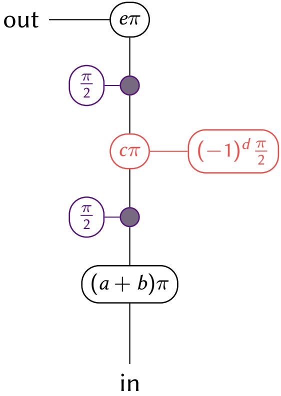


Coming up: (f), (π), ditch constants

Checking Algorithms with ZX-Calculus

Worked example

$H \overset{?}{=}$



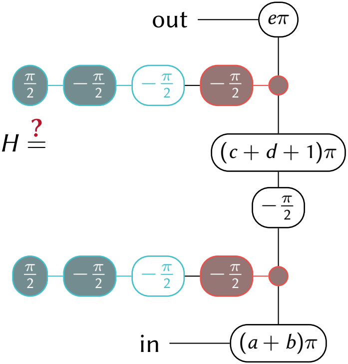
Note, $c\pi + (-1)^d \frac{\pi}{2}$
$= (c + d + 1)\pi - \frac{\pi}{2}$

Coming up: (f), (h), mod 2 trick

Checking Algorithms with ZX-Calculus

Worked example

$H \stackrel{?}{=}$

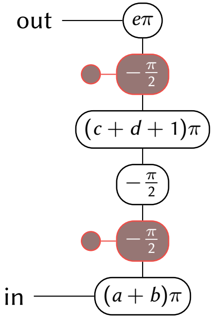


Coming up: (f), $(f) + (c)$

Checking Algorithms with ZX-Calculus

Worked example

$H \overset{?}{=}$

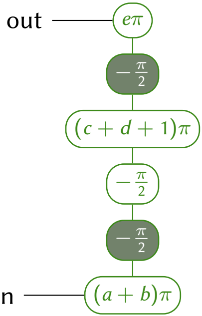


Coming up: (f)

Checking Algorithms with ZX-Calculus

Worked example

$H \overset{?}{=}$

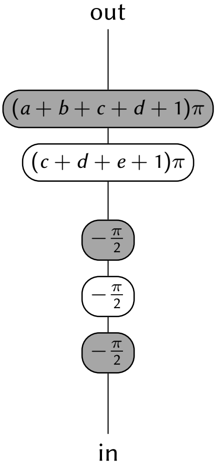


Coming up: (π), mod 2 trick

Checking Algorithms with ZX-Calculus

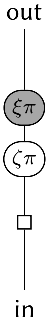
Worked example

$$H \stackrel{?}{=}$$



Checking Algorithms with ZX-Calculus

Worked example

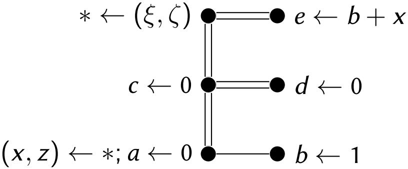
$H \stackrel{\checkmark}{=}$ 

(incoming errors reintroduced)

Other Universal Gates

Not-so-worked examples

$T \equiv$

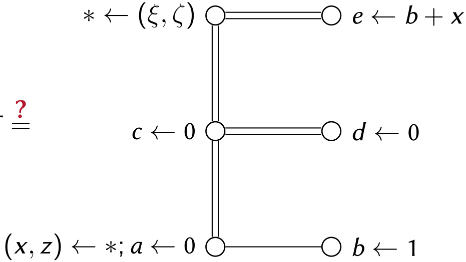


$$\text{where } \begin{cases} \xi = c + d + z + 1 \\ \zeta = a + c + d + e + x + \xi \cdot (b + z) \end{cases}$$

Other Universal Gates

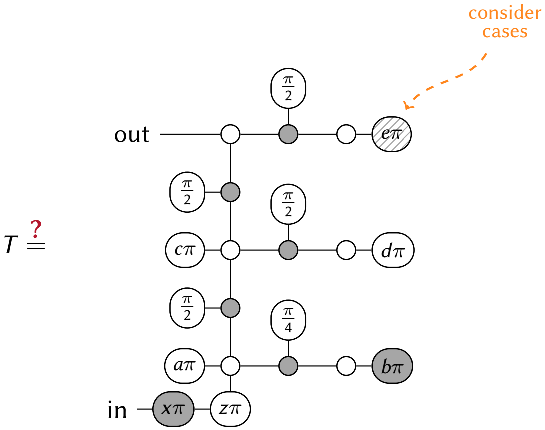
Not-so-worked examples

$T \overset{?}{=}$



Other Universal Gates

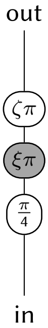
Not-so-worked examples



Other Universal Gates

Not-so-worked examples

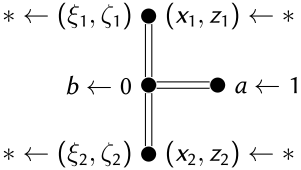
$T \overset{\checkmark}{=}$



Other Universal Gates

Not-so-worked examples

$\mathrm{C}Z \equiv$

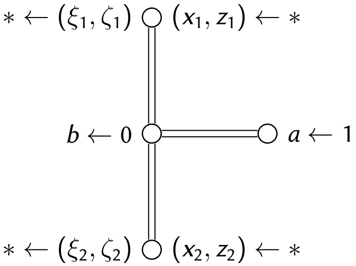


$$\text{where } \begin{cases} \xi_i = x_i \\ \zeta_1 = a + b + 1 + z_1 + x_2 \\ \zeta_2 = a + b + 1 + z_2 + x_1 \end{cases}$$

Other Universal Gates

Not-so-worked examples

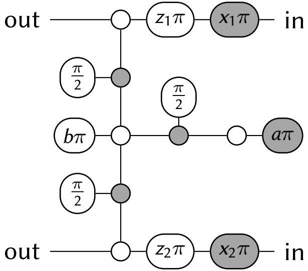
$\mathrm{C}Z \stackrel{?}{=}$



Other Universal Gates

Not-so-worked examples

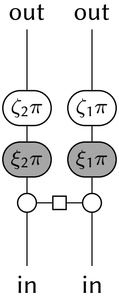
$\mathrm{C}Z \stackrel{?}{=}$



Other Universal Gates

Not-so-worked examples

$\mathrm{C}Z \stackrel{\checkmark}{=}$



Automating ZX-Calculus

```
In [4]: c = zx.Circuit(3)
        c.add_gate("TOF",0,1,2)
        g = c.to_basic_gates().to_graph()
```

```
In [6]: e = zx.editor.edit(g)
```

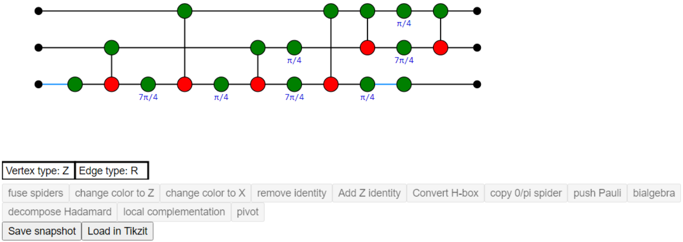


Figure: PyZX

Automating ZX-Calculus

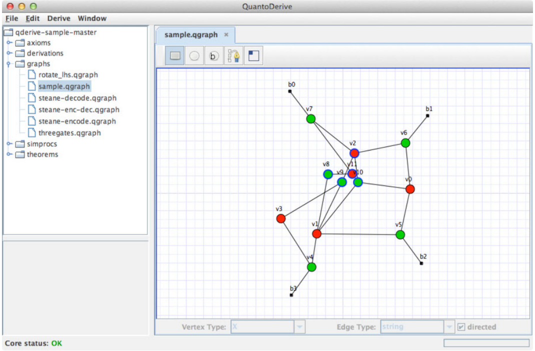


Figure: Quantomatic

References I

[1] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004.

[2] S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, pages 415–425. IEEE, 2004.

[3] M. Backens. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*, 16(9):093021, 2014.

[4] M. Backens, A. Kissinger, H. Miller-Bakewell, J. van de Wetering, and S. Wolffs. Completeness of the zh-calculus. *arXiv preprint arXiv:2103.06610*, 2021.

[5] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, 2009.

[6] B. Coecke and R. Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, 2011.

References II

[7] B. Coecke and A. Kissinger. *Picturing Quantum Processes.* Cambridge University Press, 2017.

[8] S. X. Cui, D. Gottesman, and A. Krishna. Diagonal gates in the Clifford hierarchy. *Physical Review A*, 95(1):012329, 2017.

[9] R. Duncan and S. Perdrix. Graph states and the necessity of euler decomposition. In *Conference on Computability in Europe*, pages 167–177. Springer, 2009.

[10] R. Duncan and S. Perdrix. Rewriting measurement-based quantum computations with generalised flow. In *International Colloquium on Automata, Languages, and Programming*, pages 285–296. Springer, 2010.

[11] E. Jeandel, S. Perdrix, and R. Vilmart. Diagrammatic reasoning beyond Clifford+T quantum mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 569–578, 2018.

References III

[12] A. Kissinger and J. van de Wetering. PyZX: Large scale automated diagrammatic reasoning. *arXiv e-prints*, pages arXiv–1904, 2019.

[13] A. Kissinger and J. van de Wetering. Universal MBQC with generalised parity-phase interactions and Pauli measurements. *Quantum*, 3:134, 2019.

[14] A. Kissinger and V. Zamdzhiev. Quantomatic: A proof assistant for diagrammatic reasoning. In *International Conference on Automated Deduction*, pages 326–336. Springer, 2015.

[15] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[16] R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Physical Review Letters*, 86(22):5188, 2001.

[17] R. Raussendorf, D. E. Browne, and H. J. Briegel. Measurement-based quantum computation on cluster states. *Physical review A*, 68(2):022312, 2003.