Quantum circuit optimisation, verification, and simulation with PyZX

Aleks Kissinger aleks.kissinger@cs.ox.ac.uk John van de Wetering john@vdwetering.name

Institute for Computing and Information Sciences Radboud University Nijmegen

February 1, 2020

PyZX: a Python library for manipulating large ZX-diagrams



・ロト ・聞ト ・ヨト ・ヨト

• Quantum computation is done by *quantum circuits*.

• Quantum computation is done by *quantum circuits*.

(ロ)、(型)、(E)、(E)、 E) の(の)

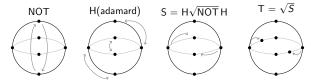
• A quantum circuit consists of *quantum gates*.

• Quantum computation is done by *quantum circuits*.

・ロト・日本・モート モー うへぐ

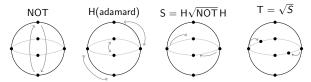
- A quantum circuit consists of *quantum gates*.
- Single qubit gates: NOT, S, T, H.

- Quantum computation is done by *quantum circuits*.
- A quantum circuit consists of *quantum gates*.
- Single qubit gates: NOT, S, T, H.



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQ@

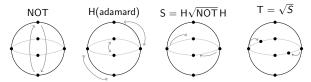
- Quantum computation is done by *quantum circuits*.
- A quantum circuit consists of *quantum gates*.
- ▶ Single qubit gates: NOT, S, T, H.



▲ロト ▲帰 ト ▲ ヨ ト ▲ ヨ ト ・ ヨ ・ の Q ()

Two qubit gate: CNOT (controlled NOT).

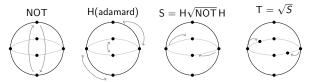
- Quantum computation is done by *quantum circuits*.
- A quantum circuit consists of *quantum gates*.
- ▶ Single qubit gates: NOT, S, T, H.



▲ロト ▲帰 ト ▲ ヨ ト ▲ ヨ ト ・ ヨ ・ の Q ()

- Two qubit gate: CNOT (controlled NOT).
- These are all the gates you need.

- Quantum computation is done by *quantum circuits*.
- A quantum circuit consists of *quantum gates*.
- Single qubit gates: NOT, S, T, H.



- Two qubit gate: CNOT (controlled NOT).
- These are all the gates you need.
- Our objective (for now) is to minimize number of gates needed

▲ロト ▲帰 ト ▲ ヨ ト ▲ ヨ ト ・ ヨ ・ の Q ()

Circuit diagrams

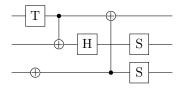




Circuit diagrams

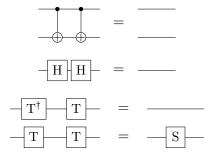


An example quantum circuit:



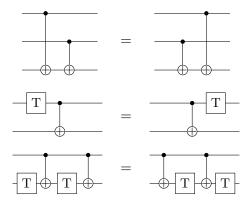
◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

Circuit identities



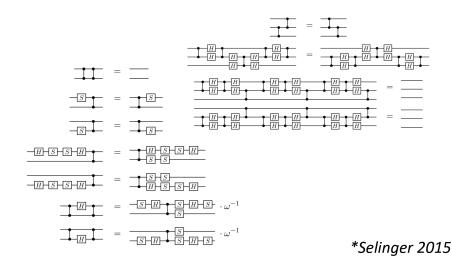
◆□▶ ◆□▶ ◆ □▶ ★ □▶ = 三 の < ⊙

Gate commutation



◆□ > ◆□ > ◆臣 > ◆臣 > ○臣 ○ のへで

More circuit equalities



- ◆ □ ▶ ◆ ■ ▶ ◆ ■ ▶ ● ■ ● ● ● ●

And more circuit equalities

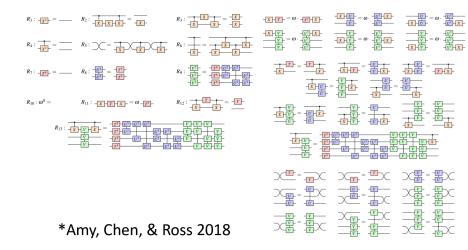
-S - A - = -A - S - $-\underline{H} - \underline{A} = -\underline{A} + \underline{X} + \underline{S} + \underline{S} + \underline{S} + \underline{S} - \cdot \omega$ $-S-A_{2} = -A_{2}-S-S-\cdot\omega$ $-H-B_{2} = -B_{1}$ $-\underline{H}_{B_4} = -\underline{B_2}$ $B_i = B_i$ $B_2 = B_3 + B_4$ $\overline{S} = B_{1} = B_{1} + \overline{S} = \overline{B_{1}} + \overline{S} = \overline{B_{2}} + \overline{B_{2}} + \overline{B_{2}} + \overline{B_{2}} = \overline{B_{2}} + \overline{B_{2}} +$ $B_{4} = B_{4} + B_{5} + B_{5}$ $= B_{3} + B$ -X - G - = -G - G-S-G- = -G-S-S-S- · w²

 $-\underline{S} - \underline{B_1} = -\underline{B_1} - \underline{H} - \underline{S} - \underline{H} - \underline{H} - \underline{S} - \underline{H} - \underline{H} - \underline{S} - \underline{H} - \underline$ $-S - B_2 = -B_3 - S - S - B - H - S$ $-\underline{S} - \underline{B_2} = -\underline{B_2} - \underline{S} - \underline{H} - \underline{H} - \underline{S} - \underline{H} - \underline{H} - \underline{S} - \underline{H} - \underline$ $-S - B_{4} = -B_{4} - B_{4} - B_{4}$ $B_1 = B_1 \overline{X}$ $B_1 = B_1 - X$ $B_4 = B_4 - X$

 $-\frac{p_1}{p_1} = -\frac{p_1}{p_1} \frac{m \cdot S \cdot m \cdot S \cdot m \cdot S}{m \cdot S \cdot m \cdot S \cdot m \cdot S} \cdot \omega^n$ *Selinger 2015

▲ロト ▲帰 ト ▲ ヨ ト ▲ ヨ ト ・ ヨ ・ の Q ()

And even more circuit equalities



▲□▶▲圖▶▲≣▶▲≣▶ ≣ のQ@

Things get messy because circuits are very rigid

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

Things get messy because circuits are very rigid

Enter ZX-diagrams

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

What gates are to circuits, *spiders* are to ZX-diagrams.

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

What gates are to circuits, *spiders* are to ZX-diagrams.

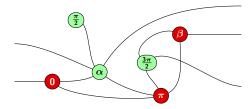


・ロト ・ 理 ト ・ ヨ ト ・ ヨ ト

What gates are to circuits, *spiders* are to ZX-diagrams.

Z-spiders: $\alpha \in [0, 2\pi]$ X-spiders: $\alpha \in [0, 2\pi]$

Spiders can be wired in any way:



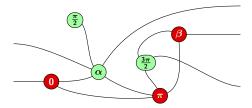
(日)、

э

What gates are to circuits, *spiders* are to ZX-diagrams.

Z-spiders: $\alpha \in [0, 2\pi]$ X-spiders: $\alpha \in [0, 2\pi]$

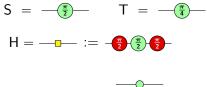
Spiders can be wired in any way:



Note: "Only connectivity matters"

Quantum gates as ZX-diagrams

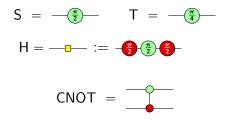
Every quantum gate can be written as a ZX-diagram:





Quantum gates as ZX-diagrams

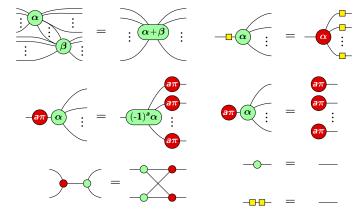
Every quantum gate can be written as a ZX-diagram:



Theorem

Any linear map between qubits can be represented as a ZX-diagram.

Rules for ZX-diagrams: The ZX-calculus



 $\alpha,\beta\in [0,2\pi]\text{, }\textbf{\textit{a}}\in\{0,1\}$

▲□▶ ▲圖▶ ▲臣▶ ▲臣▶ ―臣 … のへで

Completeness of the ZX-calculus

Theorem

If two ZX-diagrams represent the same computation, then they can be transformed into one another using the previous rules (and one additional one).

Completeness of the ZX-calculus

Theorem

If two ZX-diagrams represent the same computation, then they can be transformed into one another using the previous rules (and one additional one).

So instead of dozens of circuit equalities, we just have a few simple rules.



PyZX is an open-source Python library.

> github.com/Quantomatic/pyzx



- PyZX is an open-source Python library.
- > github.com/Quantomatic/pyzx
- Its goal is to allow easy manipulation of large ZX-diagrams.

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

PyZX

- PyZX is an open-source Python library.
- > github.com/Quantomatic/pyzx
- Its goal is to allow easy manipulation of large ZX-diagrams.

- Does circuit optimisation
- Does circuit verification
- Does circuit simulation (WIP)

Demonstration time

◆□ ▶ < 圖 ▶ < 圖 ▶ < 圖 ▶ < 圖 • 의 Q @</p>

Want to learn more?

> github.com/Quantomatic/pyzx

> zxcalculus.com

Want to learn more?

- > github.com/Quantomatic/pyzx
- zxcalculus.com

PICTURING QUANTUM PROCESSES

A First Course in Quantum Theory and Diagrammatic Reasoning

BOB COECKE AND ALEKS KISSINGER



Want to learn more?

- > github.com/Quantomatic/pyzx
- zxcalculus.com

PICTURING QUANTUM PROCESSES

A First Course in Quantum Theory and Diagrammatic Reasoning

BOB COECKE AND ALEKS KISSINGER



Thank you for your attention!

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへ⊙