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

Quantum computation

- Quantum computation is done by *quantum circuits*.

Quantum computation

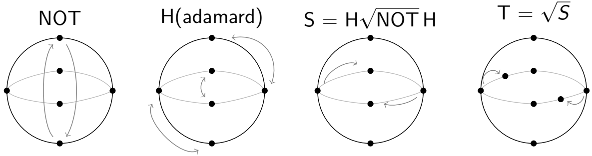
- Quantum computation is done by *quantum circuits*.
- A quantum circuit consists of *quantum gates*.

Quantum computation

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

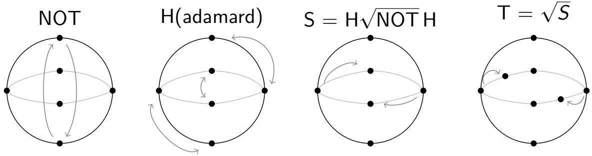
Quantum computation

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



Quantum computation

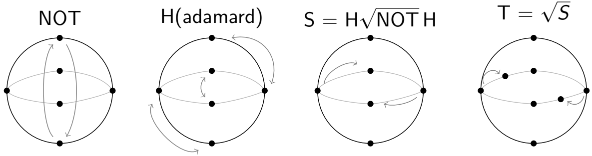
- Quantum computation is done by *quantum circuits*.
- A quantum circuit consists of *quantum gates*.
- Single qubit gates: NOT, S, T, H.



- Two qubit gate: CNOT (controlled NOT).

Quantum computation

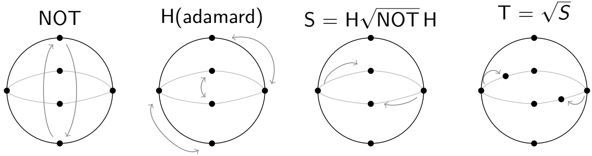
- Quantum computation is done by *quantum circuits*.
- A quantum circuit consists of *quantum gates*.
- Single qubit gates: NOT, S, T, H.



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

Quantum computation

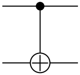
- Quantum computation is done by *quantum circuits*.
- A quantum circuit consists of *quantum gates*.
- Single qubit gates: NOT, S, T, H.



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

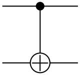
Circuit diagrams

NOT $=$ 

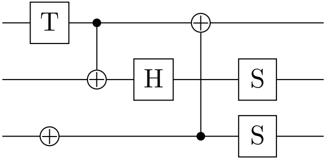
CNOT $=$ 

Circuit diagrams

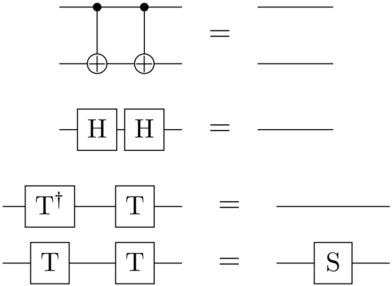
NOT $=$ 

CNOT $=$ 

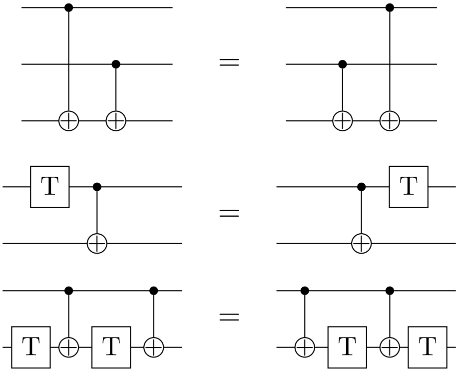
An example quantum circuit:



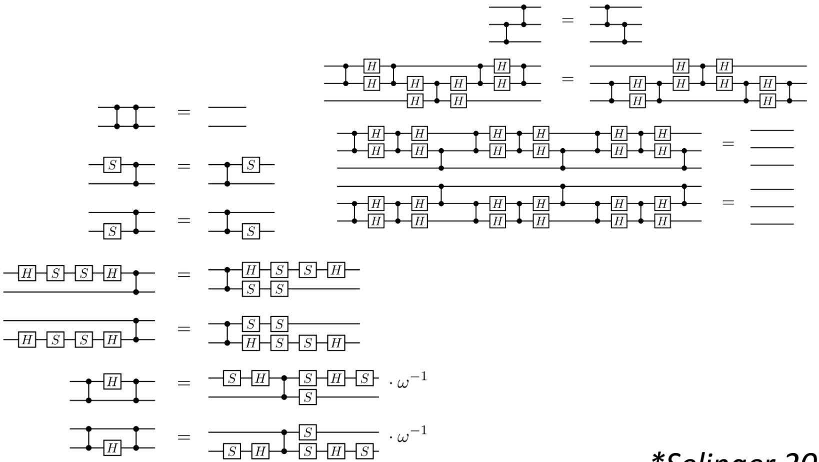
Circuit identities



Gate commutation

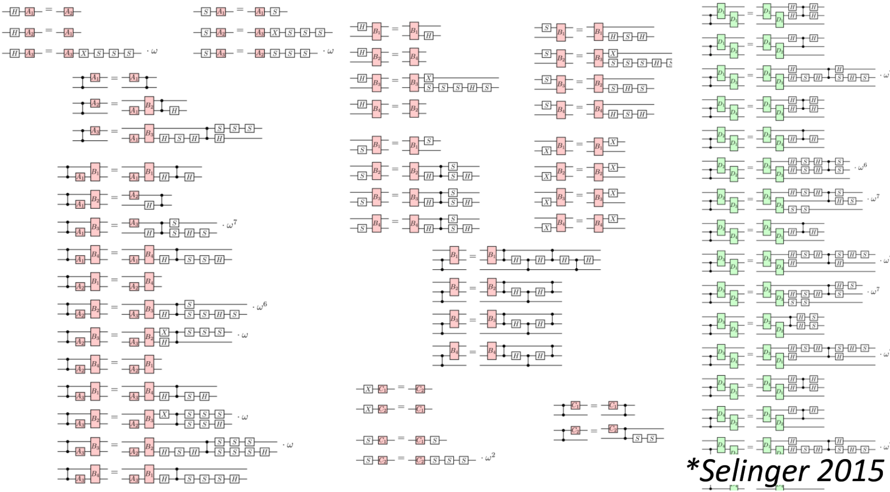


More circuit equalities



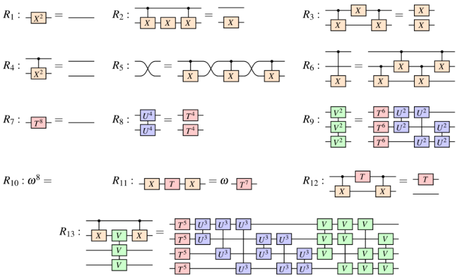
**Selinger 2015*

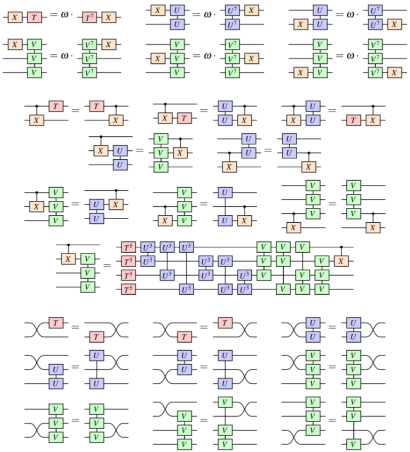
And more circuit equalities



*Selinger 2015

And even more circuit equalities





*Amy, Chen, & Ross 2018

Things get messy
because circuits are very rigid

Things get messy
because circuits are very rigid

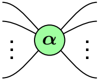
Enter ZX-diagrams

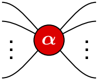
ZX-diagrams

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

ZX-diagrams

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

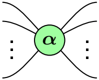
Z-spiders: 

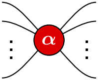
X-spiders: 

$(\alpha \in [0, 2\pi])$

ZX-diagrams

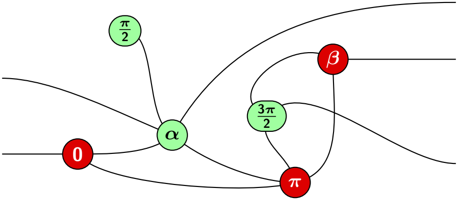
What gates are to circuits,
spiders are to ZX-diagrams.

Z-spiders: 

X-spiders: 

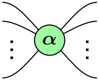
($\alpha \in [0, 2\pi]$)

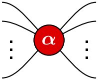
Spiders can be wired in any way:



ZX-diagrams

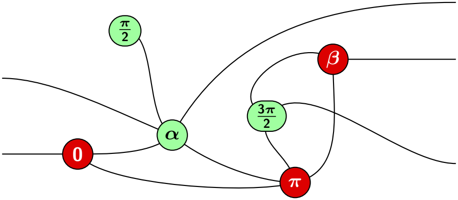
What gates are to circuits,
spiders are to ZX-diagrams.

Z-spiders: 

X-spiders: 

$(\alpha \in [0, 2\pi])$

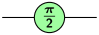
Spiders can be wired in any way:

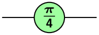


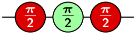
Note: “Only connectivity matters”

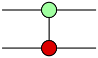
Quantum gates as ZX-diagrams

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

S = 

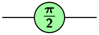
T = 

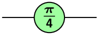
H =  := 

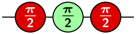
CNOT = 

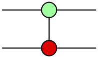
Quantum gates as ZX-diagrams

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

S = 

T = 

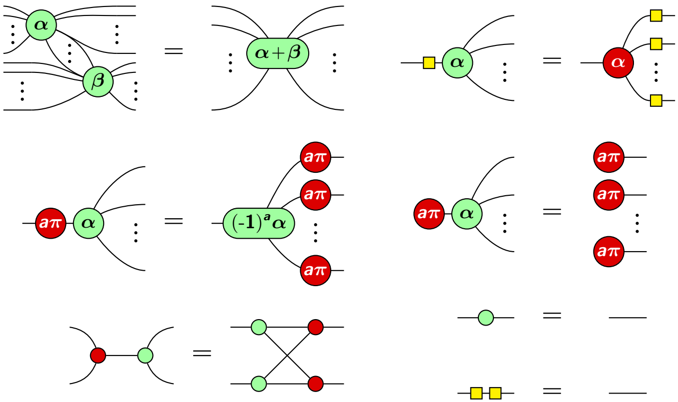
H =  := 

CNOT = 

Theorem

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

Rules for ZX-diagrams: The ZX-calculus



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

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

- PyZX is an open-source Python library.
- `github.com/Quantomatic/pyzx`

PyZX

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

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)

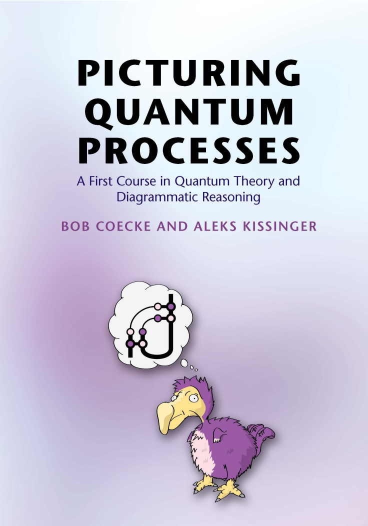
Demonstration time

Want to learn more?

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

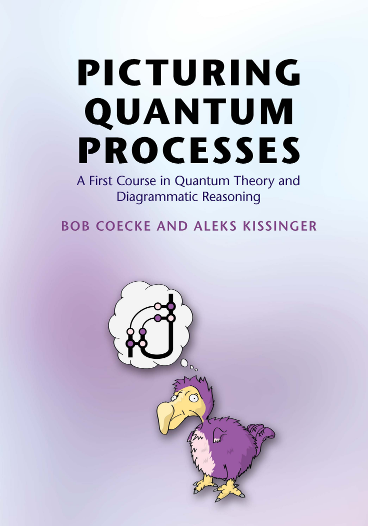
Want to learn more?

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



Want to learn more?

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



Thank you for your attention!