Understanding quantum computation using pictures

John van de Wetering
University of Amsterdam

September 25th 2024 – IvI All-Hands Meeting

Quantum Computing 101

- Stuff at small scale is not like stuff at big scale.
- Spooky action at a distance, too strong nonlocal correlations, can't copy information, can't know the full state...
- This freaked people out.

Quantum Computing 101

- Stuff at small scale is not like stuff at big scale.
- Spooky action at a distance, too strong nonlocal correlations, can't copy information, can't know the full state...
- This freaked people out.
- Then Feynmann said "not a bug, but a feature".

Quantum Computing 101

- Stuff at small scale is not like stuff at big scale.
- Spooky action at a distance, too strong nonlocal correlations, can't copy information, can't know the full state...
- This freaked people out.
- Then Feynmann said "not a bug, but a feature".
- And thus the field of quantum computing was born ($\approx$1980).

Quantum Computing 101

Classical bit: $b \in \{0, 1\}$.
Sequence of classical bits: $\vec{b} \in \{0, 1\}^n$. $\vec{b} = 00110101\ldots$
Storing n classical bits requires n classical bits of storage.

Quantum Computing 101

Classical bit: $b \in \{0, 1\}$.
Sequence of classical bits: $\vec{b} \in \{0, 1\}^n$. $\vec{b} = 00110101\ldots$
Storing n classical bits requires n classical bits of storage.

Quantum bit: $v \in \mathbb{C}^2$, $\|v\| = 1$. (usually write $|\psi\rangle$)
Sequence of quantum bits: $\mathbf{v} \in (\mathbb{C}^2)^{\otimes n} = \mathbb{C}^{2^n}$.
Storing n quantum bits requires $\approx 2^n$ classical bits of storage.

Quantum Computing 101

Classical bit: $b \in \{0, 1\}$.
Sequence of classical bits: $\vec{b} \in \{0, 1\}^n$. $\vec{b} = 00110101\ldots$
Storing n classical bits requires n classical bits of storage.

Quantum bit: $v \in \mathbb{C}^2$, $\|v\| = 1$. (usually write $|\psi\rangle$)
Sequence of quantum bits: $\mathbf{v} \in (\mathbb{C}^2)^{\otimes n} = \mathbb{C}^{2^n}$.
Storing n quantum bits requires $\approx 2^n$ classical bits of storage.

So: seems reasonable we should be able to do faster calculations (or at least store more data) using quantum systems as opposed to classical systems.

Quantum Computing 101

Classical bit: $b \in \{0, 1\}$.
Sequence of classical bits: $\vec{b} \in \{0, 1\}^n$. $\vec{b} = 00110101\ldots$
Storing n classical bits requires n classical bits of storage.

Quantum bit: $v \in \mathbb{C}^2$, $\|v\| = 1$. (usually write $|\psi\rangle$)
Sequence of quantum bits: $\mathbf{v} \in (\mathbb{C}^2)^{\otimes n} = \mathbb{C}^{2^n}$.
Storing n quantum bits requires $\approx 2^n$ classical bits of storage.

So: seems reasonable we should be able to do faster calculations (or at least store more data) using quantum systems as opposed to classical systems.

But it is harder than you might think.

The obvious thing to try

Suppose we have a classical function $f : \{0,1\}^n \to \{0,1\}^m$.
Classically: evaluate it *one* input at a time.

The obvious thing to try

Suppose we have a classical function $f : \{0,1\}^n \to \{0,1\}^m$.
Classically: evaluate it *one* input at a time.

Quantumly: implement a *uniform superposition* over all inputs (this is actually not that hard):

$$|\phi\rangle = \frac{1}{2^{n/2}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle$$

The obvious thing to try

Suppose we have a classical function $f : \{0,1\}^n \to \{0,1\}^m$.
Classically: evaluate it *one* input at a time.

Quantumly: implement a *uniform superposition* over all inputs (this is actually not that hard):

$$|\phi\rangle = \frac{1}{2^{n/2}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle$$

Applying f to this state we get *all the outcomes at the same time*:

$$f(|\phi\rangle) = \frac{1}{2^{n/2}} \sum_{\mathbf{x} \in \{0,1\}^n} |f(\mathbf{x})\rangle$$

The obvious thing to try

Suppose we have a classical function $f : \{0,1\}^n \rightarrow \{0,1\}^m$.
Classically: evaluate it *one* input at a time.

Quantumly: implement a *uniform superposition* over all inputs (this is actually not that hard):

$$|\phi\rangle = \frac{1}{2^{n/2}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle$$

Applying f to this state we get *all the outcomes at the same time*:

$$f(|\phi\rangle) = \frac{1}{2^{n/2}} \sum_{\mathbf{x} \in \{0,1\}^n} |f(\mathbf{x})\rangle$$

Amazing! But then why haven't we solved world hunger yet?

The big problem of quantum computing

While physics is quantum,
our minds are classical

The big problem of quantum computing

While physics is quantum,
our minds are classical

To get the quantum information into our brains
we need to *measure* it.

Measuring destroys superpositions.

The big problem of quantum computing

While physics is quantum,
our minds are classical

To get the quantum information into our brains
we need to *measure* it.

Measuring destroys superpositions.

Our amazing trick fails :'(

(unless we use other really smart tricks)

Quantum gates as rotations

- Quantum computation is done by *quantum circuits*.

Quantum gates as rotations

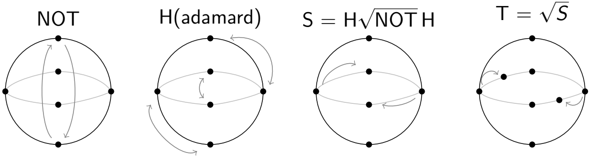
- Quantum computation is done by *quantum circuits*.
- A quantum circuit consists of *quantum gates*.

Quantum gates as rotations

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

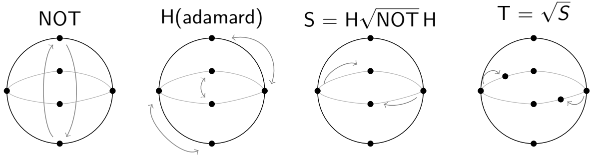
Quantum gates as rotations

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



Quantum gates as rotations

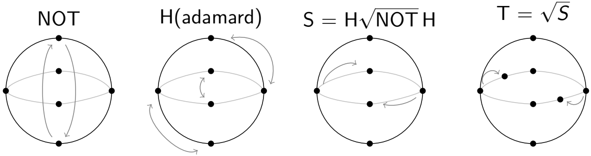
- Quantum computation is done by *quantum circuits*.
- A quantum circuit consists of *quantum gates*.
- Single qubit gates: NOT, S, T, H.



- Two qubit gate: CNOT (controlled NOT): $|x, y\rangle \mapsto |x, x \oplus y\rangle$.

Quantum gates as rotations

- 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): $|x, y\rangle \mapsto |x, x \oplus y\rangle$.
- This is an *approximately universal* gate set.

Quantum gates as matrices

Note: quantum gates are just matrices...

Quantum gates as matrices

Note: quantum gates are just matrices...

$$\mathsf{S} \;=\; \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \qquad \mathsf{H} \;=\; \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\mathsf{CNOT} \;=\; \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Quantum gates as matrices

Note: quantum gates are just matrices...

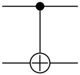
$$\mathsf{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \qquad \mathsf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

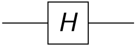
$$\mathsf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

... but nobody wants to deal with these things directly.

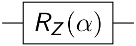
Quantum gates as gates

$$X \;=\; \text{NOT} \;=\;$$  $$=\; \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\text{CNOT} \;=\;$$  $$=\; \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\text{Hadamard} \;=\;$$ 

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

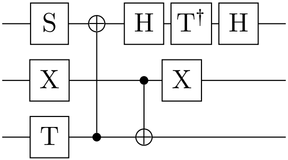
$$R_Z(\alpha) \;=\;$$ 

$$=\; \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$$

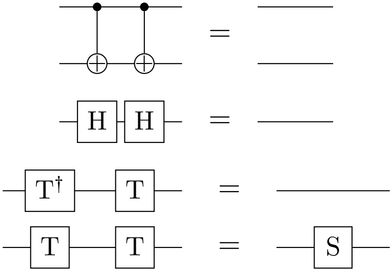
$$Z := R_Z(\pi) \qquad S := R_Z(\tfrac{\pi}{2}) \qquad T := R_Z(\tfrac{\pi}{4})$$

Quantum circuits

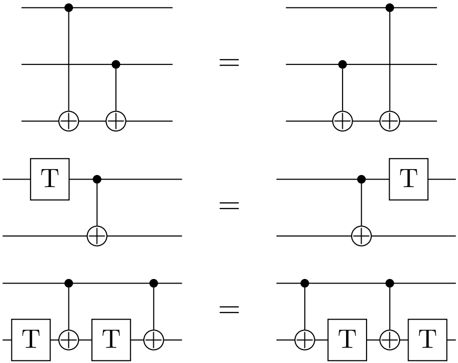
An example quantum circuit:



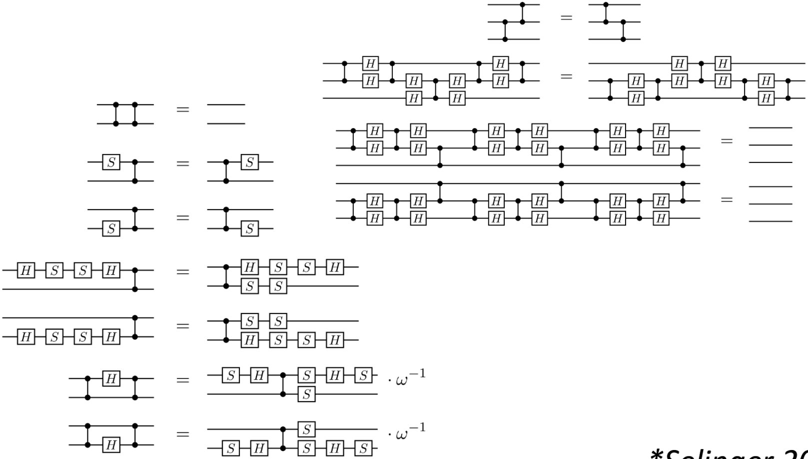
Circuit identities



Gate commutation

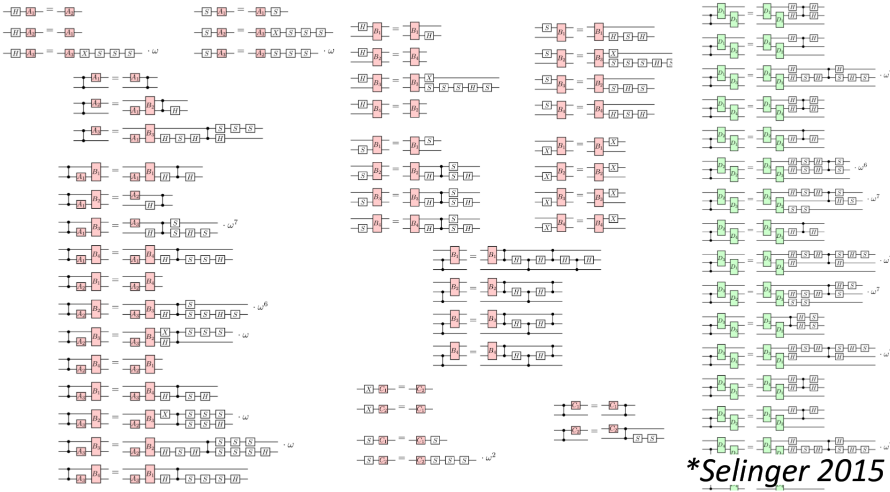


More circuit equalities



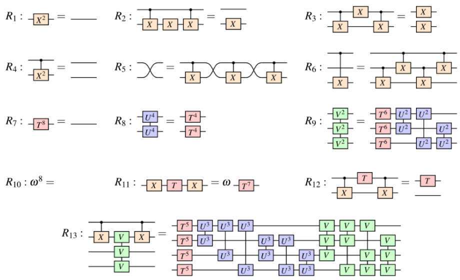
**Selinger 2015*

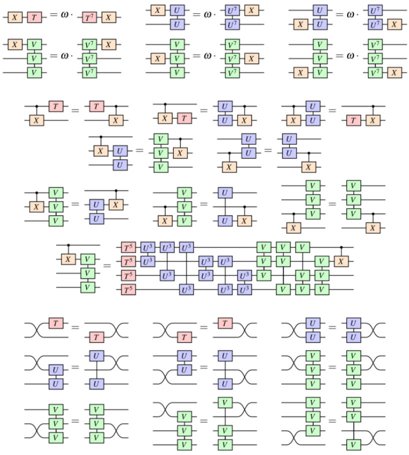
And more circuit equalities



*Selinger 2015

And even more circuit equalities





*Amy, Chen, & Ross 2018

Quantum circuits bad!

Why is this so terrible?

- Choice of gates is a bit arbitrary
- The notation is not “quantum native”
- Wires are rigid going from left-to-right

Quantum circuits bad!

Why is this so terrible?

- Choice of gates is a bit arbitrary
- The notation is not “quantum native”
- Wires are rigid going from left-to-right

The *ZX-calculus* essentially gets rid of these problems

The ZX-calculus

The ZX-calculus is a graphical language for quantum computation

The ZX-calculus

The ZX-calculus is a graphical language for quantum computation

Used in:

- Quantum circuit compilation and simulation
- Measurement-based quantum computation
- Surface codes and lattice surgery
- ...

The ZX-calculus

The ZX-calculus is a graphical language for quantum computation

Used in:

- Quantum circuit compilation and simulation
- Measurement-based quantum computation
- Surface codes and lattice surgery
- ...

It is also a convenient tool for day-to-day quantum reasoning

Spiders

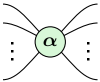
What gates are to circuits, *spiders* are to ZX-diagrams.

Spiders

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

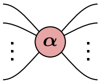
Z-spider

$$|0\cdots0\rangle\langle0\cdots0| + e^{i\alpha}\,|1\cdots1\rangle\langle1\cdots1|$$



X-spider

$$|+\cdots+\rangle\langle+\cdots+| + e^{i\alpha}\,|-\cdots-\rangle\langle-\cdots-|$$



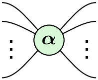
where $|\pm\rangle := |0\rangle \pm |1\rangle$

Spiders

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

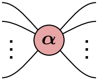
Z-spider

$$|0\cdots0\rangle\langle0\cdots0| + e^{i\alpha}\,|1\cdots1\rangle\langle1\cdots1|$$



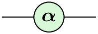
X-spider

$$|+\cdots+\rangle\langle+\cdots+| + e^{i\alpha}\,|-\cdots-\rangle\langle-\cdots-|$$



where $|\pm\rangle := |0\rangle \pm |1\rangle$

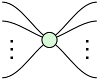
For example:



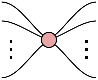
$$= |0\rangle\langle0| + e^{i\alpha}\,|1\rangle\langle1| = \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} + \begin{pmatrix}0 & 0\\ 0 & e^{i\alpha}\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & e^{i\alpha}\end{pmatrix}$$

Spiders cont.

If $\alpha = 0$ we drop the label:



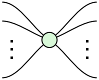
$$= \quad |0 \cdots 0\rangle\langle 0 \cdots 0| + |1 \cdots 1\rangle\langle 1 \cdots 1|$$



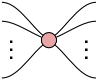
$$= \quad |+ \cdots +\rangle\langle + \cdots +| + |- \cdots -\rangle\langle - \cdots -|$$

Spiders cont.

If $\alpha = 0$ we drop the label:

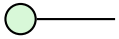


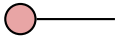
$$= \quad |0\cdots 0\rangle\langle 0\cdots 0| + |1\cdots 1\rangle\langle 1\cdots 1|$$

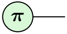


$$= \quad |+\cdots +\rangle\langle +\cdots +| + |-\cdots -\rangle\langle -\cdots -|$$

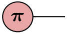
Example:

 $= \quad |0\rangle + |1\rangle \; = \sqrt{2}\,|+\rangle$

 $= \quad |+\rangle + |-\rangle \; = \sqrt{2}\,|0\rangle$



$= \quad |0\rangle - |1\rangle \; = \sqrt{2}\,|-\rangle$



$= \quad |+\rangle - |-\rangle \; = \sqrt{2}\,|1\rangle$

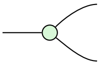
We will ignore these $\sqrt{2}$ scalar factors

Formal composition

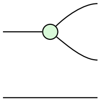
Spiders can be composed in two ways.

Formal composition

Spiders can be composed in two ways.
Vertical composition gives tensor product:



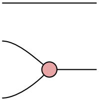
$$= \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$



$$= \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Formal composition

Other tensor product:

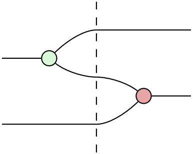


$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Formal composition

Horizontal composition is regular composition of linear maps:

 $=$

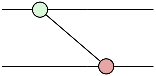
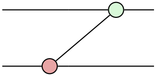
$$\frac{1}{\sqrt{2}}\begin{pmatrix}1&0&0&1&0&0&0&0\\0&1&1&0&0&0&0&0\\0&0&0&0&1&0&0&1\\0&0&0&0&0&1&1&0\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$

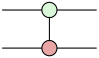
Building ZX-diagrams

Any ZX-diagram is built by simply iterating
these vertical and horizontal compositions

Symmetries

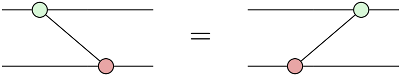
Note:

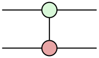
 $=$ 

Hence, we may write 

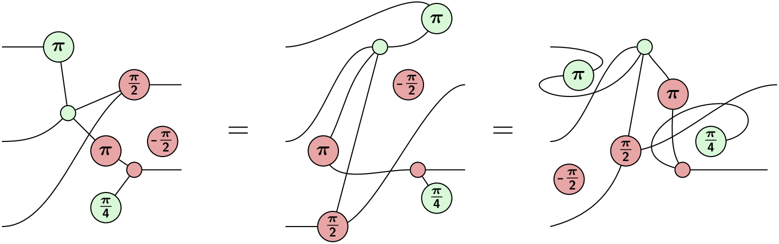
Symmetries

Note:



Hence, we may write 

In general: *only connectivity matters*



ZX-diagrams summary

- Two types of generators: Z-spiders and X-spiders
- Can compose both horizontally and vertically
- Wires can connect every which way

ZX-diagrams summary

- Two types of generators: Z-spiders and X-spiders
- Can compose both horizontally and vertically
- Wires can connect every which way

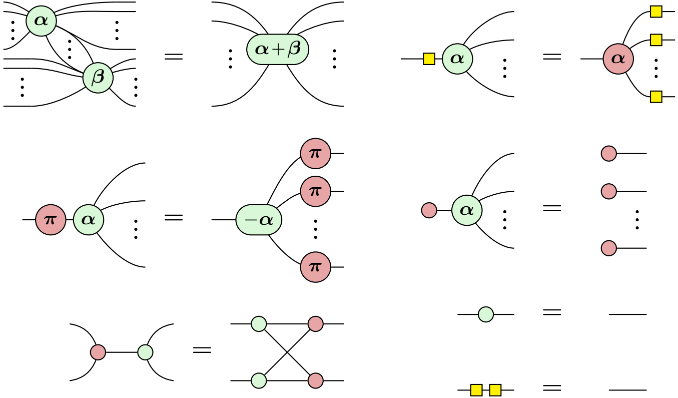
How powerful are ZX-diagrams as a representation?

Theorem

ZX-diagrams are *universal*: any linear map between qubits can be represented as a ZX-diagram.

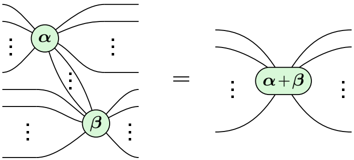
So far it's just notation. What can we do with it?

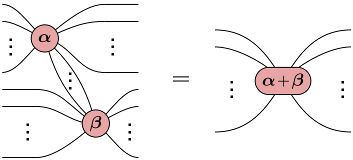
Rules for ZX-diagrams: The ZX-calculus



$\forall \alpha, \beta \in [0, 2\pi]$

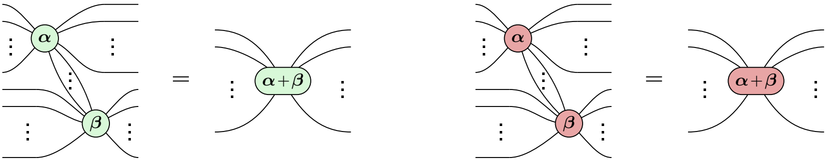
Spider fusion



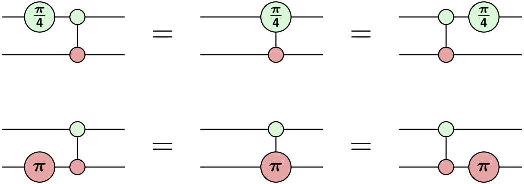


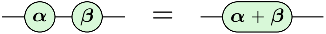
Connected spiders of same colour fuse

Spider fusion

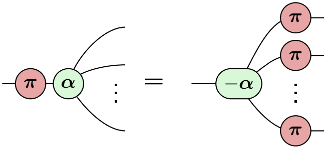


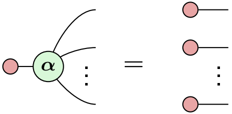
Connected spiders of same colour fuse





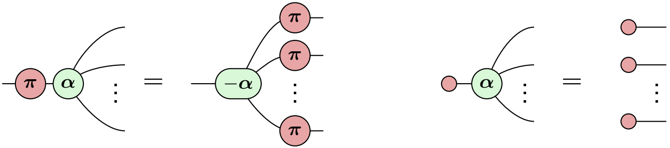
State and pi-copy





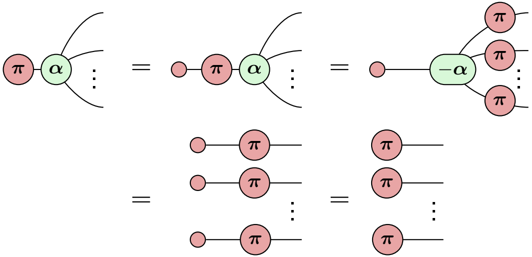
π's and phase-free states copy through the other colour

State and pi-copy



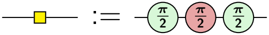
π's and phase-free states copy through the other colour

Combining rules:



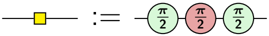
Hadamards and colour-changing

Definition of Hadamard in ZX:

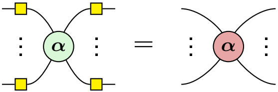


Hadamards and colour-changing

Definition of Hadamard in ZX:

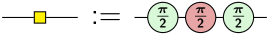


Rules:

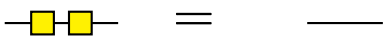


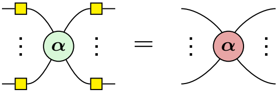
Hadamards and colour-changing

Definition of Hadamard in ZX:

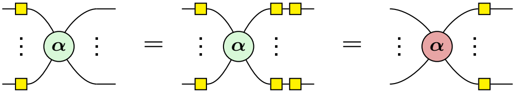


Rules:



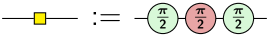


Derived rule: *commuting Hadamards changes colour*

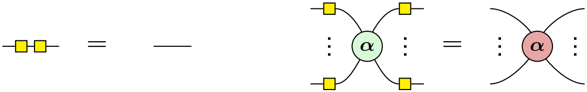


Hadamards and colour-changing

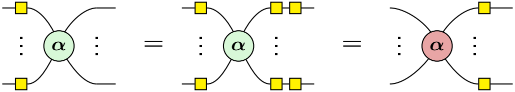
Definition of Hadamard in ZX:



Rules:



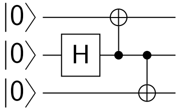
Derived rule: *commuting Hadamards changes colour*



Consequence: *Everything in ZX holds with colours reversed*

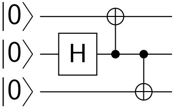
Example 1: GHZ-preparation circuit

GHZ-state is $|000\rangle + |111\rangle$.
The following circuit creates a GHZ-state:

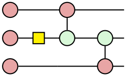


Example 1: GHZ-preparation circuit

GHZ-state is $|000\rangle + |111\rangle$.
The following circuit creates a GHZ-state:

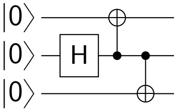


Proof:

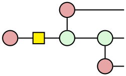


Example 1: GHZ-preparation circuit

GHZ-state is $|000\rangle + |111\rangle$.
The following circuit creates a GHZ-state:

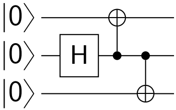


Proof:

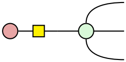


Example 1: GHZ-preparation circuit

GHZ-state is $|000\rangle + |111\rangle$.
The following circuit creates a GHZ-state:

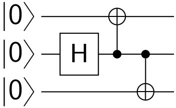


Proof:

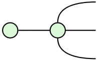


Example 1: GHZ-preparation circuit

GHZ-state is $|000\rangle + |111\rangle$.
The following circuit creates a GHZ-state:

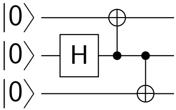


Proof:

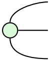


Example 1: GHZ-preparation circuit

GHZ-state is $|000\rangle + |111\rangle$.
The following circuit creates a GHZ-state:

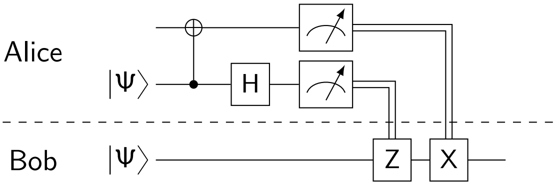


Proof:



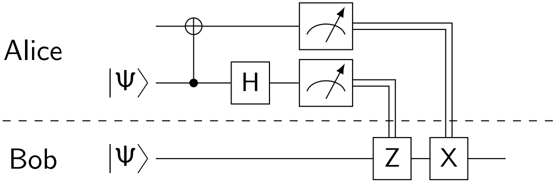
Example 2: Teleportation

Let $|\Psi\rangle$ represent a side of a Bell state.
Then this is the standard quantum teleportation protocol:

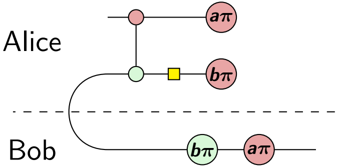


Example 2: Teleportation

Let $|\Psi\rangle$ represent a side of a Bell state.
Then this is the standard quantum teleportation protocol:

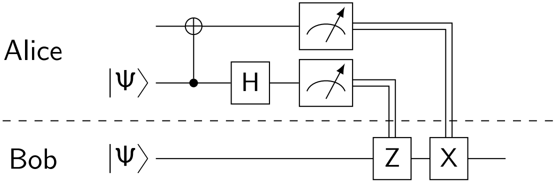


Proof:

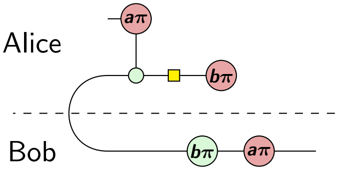


Example 2: Teleportation

Let $|\Psi\rangle$ represent a side of a Bell state.
Then this is the standard quantum teleportation protocol:

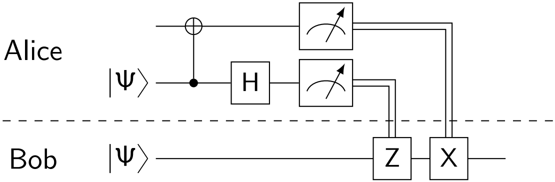


Proof:

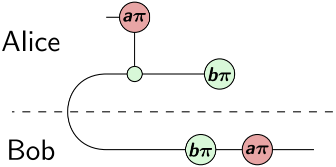


Example 2: Teleportation

Let $|\Psi\rangle$ represent a side of a Bell state.
Then this is the standard quantum teleportation protocol:

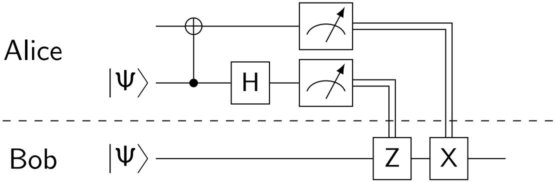


Proof:

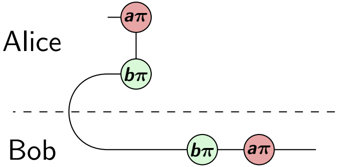


Example 2: Teleportation

Let $|\Psi\rangle$ represent a side of a Bell state.
Then this is the standard quantum teleportation protocol:

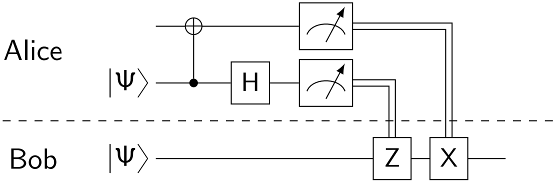


Proof:

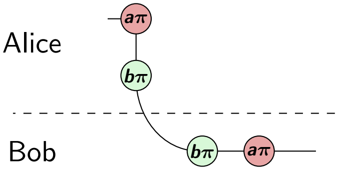


Example 2: Teleportation

Let $|\Psi\rangle$ represent a side of a Bell state.
Then this is the standard quantum teleportation protocol:

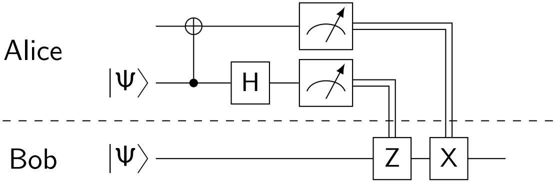


Proof:

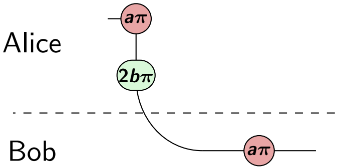


Example 2: Teleportation

Let $|\Psi\rangle$ represent a side of a Bell state.
Then this is the standard quantum teleportation protocol:

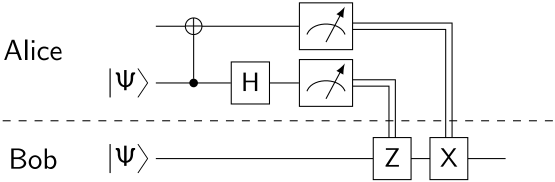


Proof:

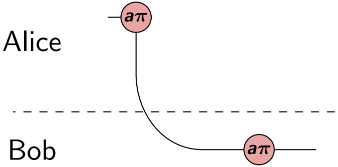


Example 2: Teleportation

Let $|\Psi\rangle$ represent a side of a Bell state.
Then this is the standard quantum teleportation protocol:

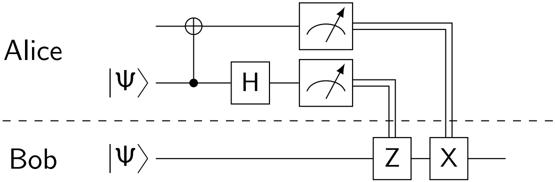


Proof:

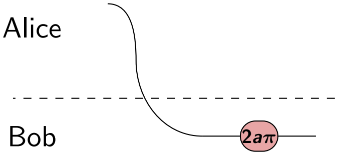


Example 2: Teleportation

Let $|\Psi\rangle$ represent a side of a Bell state.
Then this is the standard quantum teleportation protocol:

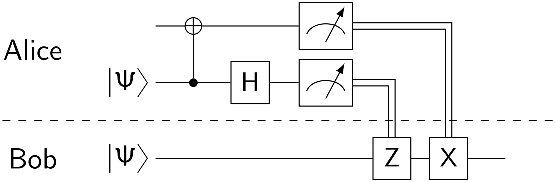


Proof:

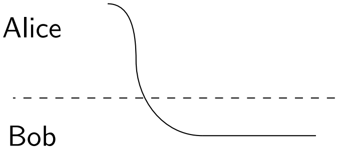


Example 2: Teleportation

Let $|\Psi\rangle$ represent a side of a Bell state.
Then this is the standard quantum teleportation protocol:

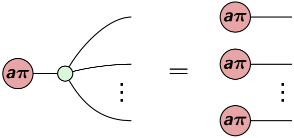


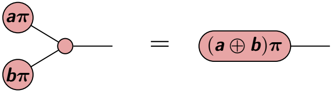
Proof:



Bialgebra

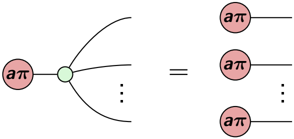
Z-spiders act like COPY; X-spiders act like XOR:

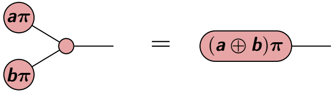




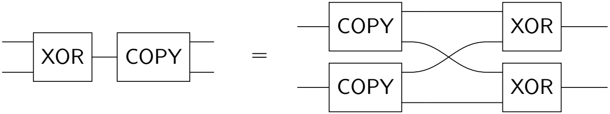
Bialgebra

Z-spiders act like COPY; X-spiders act like XOR:



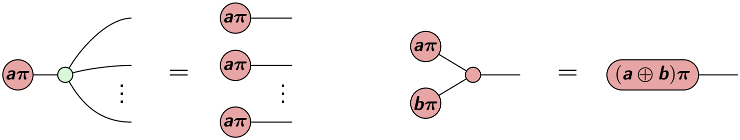


Classically we have:

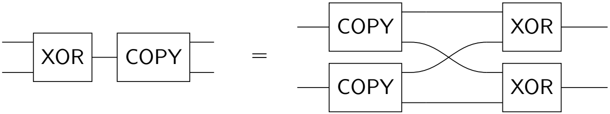


Bialgebra

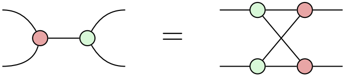
Z-spiders act like COPY; X-spiders act like XOR:



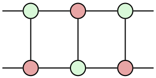
Classically we have:



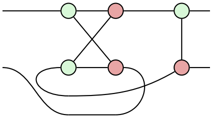
Hence:



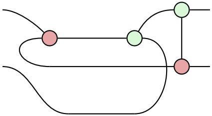
Example 3: Three CNOTs make SWAP



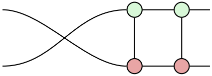
Example 3: Three CNOTs make SWAP



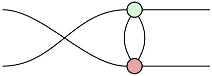
Example 3: Three CNOTs make SWAP



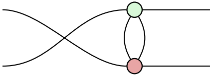
Example 3: Three CNOTs make SWAP

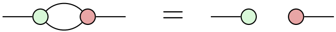


Example 3: Three CNOTs make SWAP

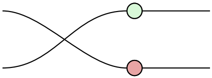


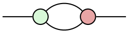
Example 3: Three CNOTs make SWAP





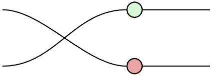
Example 3: Three CNOTs make SWAP



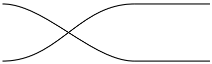


$=$

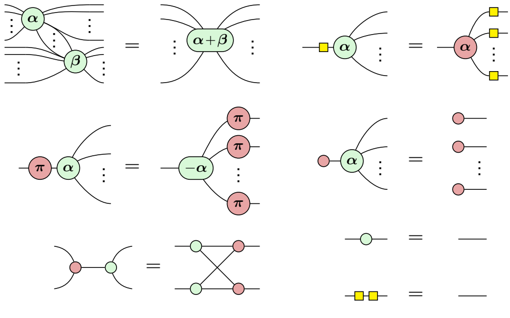
Example 3: Three CNOTs make SWAP



Example 3: Three CNOTs make SWAP



Rules for ZX-diagrams: The ZX-calculus

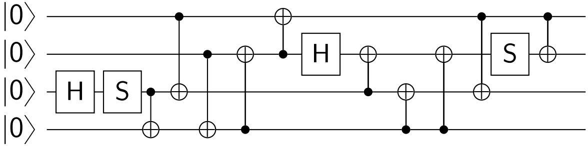


$$\alpha, \beta \in [0, 2\pi]$$

- All derivations hold in any orientation
- All derivations hold with colours interchanged
- All derivations hold with phases negated

Example 4: Detecting entanglement

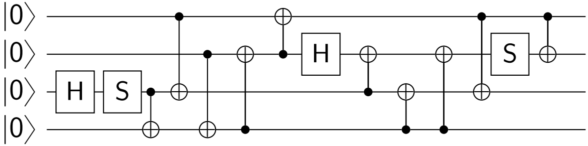
Consider the following circuit:



Q: Which qubits are entangled in the end?

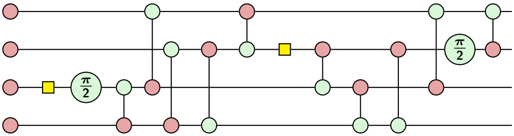
Example 4: Detecting entanglement

Consider the following circuit:



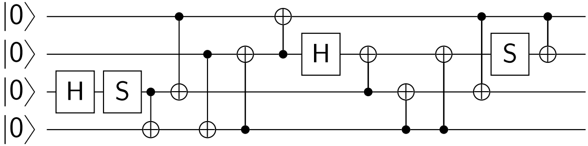
Q: Which qubits are entangled in the end?

Step 1: Write it as a ZX-diagram



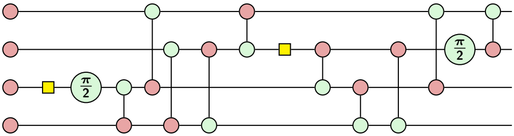
Example 4: Detecting entanglement

Consider the following circuit:



Q: Which qubits are entangled in the end?

Step 1: Write it as a ZX-diagram



Step 2: Open up PyZX

Completeness

How much can we prove using the rules?

Theorem

The rules shown so far suffice to show any true equality between *Clifford* diagrams (where phases are $\frac{\pi}{2}$).

Completeness

How much can we prove using the rules?

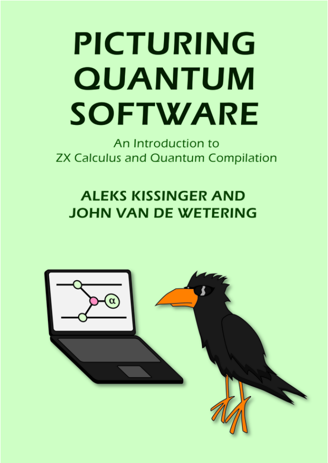
Theorem

The rules shown so far suffice to show any true equality between *Clifford* diagrams (where phases are $\frac{\pi}{2}$).

Theorem

These rules + one more suffice to show *any* true equality.

New book on quantum compilation



- Over 500 pages and 100 exercises
- Synthesis of quantum circuits
- Optimisation, verification, simulation
- A new approach to understanding quantum error correction
- And all this using ZX-diagrams!
- And available for free for everyone!

https://github.com/zxcalc/book

Classical simulation using ZX

Veni proposal: Combine tensor network techniques with *stabiliser decomposition* approach.

- Write computation as ZX-diagram.
- Optimise ZX-diagram.
- Replace ‘expensive’ resource states by sum of ‘cheap’ states.
- Further optimise each term in the sum.
- Repeat until a number falls out.

Classical simulation using ZX

Veni proposal: Combine tensor network techniques with *stabiliser decomposition* approach.

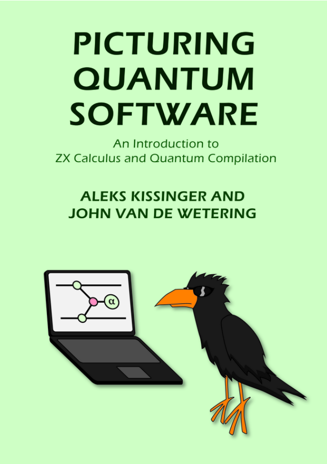
- Write computation as ZX-diagram.
- Optimise ZX-diagram.
- Replace 'expensive' resource states by sum of 'cheap' states.
- Further optimise each term in the sum.
- Repeat until a number falls out.

Apply this to:

- Simulate quantum computations
- Calculate properties of condensed matter systems
- SAT/model counting problems

Thank you for your attention!

Kissinger & vdW 2021, arXiv:2109.01076
Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions



https://github.com/zxcalc/book