Simulation of quantum circuits by ZX-diagram contraction

John van de Wetering

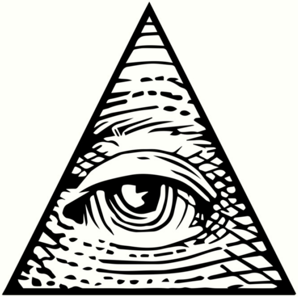
john@vdwetering.name

Institute for Computing and Information Sciences
Radboud University Nijmegen

September 6, 2019

Holy Trinity of quantum circuits

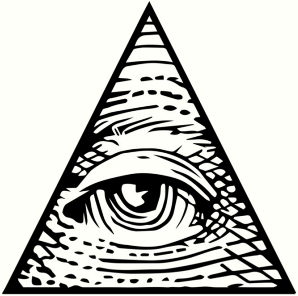
Optimization



Verification

Holy Trinity of quantum circuits

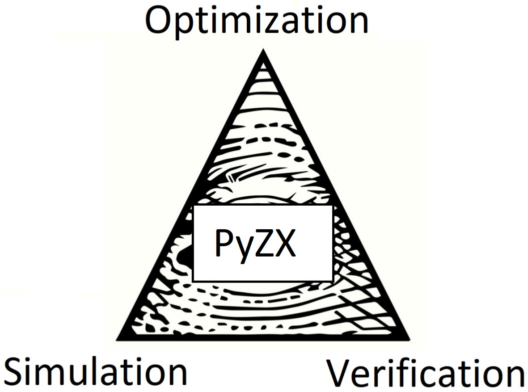
Optimization



Simulation

Verification

Holy Trinity of quantum circuits



Quantum circuit simulation

The Problem: Given quantum circuit C, and input state $|\psi\rangle$, answer *some* question about $C|\psi\rangle$.

Quantum circuit simulation

The Problem: Given quantum circuit C, and input state $|\psi\rangle$, answer *some* question about $C|\psi\rangle$.
E.g. Find the probability $|\langle 0\cdots 0|C|\psi\rangle|^2$.

Quantum circuit simulation

The Problem: Given quantum circuit C, and input state $|\psi\rangle$, answer *some* question about $C|\psi\rangle$.
E.g. Find the probability $|\langle 0 \cdots 0| C |\psi\rangle|^2$.

Why do we care?

- Verification of correctness of circuits.
- Modelling physical systems.
- Understand when *quantum supremacy* has been reached.

Weak versus Strong simulation

Suppose we measure all qubits in computational basis at the end of the circuit. Then we get a probability distribution

$$P(x_1 \cdots x_n) = |\langle x_1 \cdots x_n| C |\psi\rangle|^2$$

Weak versus Strong simulation

Suppose we measure all qubits in computational basis at the end of the circuit. Then we get a probability distribution

$$P(x_1 \cdots x_n) = |\langle x_1 \cdots x_n| C |\psi\rangle|^2$$

Weak simulation: sample from this distribution.
This is **BQP**-complete.

Weak versus Strong simulation

Suppose we measure all qubits in computational basis at the end of the circuit. Then we get a probability distribution

$$P(x_1 \cdots x_n) = |\langle x_1 \cdots x_n| C |\psi\rangle|^2$$

Weak simulation: sample from this distribution.
This is **BQP**-complete.

Strong simulation: get any marginal probability of $P(x_1 \cdots x_n)$.
This is **#P**-hard.

Direct simulation

Write n-qubit state as 2^n complex numbers.
Every gate in the circuit modifies this vector.

Direct simulation

Write n-qubit state as 2^n complex numbers.
Every gate in the circuit modifies this vector.

- Naively, for $n = 50$ would take $\approx 1000\,TB$.

Direct simulation

Write n-qubit state as 2^n complex numbers.
Every gate in the circuit modifies this vector.

- Naively, for $n = 50$ would take $\approx 1000\,TB$.
- But $n \approx 80$ has been achieved in practice by being smart. (exploiting sparsity, limited depth of circuit, etc.)

Direct simulation

Write n-qubit state as 2^n complex numbers.
Every gate in the circuit modifies this vector.

- Naively, for $n = 50$ would take $\approx 1000\,TB$.
- But $n \approx 80$ has been achieved in practice by being smart. (exploiting sparsity, limited depth of circuit, etc.)

All these methods in a sense rely on *tensor contraction*.
They are all exponential in number of qubits.

Stabilizer decompositions 1

Gottesman-Knill Theorem

Any Clifford computation can be efficiently simulated.

Q: Can we exploit this somehow to simulate arbitrary circuits?

Stabilizer decompositions 1

Gottesman-Knill Theorem

Any Clifford computation can be efficiently simulated.

Q: Can we exploit this somehow to simulate arbitrary circuits?

Observation: Clifford states linearly span the set of all states.

Stabilizer decompositions 1

Gottesman-Knill Theorem

Any Clifford computation can be efficiently simulated.

Q: Can we exploit this somehow to simulate arbitrary circuits?

Observation: Clifford states linearly span the set of all states.

1. Gadgetize T-gates to write any circuit as:
   Clifford circuit C & $|T\rangle$ magic state ancillae.

Stabilizer decompositions 1

Gottesman-Knill Theorem

Any Clifford computation can be efficiently simulated.

Q: Can we exploit this somehow to simulate arbitrary circuits?

Observation: Clifford states linearly span the set of all states.

1. Gadgetize T-gates to write any circuit as:
   Clifford circuit C & $|T\rangle$ magic state ancillae.
2. Write input state + ancillae as linear combination of Cliffords:
   $|\psi\rangle = \sum_i^n \lambda_i |\phi_i\rangle$.

Stabilizer decompositions 1

Gottesman-Knill Theorem

Any Clifford computation can be efficiently simulated.

Q: Can we exploit this somehow to simulate arbitrary circuits?

Observation: Clifford states linearly span the set of all states.

1. Gadgetize T-gates to write any circuit as:
   Clifford circuit C & $|T\rangle$ magic state ancillae.
2. Write input state + ancillae as linear combination of Cliffords:
   $|\psi\rangle = \sum_i^n \lambda_i |\phi_i\rangle$.
3. Note: Each $C|\phi_i\rangle$ can be efficiently simulated!

Stabilizer decompositions 2

Given Clifford circuit C and input $|\psi\rangle = \sum_i^n \lambda_i |\phi_i\rangle$ where the $|\phi_i\rangle$ are Clifford. How do we approximate $C|\psi\rangle$?

Stabilizer decompositions 2

Given Clifford circuit C and input $|\psi\rangle = \sum_i^n \lambda_i |\phi_i\rangle$ where the $|\phi_i\rangle$ are Clifford. How do we approximate $C|\psi\rangle$?

Two methods:

1. Monte-Carlo over the $|\phi_i\rangle$ weighted by $|\lambda_i|$.
   Polynomial in the *negativity*: $\lambda = \sum_i^n |\lambda_i|$.

Stabilizer decompositions 2

Given Clifford circuit C and input $|\psi\rangle = \sum_i^n \lambda_i |\phi_i\rangle$ where the $|\phi_i\rangle$ are Clifford. How do we approximate $C|\psi\rangle$?

Two methods:

1. Monte-Carlo over the $|\phi_i\rangle$ weighted by $|\lambda_i|$.
   Polynomial in the *negativity*: $\lambda = \sum_i^n |\lambda_i|$.
2. Compute $\sum_i^n \lambda_i C |\phi_i\rangle$.
   Polynomial in the *stabilizer rank*: $\mathrm{R}(|\psi\rangle) = n$.

Stabilizer decompositions 2

Given Clifford circuit C and input $|\psi\rangle = \sum_i^n \lambda_i |\phi_i\rangle$ where the $|\phi_i\rangle$ are Clifford. How do we approximate $C|\psi\rangle$?

Two methods:

1. Monte-Carlo over the $|\phi_i\rangle$ weighted by $|\lambda_i|$.
   Polynomial in the *negativity*: $\lambda = \sum_i^n |\lambda_i|$.
2. Compute $\sum_i^n \lambda_i C |\phi_i\rangle$.
   Polynomial in the *stabilizer rank*: $\mathrm{R}(|\psi\rangle) = n$.

Benefit of first: can deal with density matrices and noise.
Benefit of second: better constants and thus scaling.

Stabilizer decompositions 2

Given Clifford circuit C and input $|\psi\rangle = \sum_i^n \lambda_i |\phi_i\rangle$ where the $|\phi_i\rangle$ are Clifford. How do we approximate $C|\psi\rangle$?

Two methods:

1. Monte-Carlo over the $|\phi_i\rangle$ weighted by $|\lambda_i|$.
   Polynomial in the *negativity*: $\lambda = \sum_i^n |\lambda_i|$.
2. Compute $\sum_i^n \lambda_i C |\phi_i\rangle$.
   Polynomial in the *stabilizer rank*: $\mathrm{R}(|\psi\rangle) = n$.

Benefit of first: can deal with density matrices and noise.
Benefit of second: better constants and thus scaling.

We will only use the second approach.

Stabilizer rank

T-magic state $|T\rangle := |0\rangle + e^{i\pi/4}|1\rangle$ has rank $R(|T\rangle) = 2$.

Stabilizer rank

T-magic state $|T\rangle := |0\rangle + e^{i\pi/4}|1\rangle$ has rank $R(|T\rangle) = 2$.
Hence:

$$R(|T\rangle^{\otimes n}) \leqslant 2^n$$

e.g. $|T\rangle \otimes |T\rangle = |00\rangle + e^{i\pi/4}|01\rangle + e^{i\pi/4}|10\rangle + e^{i\pi/2}|11\rangle$

Stabilizer rank

T-magic state $|T\rangle := |0\rangle + e^{i\pi/4}|1\rangle$ has rank $R(|T\rangle) = 2$.
Hence:

$$R(|T\rangle^{\otimes n}) \leqslant 2^n$$

e.g. $|T\rangle \otimes |T\rangle = |00\rangle + e^{i\pi/4}|01\rangle + e^{i\pi/4}|10\rangle + e^{i\pi/2}|11\rangle$

But also: $|T\rangle \otimes |T\rangle = (|00\rangle + i|11\rangle) + e^{i\pi/4}(|01\rangle + |10\rangle)$,
so actually $R(|T\rangle^{\otimes 2}) = 2$,

Stabilizer rank

T-magic state $|T\rangle := |0\rangle + e^{i\pi/4}|1\rangle$ has rank $R(|T\rangle) = 2$.
Hence:

$$R(|T\rangle^{\otimes n}) \leqslant 2^n$$

e.g. $|T\rangle \otimes |T\rangle = |00\rangle + e^{i\pi/4}|01\rangle + e^{i\pi/4}|10\rangle + e^{i\pi/2}|11\rangle$

But also: $|T\rangle \otimes |T\rangle = (|00\rangle + i|11\rangle) + e^{i\pi/4}(|01\rangle + |10\rangle)$,
so actually $R(|T\rangle^{\otimes 2}) = 2$, and hence:

$$R(|T\rangle^{\otimes n}) = R((|T\rangle \otimes |T\rangle)^{n/2}) \leqslant 2^{n/2}$$

Stabilizer rank

T-magic state $|T\rangle := |0\rangle + e^{i\pi/4}|1\rangle$ has rank $R(|T\rangle) = 2$.
Hence:

$$R(|T\rangle^{\otimes n}) \leqslant 2^n$$

e.g. $|T\rangle \otimes |T\rangle = |00\rangle + e^{i\pi/4}|01\rangle + e^{i\pi/4}|10\rangle + e^{i\pi/2}|11\rangle$

But also: $|T\rangle \otimes |T\rangle = (|00\rangle + i|11\rangle) + e^{i\pi/4}(|01\rangle + |10\rangle)$,
so actually $R(|T\rangle^{\otimes 2}) = 2$, and hence:

$$R(|T\rangle^{\otimes n}) = R((|T\rangle \otimes |T\rangle)^{n/2}) \leqslant 2^{n/2}$$

Can also show that $R(|T\rangle^{\otimes 6}) = 7$, and hence:

$$R(|T\rangle^{\otimes n}) \leqslant 2^{\alpha n} \qquad \text{where } \alpha = \log_2(7)/6 \approx 0.468$$

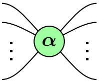
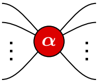
The goal:
combine tensor contraction
& stabilizer decompositions
using the ZX-calculus.

ZX-diagrams

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

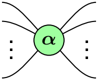
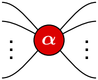
ZX-diagrams

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

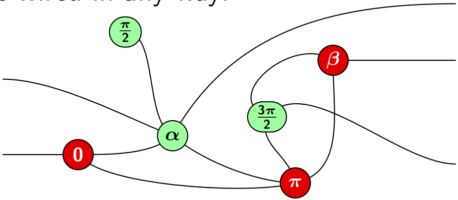
Z-spider	X-spider
$\vert 0 \cdots 0\rangle\langle 0 \cdots 0\vert$	$\vert + \cdots +\rangle\langle + \cdots +\vert$
$+ e^{i\alpha} \vert 1 \cdots 1\rangle\langle 1 \cdots 1\vert$	$+ e^{i\alpha} \vert - \cdots -\rangle\langle - \cdots -\vert$
	

ZX-diagrams

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

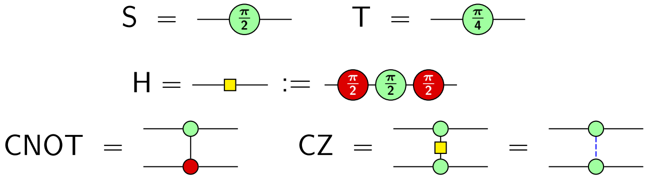
Z-spider	X-spider
$\lvert 0\cdots 0\rangle\langle 0\cdots 0\rvert$	$\lvert +\cdots +\rangle\langle +\cdots +\rvert$
$+e^{i\alpha}\lvert 1\cdots 1\rangle\langle 1\cdots 1\rvert$	$+\ e^{i\alpha}\lvert -\cdots -\rangle\langle -\cdots -\rvert$
	

Spiders can be wired in any way:



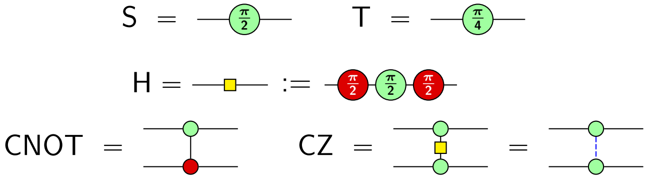
Quantum gates as ZX-diagrams

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



Quantum gates as ZX-diagrams

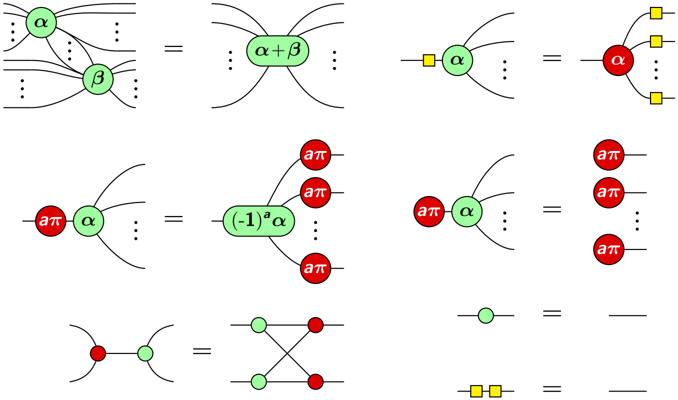
Every quantum gate can be written as a ZX-diagram:



Universality

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

ZX-diagrams representing same linear map,
can be transformed into one another
using previous rules (and some additional ones).

Circuit simulation with ZX-calculus

1. Write circuit+state as ZX-diagram.
2. Simplify using ZX-calculus rules.

Circuit simulation with ZX-calculus

1. Write circuit+state as ZX-diagram.
2. Simplify using ZX-calculus rules.
3. *Replace magic states by stabilizer decomposition.*

Circuit simulation with ZX-calculus

1. Write circuit+state as ZX-diagram.
2. Simplify using ZX-calculus rules.
3. *Replace magic states by stabilizer decomposition.*
4. Repeat.
5. ...
6. Profit!

Simplifying ZX-diagrams

Same as in previous talk
(local complementation, pivoting, gadgetization)

Simplifying ZX-diagrams

Same as in previous talk
(local complementation, pivoting, gadgetization)
But:

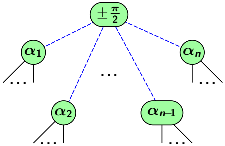
- All rewrites now need to be scalar accurate.

Simplifying ZX-diagrams

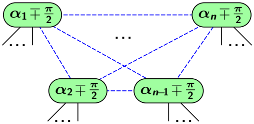
Same as in previous talk
(local complementation, pivoting, gadgetization)
But:

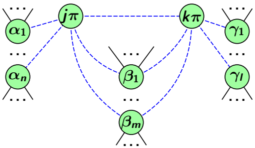
- All rewrites now need to be scalar accurate.
- We no longer care about circuit extraction, so we can do more stuff!

Scalar-accurate local complementation and pivot

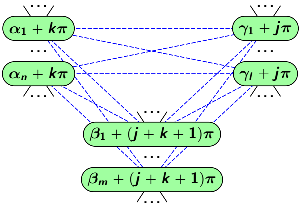


$$= e^{\pm i\pi/4}\sqrt{2}^{\frac{(n-1)(n-2)}{2}}$$

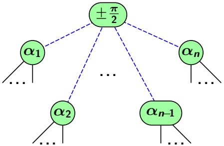




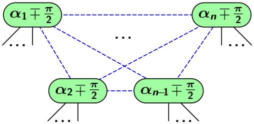
$$= \begin{array}{c} (-1)^{ab}\sqrt{2}^{(n-1)m} \\ \sqrt{2}^{(l-1)m} \\ \sqrt{2}^{(n-1)(l-1)} \end{array}$$

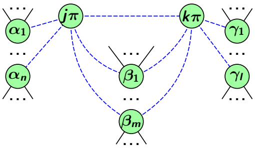


Scalar-accurate local complementation and pivot

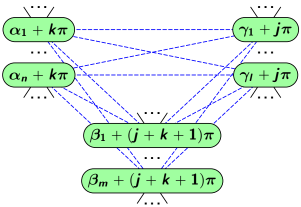


$$= e^{\pm i\pi/4}\sqrt{2}^{\frac{(n-1)(n-2)}{2}}$$





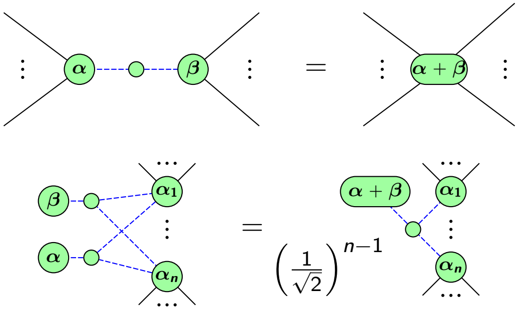
$$= (-1)^{ab}\sqrt{2}^{(n-1)m} \quad \sqrt{2}^{(l-1)m} \quad \sqrt{2}^{(n-1)(l-1)}$$



These + variations kill all internal Clifford spiders.

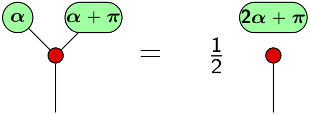
Further optimization

From previous talk:

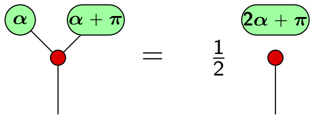


New rule: Supplementarity

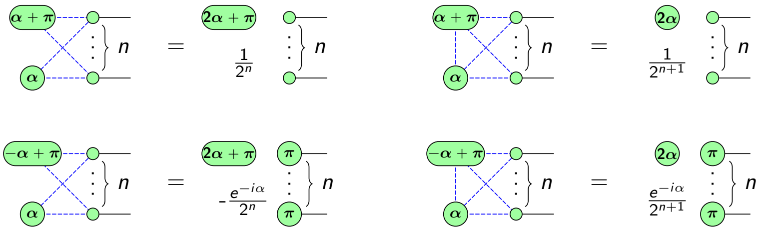
Rule used in ZX for completeness:



New rule: Supplementarity

Rule used in ZX for completeness: 

Can be generalised to following four cases:



Example

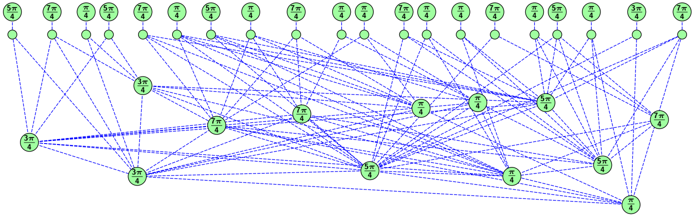
Consider benchmark circuit `hwb6`: 7 qubits, and 105 T-gates.
After PyZX simplification: 75 T-gates.

Example

Consider benchmark circuit `hwb6`: 7 qubits, and 105 T-gates.
After PyZX simplification: 75 T-gates.
Inputting the state $|{+}{+}{-}{-}{-}{+}{-}\rangle$ and effect $\langle{+}011-1{-}|$,

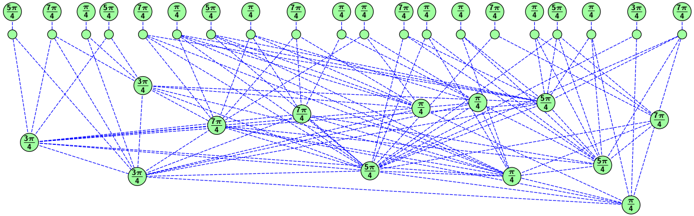
Example

Consider benchmark circuit `hwb6`: 7 qubits, and 105 T-gates.
After PyZX simplification: 75 T-gates.
Inputting the state $|++---+-\rangle$ and effect $\langle +011-1-|$,
and further simplifying gives (up to scalar):

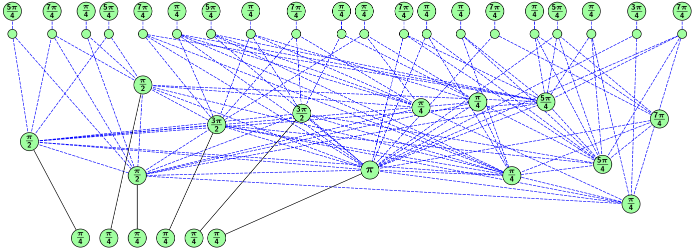


This has 33 T-gates.

Example

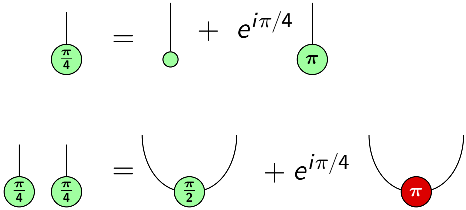


$=$



Now we should apply the stabilizer decomposition to these states.

Stabilizer decompositions in ZX



But what about the 6 T-gate rank 7 decomposition?

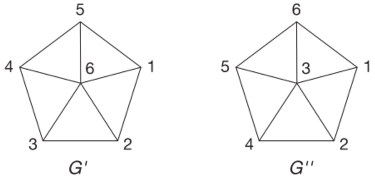


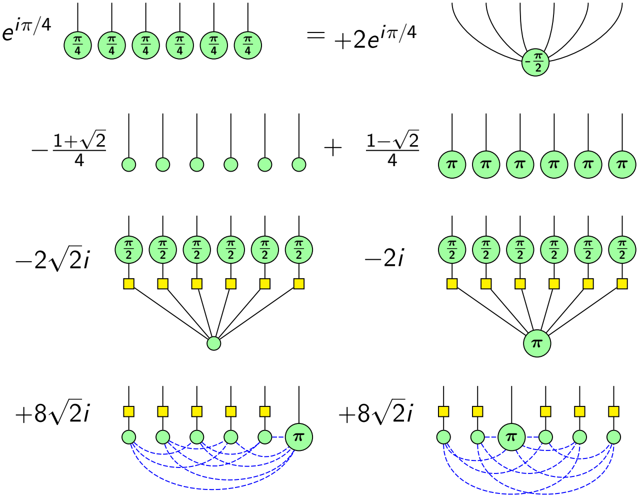
FIG. 3. Graphs G' and G'' used in the definition of stabilizer states ϕ' and ϕ''; see Eq. (11).

$$
\begin{aligned}
|H^{\otimes 6}\rangle = {} & (-16 + 12\sqrt{2})|B_{6,0}\rangle + (96 - 68\sqrt{2})|B_{6,6}\rangle \\
& + (10 - 7\sqrt{2})|E_6\rangle + (-14 + 10\sqrt{2})|O_6\rangle \\
& + (7 - 5\sqrt{2})Z^{\otimes 6}|K_6\rangle + (10 - 7\sqrt{2})|\phi'\rangle \\
& + (10 - 7\sqrt{2})|\phi''\rangle,
\end{aligned} \tag{11}
$$

where

$$
|\phi'\rangle = \prod_{(i,j)\in E'} \Lambda(Z)_{i,j}|O_6\rangle \quad \text{and} \quad |\phi''\rangle = \prod_{(i,j)\in E''} \Lambda(Z)_{i,j}|O_6\rangle.
$$

Source: Sergey Bravyi, Graeme Smith, and John A Smolin.
Trading classical and quantum computational resources (2016).


$e^{i\pi/4}$
$\frac{\pi}{4}$ $\frac{\pi}{4}$ $\frac{\pi}{4}$ $\frac{\pi}{4}$ $\frac{\pi}{4}$ $\frac{\pi}{4}$
$= +2e^{i\pi/4}$
$-\frac{\pi}{2}$
$-\frac{1+\sqrt{2}}{4}$
$+\ \frac{1-\sqrt{2}}{4}$
π π π π π π
$-2\sqrt{2}i$
$\frac{\pi}{2}$ $\frac{\pi}{2}$ $\frac{\pi}{2}$ $\frac{\pi}{2}$ $\frac{\pi}{2}$ $\frac{\pi}{2}$
$-2i$
$\frac{\pi}{2}$ $\frac{\pi}{2}$ $\frac{\pi}{2}$ $\frac{\pi}{2}$ $\frac{\pi}{2}$ $\frac{\pi}{2}$
π
$+8\sqrt{2}i$
π
$+8\sqrt{2}i$
π

Demo time

Conclusions

- With ZX-calculus we can combine tensor contraction with stabilizer decomposition.

Conclusions

- With ZX-calculus we can combine tensor contraction with stabilizer decomposition.
- With rewriting we can further reduce amount of non-Cliffords in each sub-diagram.

Conclusions

- With ZX-calculus we can combine tensor contraction with stabilizer decomposition.
- With rewriting we can further reduce amount of non-Cliffords in each sub-diagram.
- Even removing just 1 extra spider in every diagram would allow $\approx 15\%$ bigger circuits.

Future work

- Investigate which groups of spiders should be replaced.
- Find right trade-off in using more computation early on.

Future work

- Investigate which groups of spiders should be replaced.
- Find right trade-off in using more computation early on.
- Approximate decompositions and pruning of small branches.

Future work

- Investigate which groups of spiders should be replaced.
- Find right trade-off in using more computation early on.
- Approximate decompositions and pruning of small branches.
- Make high-performance implementation of the algorithm.

Future work

- Investigate which groups of spiders should be replaced.
- Find right trade-off in using more computation early on.
- Approximate decompositions and pruning of small branches.
- Make high-performance implementation of the algorithm.
- Marginal probabilities possible with CPM construction. Is there a better way?

Thank you for your attention!

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