Combinatorial Invariants and Quantum Circuits

(With speculation on the status of "quantum supremacy")

Kenneth W. Regan[1]
University at Buffalo (SUNY)

28 Sept., 2024

[1]Joint work with Amlan Chakrabarti, University of Calcutta, and Chaowen Guan, University of Cincinnati

Status of Universal Quantum Computing

Status of Universal Quantum Computing

- Is represented by **BQP**, which includes the Factoring problem.

Status of Universal Quantum Computing

- Is represented by **BQP**, which includes the Factoring problem.
- Factoring is believed outside the class of **P**: problems deemed solvable on classical computers

Status of Universal Quantum Computing

- Is represented by **BQP**, which includes the Factoring problem.
- Factoring is believed outside the class of **P**: problems deemed solvable on classical computers—or **BPP** if we add randomness.

Status of Universal Quantum Computing

- Is represented by **BQP**, which includes the Factoring problem.
- Factoring is believed outside the class of **P**: problems deemed solvable on classical computers—or **BPP** if we add randomness.
- If the **NP**-complete **SAT** problem requires exponential size **circuits**, then **BPP** = **P** anyway.

Status of Universal Quantum Computing

- Is represented by **BQP**, which includes the Factoring problem.
- Factoring is believed outside the class of **P**: problems deemed solvable on classical computers—or **BPP** if we add randomness.
- If the **NP**-complete **SAT** problem requires exponential size **circuits**, then **BPP** $=$ **P** anyway.
- Neither Factoring nor **BQP** seem to reach **NP**-complete level.

Status of Universal Quantum Computing

- Is represented by **BQP**, which includes the Factoring problem.
- Factoring is believed outside the class of **P**: problems deemed solvable on classical computers—or **BPP** if we add randomness.
- If the **NP**-complete **SAT** problem requires exponential size **circuits**, then **BPP** = **P** anyway.
- Neither Factoring nor **BQP** seem to reach **NP**-complete level.
- **BQP** $\subseteq$ **#P**, which is the analogue of **NP** for *counting problems*.

Status of Universal Quantum Computing

- Is represented by **BQP**, which includes the Factoring problem.
- Factoring is believed outside the class of **P**: problems deemed solvable on classical computers—or **BPP** if we add randomness.
- If the **NP**-complete **SAT** problem requires exponential size **circuits**, then **BPP** = **P** anyway.
- Neither Factoring nor **BQP** seem to reach **NP**-complete level.
- **BQP** $\subseteq$ **#P**, which is the analogue of **NP** for *counting problems*.
- E.g., **#SAT** asks "how many solutions?", not "is there a solution?"

Status of Universal Quantum Computing

- Is represented by **BQP**, which includes the Factoring problem.
- Factoring is believed outside the class of **P**: problems deemed solvable on classical computers—or **BPP** if we add randomness.
- If the **NP**-complete **SAT** problem requires exponential size **circuits**, then **BPP** = **P** anyway.
- Neither Factoring nor **BQP** seem to reach **NP**-complete level.
- **BQP** $\subseteq$ **#P**, which is the analogue of **NP** for *counting problems*.
- E.g., **#SAT** asks "how many solutions?", not "is there a solution?"
- There has still not been a *clear* instance of factoring an integer larger than $21 = 3 \times 7$ via **Shor's Algorithm** on a universal QC.

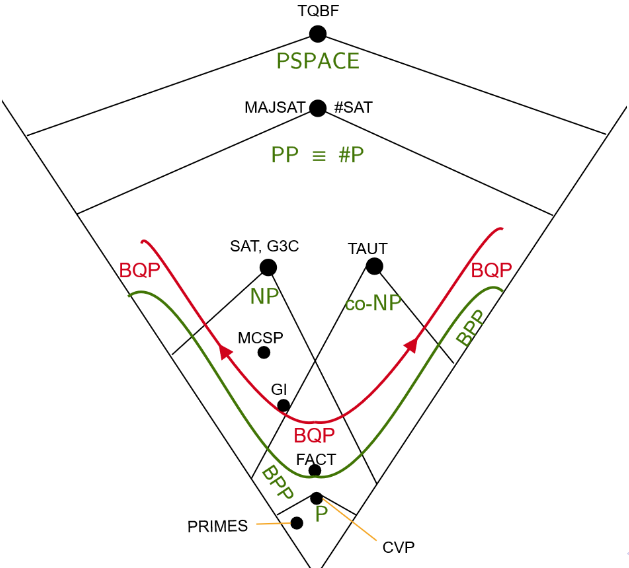
Status of Universal Quantum Computing

- Is represented by **BQP**, which includes the Factoring problem.
- Factoring is believed outside the class of **P**: problems deemed solvable on classical computers—or **BPP** if we add randomness.
- If the **NP**-complete **SAT** problem requires exponential size **circuits**, then **BPP** $=$ **P** anyway.
- Neither Factoring nor **BQP** seem to reach **NP**-complete level.
- **BQP** $\subseteq$ **#P**, which is the analogue of **NP** for *counting problems*.
- E.g., **#SAT** asks "how many solutions?", not "is there a solution?"
- There has still not been a *clear* instance of factoring an integer larger than $21 = 3 \times 7$ via **Shor's Algorithm** on a universal QC.
- **Adiabatic** quantum computing is theoretically universal but its computations are ephemeral.

Status of Universal Quantum Computing

- Is represented by **BQP**, which includes the Factoring problem.
- Factoring is believed outside the class of **P**: problems deemed solvable on classical computers—or **BPP** if we add randomness.
- If the **NP**-complete **SAT** problem requires exponential size **circuits**, then **BPP = P** anyway.
- Neither Factoring nor **BQP** seem to reach **NP**-complete level.
- **BQP** $\subseteq$ **#P**, which is the analogue of **NP** for *counting problems*.
- E.g., **#SAT** asks "how many solutions?", not "is there a solution?"
- There has still not been a *clear* instance of factoring an integer larger than $21 = 3 \times 7$ via **Shor's Algorithm** on a universal QC.
- **Adiabatic** quantum computing is theoretically universal but its computations are ephemeral. Also has stability issues in practice.

The Complexity Class Neighborhood...



Structural Forebodings

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.
- Similar between **P** and **#P**, per "Dichotomy" results by Jin-Yi Cai and others.

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.
- Similar between **P** and **#P**, per "Dichotomy" results by Jin-Yi Cai and others.
- Except that **BQP** is in the latter desert.

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.
- Similar between **P** and **#P**, per "Dichotomy" results by Jin-Yi Cai and others.
- Except that **BQP** is in the latter desert. Is **BQP** squeezed out?

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.
- Similar between **P** and **#P**, per "Dichotomy" results by Jin-Yi Cai and others.
- Except that **BQP** is in the latter desert. Is **BQP** squeezed out?
- Not many exponential-saving quantum algorithms besides Shor's.

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.
- Similar between **P** and **#P**, per "Dichotomy" results by Jin-Yi Cai and others.
- Except that **BQP** is in the latter desert. Is **BQP** squeezed out?
- Not many exponential-saving quantum algorithms besides Shor's.
- **Grover's Algorithm** is only quadratic savings, and for **SAT** and **#SAT**, saves only $\sqrt{\exp(n)} = \exp(n/2)$.

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.
- Similar between **P** and **#P**, per "Dichotomy" results by Jin-Yi Cai and others.
- Except that **BQP** is in the latter desert. Is **BQP** squeezed out?
- Not many exponential-saving quantum algorithms besides Shor's.
- **Grover's Algorithm** is only quadratic savings, and for **SAT** and **#SAT**, saves only $\sqrt{\exp(n)} = \exp(n/2)$.
- "Quantum supremacy" knocked down? Shor's algorithm dinged, or is it improved? A major app de-quantized?

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.
- Similar between **P** and **#P**, per "Dichotomy" results by Jin-Yi Cai and others.
- Except that **BQP** is in the latter desert. Is **BQP** squeezed out?
- Not many exponential-saving quantum algorithms besides Shor's.
- **Grover's Algorithm** is only quadratic savings, and for **SAT** and **#SAT**, saves only $\sqrt{\exp(n)} = \exp(n/2)$.
- "Quantum supremacy" knocked down? Shor's algorithm dinged, or is it improved? A major app de-quantized?
- Many **NP**-complete problems have adept heuristics.

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.
- Similar between **P** and **#P**, per "Dichotomy" results by Jin-Yi Cai and others.
- Except that **BQP** is in the latter desert. Is **BQP** squeezed out?
- Not many exponential-saving quantum algorithms besides Shor's.
- **Grover's Algorithm** is only quadratic savings, and for **SAT** and **#SAT**, saves only $\sqrt{\exp(n)} = \exp(n/2)$.
- "Quantum supremacy" knocked down? Shor's algorithm dinged, or is it improved? A major app de-quantized?
- Many **NP**-complete problems have adept heuristics.
- Also for **#SAT**: software sharpSAT, Cachet.

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.
- Similar between **P** and **#P**, per "Dichotomy" results by Jin-Yi Cai and others.
- Except that **BQP** is in the latter desert. Is **BQP** squeezed out?
- Not many exponential-saving quantum algorithms besides Shor's.
- **Grover's Algorithm** is only quadratic savings, and for **SAT** and **#SAT**, saves only $\sqrt{\exp(n)} = \exp(n/2)$.
- "Quantum supremacy" knocked down? Shor's algorithm dinged, or is it improved? A major app de-quantized?
- Many **NP**-complete problems have adept heuristics.
- Also for **#SAT**: software sharpSAT, Cachet.
- However, **SAT**-encoded cases of Factoring remain hard for them.

Structural Forebodings

- Between **P** and **NP**-complete is mostly deserted.
- Similar between **P** and **#P**, per "Dichotomy" results by Jin-Yi Cai and others.
- Except that **BQP** is in the latter desert. Is **BQP** squeezed out?
- Not many exponential-saving quantum algorithms besides Shor's.
- **Grover's Algorithm** is only quadratic savings, and for **SAT** and **#SAT**, saves only $\sqrt{\exp(n)} = \exp(n/2)$.
- "Quantum supremacy" knocked down? Shor's algorithm dinged, or is it improved? A major app de-quantized?
- Many **NP**-complete problems have adept heuristics.
- Also for **#SAT**: software sharpSAT, Cachet.
- However, **SAT**-encoded cases of Factoring remain hard for them.

Can we capture **quantum circuits** by combinatorial invariants that lead to new heuristics for *classically* simulating them?

Dichotomy Example Over $\mathbb{Z}_4$

Consider *quadratic* polynomials $f(x_1, x_2, \ldots, x_n)$ **modulo 4**.

Dichotomy Example Over $\mathbb{Z}_4$

Consider *quadratic* polynomials $f(x_1, x_2, \ldots, x_n)$ **modulo 4**.

- Counting the number of zeroes is in P.

Dichotomy Example Over $\mathbb{Z}_4$

Consider *quadratic* polynomials $f(x_1, x_2, \ldots, x_n)$ **modulo 4**.

- Counting the number of zeroes is in P. (Follows by [Cai-Chen-Lipton-Luo, 2010].)

Dichotomy Example Over $\mathbb{Z}_4$

Consider *quadratic* polynomials $f(x_1, x_2, \ldots, x_n)$ **modulo 4**.

- Counting the number of zeroes is in P. (Follows by [Cai-Chen-Lipton-Luo, 2010].)
- Counting the number of zeroes in $\{0, 1\}^n$ is #P-complete.

Dichotomy Example Over $\mathbb{Z}_4$

Consider *quadratic* polynomials $f(x_1, x_2, \ldots, x_n)$ **modulo 4**.

- Counting the number of zeroes is in P. (Follows by [Cai-Chen-Lipton-Luo, 2010].)
- Counting the number of zeroes in $\{0, 1\}^n$ is #P-complete.
- But if all cross-terms are $2x_ix_j$ it is in P again.

Dichotomy Example Over $\mathbb{Z}_4$

Consider *quadratic* polynomials $f(x_1, x_2, \ldots, x_n)$ **modulo 4**.

- Counting the number of zeroes is in P. (Follows by [Cai-Chen-Lipton-Luo, 2010].)
- Counting the number of zeroes in $\{0, 1\}^n$ is #P-complete.
- But if all cross-terms are $2x_ix_j$ it is in P again.

We will see how polynomials over $\mathbb{Z}_4$ characterize a neglected(?) library of *universal quantum circuits*.

Dichotomy Example Over $\mathbb{Z}_4$

Consider *quadratic* polynomials $f(x_1, x_2, \dots, x_n)$ **modulo 4**.

- Counting the number of zeroes is in P. (Follows by [Cai-Chen-Lipton-Luo, 2010].)
- Counting the number of zeroes in $\{0,1\}^n$ is #P-complete.
- But if all cross-terms are $2x_ix_j$ it is in P again.

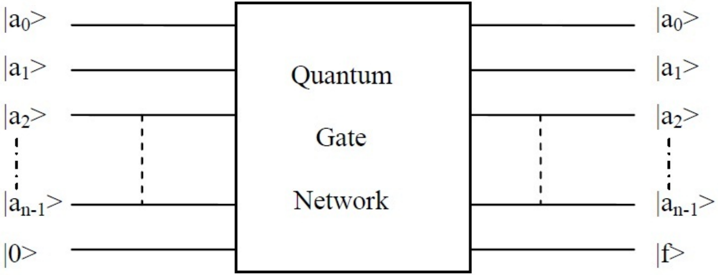
We will see how polynomials over $\mathbb{Z}_4$ characterize a neglected(?) library of *universal quantum circuits*.

Three kinds of combinatorial invariants for these circuits:

1. Phase-and-location ("Feynman Path") polynomials.
2. Graphs, and their generalization to *graphical 2-polymatroids*.
3. Versions of the Tutte Polynomial associated to such graphs and matroids.

Quantum Circuits

Quantum circuits look more constrained than Boolean circuits:

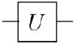


But Boolean circuits look similar if we do Savage's TM-to-circuit simulation and call each *column* for each tape cell a "cue-bit."

Quantum Gates—three slides by M. Rötteler

Quantum gates

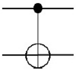
single qubit operation:



controlled-NOT:

control

target

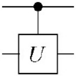


unitary matrix $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

controlled-U:

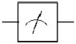
control

target



unitary matrix $= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{00} & U_{01} \\ 0 & 0 & U_{10} & U_{11} \end{pmatrix}$

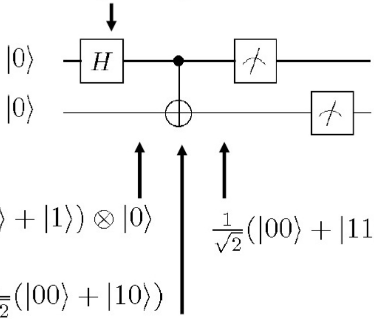
measurement in the $|0\rangle, |1\rangle$ **basis:**



[Slides concept by D. Bacon, U Washington]

Quantum circuit example

$$H \otimes \mathbf{1}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \mathbf{1}_2$$



$\text{Prob}(\text{meas. } |00\rangle) = 0.5$

$\text{Prob}(\text{meas. } |11\rangle) = 0.5$

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$$

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

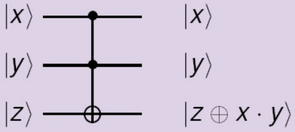
$$\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

$$|00\rangle \to |00\rangle$$
$$|01\rangle \to |01\rangle$$
$$|10\rangle \to |11\rangle$$
$$|11\rangle \to |10\rangle$$

Toffoli Gate

The Toffoli gate "TOF"

x	y	z	x'	y'	z'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0



Theorem (Toffoli, 1981)

Any reversible computation can be realized by using TOF gates and ancilla (auxiliary) bits which are initialized to 0.

Slides by Martin Rötteler

Some More Gates

$$\mathsf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathsf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathsf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

$$\mathsf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad \mathsf{T} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \quad \mathsf{R_8} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{bmatrix},$$

$$\mathsf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathsf{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad \mathsf{CS} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}.$$

Some More Gates

$$\mathsf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathsf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathsf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

$$\mathsf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad \mathsf{T} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \quad \mathsf{R}_8 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{bmatrix},$$

$$\mathsf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathsf{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad \mathsf{CS} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}.$$

- The gates $\mathsf{H}, \mathsf{X}, \mathsf{Y}, \mathsf{Z}, \mathsf{S}, \mathsf{CNOT}, \mathsf{CZ}$ generate *Clifford circuits*, which are simulatable in polynomial time.

Some More Gates

$$\mathsf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathsf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathsf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

$$\mathsf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad \mathsf{T} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \quad \mathsf{R}_8 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{bmatrix},$$

$$\mathsf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathsf{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad \mathsf{CS} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}.$$

- The gates $\mathsf{H}, \mathsf{X}, \mathsf{Y}, \mathsf{Z}, \mathsf{S}, \mathsf{CNOT}, \mathsf{CZ}$ generate *Clifford circuits*, which are simulatable in polynomial time. **(Time improved by us.)**

Some More Gates

$$\mathsf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathsf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathsf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

$$\mathsf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad \mathsf{T} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \quad \mathsf{R_8} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{bmatrix},$$

$$\mathsf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathsf{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad \mathsf{CS} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}.$$

- The gates $\mathsf{H}, \mathsf{X}, \mathsf{Y}, \mathsf{Z}, \mathsf{S}, \mathsf{CNOT}, \mathsf{CZ}$ generate *Clifford circuits*, which are simulatable in polynomial time. **(Time improved by us.)**
- Adding any of T, $\mathsf{R_8}$, CS, or Tof gives the full power of BQP.

Some More Gates

$$\mathsf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathsf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathsf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

$$\mathsf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad \mathsf{T} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \quad \mathsf{R_8} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{bmatrix},$$

$$\mathsf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathsf{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad \mathsf{CS} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}.$$

- The gates $\mathsf{H}, \mathsf{X}, \mathsf{Y}, \mathsf{Z}, \mathsf{S}, \mathsf{CNOT}, \mathsf{CZ}$ generate *Clifford circuits*, which are simulatable in polynomial time. **(Time improved by us.)**
- Adding any of T, $\mathsf{R_8}$, CS, or Tof gives the full power of BQP.
- Note: $\mathsf{T}^2 = \mathsf{S}$, $\mathsf{S}^2 = \mathsf{Z}$, $\mathsf{Z}^2 = \mathsf{I} = \mathsf{H}^2$, and $\mathsf{CS}^2 = \mathsf{CZ}$.

Three Universal Libraries

Three Universal Libraries

- The gate set H + CNOT + T is **efficiently metrically universal**, meaning that any feasible quantum circuit of size s can be approximated to within entrywise error ϵ by a circuit of these gates only in size $O(s) \cdot (\log \frac{s}{\epsilon})^{O(1)}$. (See Solovay-Kitaev theorem.)

Three Universal Libraries

- The gate set $\mathsf{H} + \mathsf{CNOT} + \mathsf{T}$ is **efficiently metrically universal**, meaning that any feasible quantum circuit of size s can be approximated to within entrywise error ϵ by a circuit of these gates only in size $O(s) \cdot (\log \frac{s}{\epsilon})^{O(1)}$. (See Solovay-Kitaev theorem.)
- Programmed improvement by Peter Selinger and Neil Ross.

Three Universal Libraries

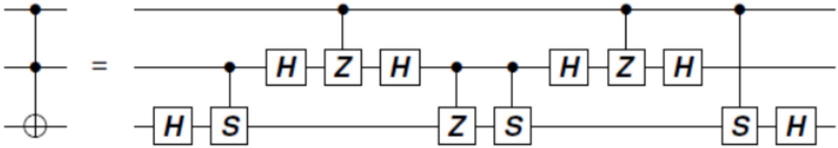
- The gate set $\mathsf{H} + \mathsf{CNOT} + \mathsf{T}$ is **efficiently metrically universal**, meaning that any feasible quantum circuit of size s can be approximated to within entrywise error ϵ by a circuit of these gates only in size $O(s) \cdot (\log \frac{s}{\epsilon})^{O(1)}$. (See Solovay-Kitaev theorem.)
- Programmed improvement by Peter Selinger and Neil Ross.
- The gate set $\mathsf{H} + \mathsf{Tof}$ is not metrically universal—it has no complex scalars—but it is **computationally universal**: It can maintain real and complex parts of quantum states in double-rail manner.

Three Universal Libraries

- The gate set H + CNOT + T is **efficiently metrically universal**, meaning that any feasible quantum circuit of size s can be approximated to within entrywise error ϵ by a circuit of these gates only in size $O(s) \cdot (\log \frac{s}{\epsilon})^{O(1)}$. (See Solovay-Kitaev theorem.)
- Programmed improvement by Peter Selinger and Neil Ross.
- The gate set H + Tof is not metrically universal—it has no complex scalars—but it is **computationally universal**: It can maintain real and complex parts of quantum states in double-rail manner.
- The gate set H + CS is efficiently metrically universal.

Three Universal Libraries

- The gate set H + CNOT + T is **efficiently metrically universal**, meaning that any feasible quantum circuit of size s can be approximated to within entrywise error ϵ by a circuit of these gates only in size $O(s) \cdot (\log \frac{s}{\epsilon})^{O(1)}$. (See Solovay-Kitaev theorem.)
- Programmed improvement by Peter Selinger and Neil Ross.
- The gate set H + Tof is not metrically universal—it has no complex scalars—but it is **computationally universal**: It can maintain real and complex parts of quantum states in double-rail manner.
- The gate set H + CS is efficiently metrically universal. **Note also:**



I. Feynman Path Polynomials

Let C have "minphase" $K = 2^k$ and let F embed K-th roots of unity ω.

- H + Tof has $k = 1$, $K = 2$.
- H + CS has $k = 2$, $K = 4$.
- H + CNOT + T has $k = 3$, $K = 8$.

I. Feynman Path Polynomials

Let C have "minphase" $K = 2^k$ and let F embed K-th roots of unity ω.

- H + Tof has $k = 1$, $K = 2$.
- H + CS has $k = 2$, $K = 4$.
- H + CNOT + T has $k = 3$, $K = 8$.

Theorem (RC 2007-09, extending Dawson et al. (2004) over $\mathbb{Z}_2$)

Any QC C of n qubits quickly transforms into a polynomial $P_C = \prod_g P_g$ over gates g and a constant $R > 0$ such that for all $x, z \in \{0,1\}^n$:

$$\langle z \mid C \mid x\rangle = \frac{1}{R}\sum_{j=0}^{K-1} \omega^j (\#y : P_C(x,y,z) = \iota(\omega^j))$$

I. Feynman Path Polynomials

Let C have "minphase" $K = 2^k$ and let F embed K-th roots of unity ω.

- H + Tof has $k = 1$, $K = 2$.
- H + CS has $k = 2$, $K = 4$.
- H + CNOT + T has $k = 3$, $K = 8$.

Theorem (RC 2007-09, extending Dawson et al. (2004) over $\mathbb{Z}_2$)

Any QC C of n qubits quickly transforms into a polynomial $P_C = \prod_g P_g$ over gates g and a constant $R > 0$ such that for all $x, z \in \{0,1\}^n$:

$$\langle z \mid C \mid x \rangle = \frac{1}{R} \sum_{j=0}^{K-1} \omega^j (\#y : P_C(x,y,z) = \iota(\omega^j)) = \frac{1}{R} \sum_y \omega^{P_C(x,y,z)},$$

where C has h nondeterministic (Hadamard) gates and $y \in \{0,1\}^h$.

Additive Case (Cf. Bacon-van Dam-Russell [2008])

Theorem (RC (2007-09), RCG (2018))

Given C and K, we can efficiently compute a polynomial $Q_C(x_1, \ldots, x_n, y_1, \ldots, y_h, z_1, \ldots, z_n, w_1, \ldots, w_t)$ of degree $O(1)$ over $\mathbb{Z}_K$ and a constant R' such that for all $x, z \in \{0,1\}^n$:

$$\langle z \mid C \mid x \rangle = \frac{1}{R'} \sum_{j=0}^{K-1} \omega^j (\# y, w : Q_C(x, y, z, w) = j)$$

Additive Case (Cf. Bacon-van Dam-Russell [2008])

Theorem (RC (2007-09), RCG (2018))

Given C and K, we can efficiently compute a polynomial $Q_C(x_1,\ldots,x_n,y_1,\ldots,y_h,z_1,\ldots,z_n,w_1,\ldots,w_t)$ of degree O(1) over $\mathbb{Z}_K$ and a constant R' such that for all $x,z\in\{0,1\}^n$:

$$\langle z \mid C \mid x\rangle = \frac{1}{R'}\sum_{j=0}^{K-1}\omega^j(\#y,w : Q_C(x,y,z,w)=j) = \frac{1}{R'}\sum_{y,w}\omega^{Q_C(x,y,z,w)},$$

where Q_C has the form $\sum_{gates\ g} q_g + \sum_{constraints\ c} q_c$.

Additive Case (Cf. Bacon-van Dam-Russell [2008])

Theorem (RC (2007-09), RCG (2018))

Given C and K, we can efficiently compute a polynomial $Q_C(x_1, \ldots, x_n, y_1, \ldots, y_h, z_1, \ldots, z_n, w_1, \ldots, w_t)$ of degree $O(1)$ over $\mathbb{Z}_K$ and a constant R' such that for all $x, z \in \{0,1\}^n$:

$$\langle z \mid C \mid x \rangle = \frac{1}{R'} \sum_{j=0}^{K-1} \omega^j (\#y, w : Q_C(x,y,z,w) = j) = \frac{1}{R'} \sum_{y,w} \omega^{Q_C(x,y,z,w)},$$

where Q_C has the form $\sum_{gates\ g} q_g + \sum_{constraints\ c} q_c$.

- Gives a particularly efficient reduction from BQP to #P.

Additive Case (Cf. Bacon-van Dam-Russell [2008])

Theorem (RC (2007-09), RCG (2018))

Given C and K, we can efficiently compute a polynomial $Q_C(x_1,\ldots,x_n,y_1,\ldots,y_h,z_1,\ldots,z_n,w_1,\ldots,w_t)$ of degree $O(1)$ over $\mathbb{Z}_K$ and a constant R' such that for all $x,z \in \{0,1\}^n$:

$$\langle z \mid C \mid x\rangle = \frac{1}{R'}\sum_{j=0}^{K-1}\omega^j(\#y,w : Q_C(x,y,z,w)=j) = \frac{1}{R'}\sum_{y,w}\omega^{Q_C(x,y,z,w)},$$

where Q_C has the form $\sum_{gates\ g} q_g + \sum_{constraints\ c} q_c$.

- Gives a particularly efficient reduction from BQP to #P.
- In P_C, illegal paths that violate some constraint incur the value 0.

Additive Case (Cf. Bacon-van Dam-Russell [2008])

Theorem (RC (2007-09), RCG (2018))

Given C and K, we can efficiently compute a polynomial $Q_C(x_1,\dots,x_n,y_1,\dots,y_h,z_1,\dots,z_n,w_1,\dots,w_t)$ of degree $O(1)$ over $\mathbb{Z}_K$ and a constant R' such that for all $x,z \in \{0,1\}^n$:

$$\langle z \mid C \mid x\rangle = \frac{1}{R'}\sum_{j=0}^{K-1}\omega^j(\#y,w : Q_C(x,y,z,w) = j) = \frac{1}{R'}\sum_{y,w}\omega^{Q_C(x,y,z,w)},$$

where Q_C has the form $\sum_{gates\ g} q_g + \sum_{constraints\ c} q_c$.

- Gives a particularly efficient reduction from BQP to #P.
- In P_C, illegal paths that violate some constraint incur the value 0.
- In Q_C, any violation creates an additive term $T = w_1 \cdots w_{\log_2 K}$ using fresh variables whose assignments give all values in $0 \,..\, K-1$, which *cancel*.

Additive Case (Cf. Bacon-van Dam-Russell [2008])

Theorem (RC (2007-09), RCG (2018))

Given C and K, we can efficiently compute a polynomial $Q_C(x_1, \dots, x_n, y_1, \dots, y_h, z_1, \dots, z_n, w_1, \dots, w_t)$ of degree $O(1)$ over $\mathbb{Z}_K$ and a constant R' such that for all $x, z \in \{0,1\}^n$:

$$\langle z \mid C \mid x \rangle = \frac{1}{R'} \sum_{j=0}^{K-1} \omega^j (\#y, w : Q_C(x,y,z,w) = j) = \frac{1}{R'} \sum_{y,w} \omega^{Q_C(x,y,z,w)},$$

where Q_C has the form $\sum_{gates\ g} q_g + \sum_{constraints\ c} q_c$.

- Gives a particularly efficient reduction from BQP to #P.
- In P_C, illegal paths that violate some constraint incur the value 0.
- In Q_C, any violation creates an additive term $T = w_1 \cdots w_{\log_2 K}$ using fresh variables whose assignments give all values in $0 \,..\, K-1$, which *cancel*. (This trick is my main original contribution.)

Constructing the Polynomials

Constructing the Polynomials

- Initially $P_C = 1$, $Q_C = 0$.

Constructing the Polynomials

- Initially $P_C = 1$, $Q_C = 0$.
- For Hadamard on line i (u_i—H–), allocate new variable y_j and do:

$$\begin{aligned} P_C \quad &*= \quad (1 - u_i y_j) \\ Q_C \quad &+= \quad 2^{k-1} u_i y_j . \end{aligned}$$

Constructing the Polynomials

- Initially $P_C = 1$, $Q_C = 0$.
- For Hadamard on line i (u_i—H–), allocate new variable y_j and do:

$$\begin{aligned} P_C &\;*= & (1 - u_i y_j) \\ Q_C &\;+= & 2^{k-1} u_i y_j. \end{aligned}$$

- CNOT with incoming terms u_i on control, u_j on target: u_i stays, $u_j := 2u_i u_j - u_i - u_j$.

Constructing the Polynomials

- Initially $P_C = 1$, $Q_C = 0$.
- For Hadamard on line i (u_i—H–), allocate new variable y_j and do:

$$\begin{aligned} P_C \quad &*= \quad (1 - u_i y_j) \\ Q_C \quad &+= \quad 2^{k-1} u_i y_j. \end{aligned}$$

- CNOT with incoming terms u_i on control, u_j on target: u_i stays, $u_j := 2u_i u_j - u_i - u_j$. No change to P_C or Q_C.

Constructing the Polynomials

- Initially $P_C = 1$, $Q_C = 0$.
- For Hadamard on line i (u_i—H–), allocate new variable y_j and do:

$$\begin{aligned} P_C &\;*= & (1 - u_i y_j) \\ Q_C &\;+= & 2^{k-1} u_i y_j. \end{aligned}$$

- CNOT with incoming terms u_i on control, u_j on target: u_i stays, $u_j := 2u_i u_j - u_i - u_j$. No change to P_C or Q_C.
- S-gate: Q_C adds u_i^2.

Constructing the Polynomials

- Initially $P_C = 1$, $Q_C = 0$.
- For Hadamard on line i (u_i—H–), allocate new variable y_j and do:

$$\begin{aligned} P_C \quad &*= \quad (1 - u_i y_j) \\ Q_C \quad &+= \quad 2^{k-1} u_i y_j. \end{aligned}$$

- CNOT with incoming terms u_i on control, u_j on target: u_i stays, $u_j := 2u_iu_j - u_i - u_j$. No change to P_C or Q_C.
- S-gate: Q_C adds u_i^2.
- CS-gate: Q_C adds u_iu_j.

Constructing the Polynomials

- Initially $P_C = 1$, $Q_C = 0$.
- For Hadamard on line i (u_i—H–), allocate new variable y_j and do:

$$\begin{aligned} P_C &\ *= \ (1 - u_i y_j) \\ Q_C &\ += \ 2^{k-1} u_i y_j. \end{aligned}$$

- CNOT with incoming terms u_i on control, u_j on target: u_i stays, $u_j := 2u_i u_j - u_i - u_j$. No change to P_C or Q_C.
- S-gate: Q_C adds u_i^2.
- CS-gate: Q_C adds $u_i u_j$.
- Thereby CS escapes the easy case over $\mathbb{Z}_4$ (with $k = 2$).

Constructing the Polynomials

- Initially $P_C = 1$, $Q_C = 0$.
- For Hadamard on line i (u_i—H–), allocate new variable y_j and do:

$$\begin{aligned} P_C &\quad *= \quad (1 - u_i y_j) \\ Q_C &\quad += \quad 2^{k-1} u_i y_j. \end{aligned}$$

- CNOT with incoming terms u_i on control, u_j on target: u_i stays, $u_j := 2u_i u_j - u_i - u_j$. No change to P_C or Q_C.
- S-gate: Q_C adds u_i^2.
- CS-gate: Q_C adds $u_i u_j$.
- Thereby CS escapes the easy case over $\mathbb{Z}_4$ (with $k = 2$).
- TOF: controls u_i, u_j stay, target u_k changes to $2u_i u_j u_k -$ $u_i u_j$ $- u_k$.

Constructing the Polynomials

- Initially $P_C = 1$, $Q_C = 0$.
- For Hadamard on line i (u_i—H–), allocate new variable y_j and do:

$$\begin{aligned} P_C &\ *= &(1 - u_i y_j) \\ Q_C &\ += &2^{k-1} u_i y_j. \end{aligned}$$

- CNOT with incoming terms u_i on control, u_j on target: u_i stays, $u_j := 2u_i u_j - u_i - u_j$. No change to P_C or Q_C.
- S-gate: Q_C adds u_i^2.
- CS-gate: Q_C adds $u_i u_j$.
- Thereby CS escapes the easy case over $\mathbb{Z}_4$ (with $k = 2$).
- TOF: controls u_i, u_j stay, target u_k changes to $2u_i u_j u_k -$ $u_i u_j$ $- u_k$.
- T-gate also goes cubic.

Logical Simulation

Theorem (C. Guan in RCG 2018)

Given C, n, K, h as above, we can quickly build a Boolean formula ϕ_C in variables $y_1, \ldots, y_h$, together with substituted-for $x_1, \ldots, x_n$, $z_1, \ldots, z_n$, and other "forced" variables such that for all $x, z \in \{0,1\}^n$:

$$\langle z \mid C \mid x \rangle = \frac{1}{R} \sum_{j=0}^{K-1} \omega^j \cdot \#sat(\phi_C).$$

Logical Simulation

Theorem (C. Guan in RCG 2018)

Given C, n, K, h as above, we can quickly build a Boolean formula ϕ_C in variables $y_1, \dots, y_h$, together with substituted-for $x_1, \dots, x_n$, $z_1, \dots, z_n$, and other "forced" variables such that for all $x, z \in \{0,1\}^n$:

$$\langle z \mid C \mid x \rangle = \frac{1}{R} \sum_{j=0}^{K-1} \omega^j \cdot \#sat(\phi_C).$$

- The ϕ is a conjunction of "controlled bitflips" $p' = p \oplus (u \wedge v)$.

Logical Simulation

Theorem (C. Guan in RCG 2018)

Given C, n, K, h as above, we can quickly build a Boolean formula ϕ_C in variables $y_1, \ldots, y_h$, together with substituted-for $x_1, \ldots, x_n$, $z_1, \ldots, z_n$, and other "forced" variables such that for all $x, z \in \{0,1\}^n$:

$$\langle z \mid C \mid x \rangle = \frac{1}{R} \sum_{j=0}^{K-1} \omega^j \cdot \#sat(\phi_C).$$

- The ϕ is a conjunction of "controlled bitflips" $p' = p \oplus (u \wedge v)$.
- Easy to transform into 3CNF (i.e., "3SAT" form). (show demo)

Logical Simulation

Theorem (C. Guan in RCG 2018)

Given C, n, K, h *as above, we can quickly build a Boolean formula* ϕ_C *in variables* $y_1, \ldots, y_h$*, together with substituted-for* $x_1, \ldots, x_n$*,* $z_1, \ldots, z_n$*, and other "forced" variables such that for all* $x, z \in \{0,1\}^n$*:*

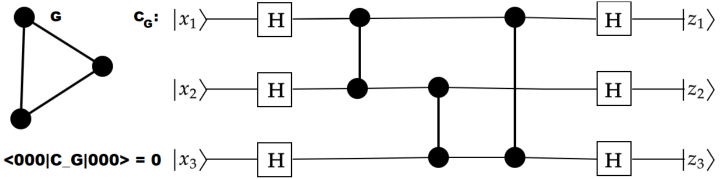
$$\langle z \mid C \mid x \rangle = \frac{1}{R} \sum_{j=0}^{K-1} \omega^j \cdot \#sat(\phi_C).$$

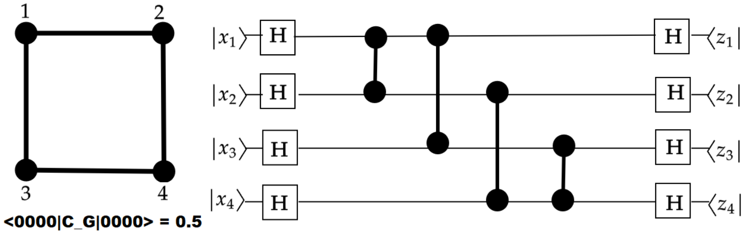
- The ϕ is a conjunction of "controlled bitflips" $p' = p \oplus (u \wedge v)$.
- Easy to transform into 3CNF (i.e., "3SAT" form). **(show demo)**
- For $K = 2, 4$ (i.e., for H + Tof and H + CS), we get the acceptance *probability* as a simple difference:

$$|\langle z \mid C \mid x \rangle|^2 = \frac{1}{R} \left(\#sat(\phi_C) - \#sat(\phi'_C) \right).$$

II. Strong Simulation of Graph State Circuits

Computing amplitudes $\langle z \mid C \mid x\rangle$ for Clifford circuits C can be efficiently reduced to computing $\langle 0^n \mid C_G \mid 0^n\rangle$ for **graph-state circuits** C_G of graphs G, using H and CZ gates, as exemplified by:





Improved From $O(n^3)$ to $O(n^{2.37155...})$

Theorem (Guan-Regan, 2019)

For n-qubit stabilizer circuits of size s, $\langle z \mid C \mid x \rangle$ *can be computed in* $O(s + n^\omega)$ *time, where* $\omega \leq 2.37155...$ *is the exponent of multiplying* $n \times n$ *matrices.*

Improved From $O(n^3)$ to $O(n^{2.37155...})$

Theorem (Guan-Regan, 2019)

For n-qubit stabilizer circuits of size s, $\langle z \mid C \mid x \rangle$ can be computed in $O(s + n^\omega)$ time, where $\omega \leq 2.37155...$ is the exponent of multiplying $n \times n$ matrices.

- Although C has $K = 2$, proof needs to use quadratic forms over $\mathbb{Z}_4$. And LDU decompositions over $\mathbb{Z}_2$ by Dumas-Pernet [2018].

Improved From $O(n^3)$ to $O(n^{2.37155...})$

Theorem (Guan-Regan, 2019)

For n-qubit stabilizer circuits of size s, $\langle z \mid C \mid x\rangle$ can be computed in $O(s + n^{\omega})$ time, where $\omega \leq 2.37155...$ is the exponent of multiplying $n \times n$ matrices.

- Although C has $K = 2$, proof needs to use quadratic forms over $\mathbb{Z}_4$. And LDU decompositions over $\mathbb{Z}_2$ by Dumas-Pernet [2018].
- **Corollary:** Counting solutions to quadratic polynomials $p(x_1, \ldots, x_n)$ over $\mathbb{Z}_2$ is in $O(n^{2.37155...})$ time.

Improved From $O(n^3)$ to $O(n^{2.37155...})$

Theorem (Guan-Regan, 2019)

For n-qubit stabilizer circuits of size s, $\langle z \mid C \mid x \rangle$ can be computed in $O(s + n^\omega)$ time, where $\omega \leq 2.37155...$ is the exponent of multiplying $n \times n$ matrices.

- Although C has $K = 2$, proof needs to use quadratic forms over $\mathbb{Z}_4$. And LDU decompositions over $\mathbb{Z}_2$ by Dumas-Pernet [2018].
- **Corollary:** Counting solutions to quadratic polynomials $p(x_1, \ldots, x_n)$ over $\mathbb{Z}_2$ is in $O(n^{2.37155...})$ time.

Improved From $O(n^3)$ to $O(n^{2.37155...})$

Theorem (Guan-Regan, 2019)

For n-qubit stabilizer circuits of size s, $\langle z \mid C \mid x\rangle$ can be computed in $O(s + n^{\omega})$ time, where $\omega \leq 2.37155...$ is the exponent of multiplying $n \times n$ matrices.

- Although C has $K = 2$, proof needs to use quadratic forms over $\mathbb{Z}_4$. And LDU decompositions over $\mathbb{Z}_2$ by Dumas-Pernet [2018].
- **Corollary:** Counting solutions to quadratic polynomials $p(x_1, \ldots, x_n)$ over $\mathbb{Z}_2$ is in $O(n^{2.37155...})$ time.
- Improves $O(n^3)$ time of Ehrenfeucht-Karpinski (1990).

Improved From $O(n^3)$ to $O(n^{2.37155...})$

Theorem (Guan-Regan, 2019)

For n-qubit stabilizer circuits of size s, $\langle z \mid C \mid x \rangle$ can be computed in $O(s + n^\omega)$ time, where $\omega \leq 2.37155...$ is the exponent of multiplying $n \times n$ matrices.

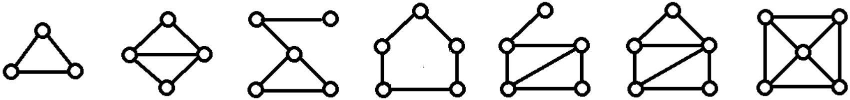
- Although C has $K = 2$, proof needs to use quadratic forms over $\mathbb{Z}_4$. And LDU decompositions over $\mathbb{Z}_2$ by Dumas-Pernet [2018].
- **Corollary:** Counting solutions to quadratic polynomials $p(x_1, \ldots, x_n)$ over $\mathbb{Z}_2$ is in $O(n^{2.37155...})$ time.
- Improves $O(n^3)$ time of Ehrenfeucht-Karpinski (1990).
- See Beaudrap and Herbert [2021] for other time/size/#H tradeoffs.

Improved From $O(n^3)$ to $O(n^{2.37155...})$

Theorem (Guan-Regan, 2019)

For n-qubit stabilizer circuits of size s, $\langle z \mid C \mid x\rangle$ can be computed in $O(s + n^{\omega})$ time, where $\omega \leq 2.37155...$ is the exponent of multiplying $n \times n$ matrices.

- Although C has $K = 2$, proof needs to use quadratic forms over $\mathbb{Z}_4$. And LDU decompositions over $\mathbb{Z}_2$ by Dumas-Pernet [2018].
- **Corollary:** Counting solutions to quadratic polynomials $p(x_1, \dots, x_n)$ over $\mathbb{Z}_2$ is in $O(n^{2.37155...})$ time.
- Improves $O(n^3)$ time of Ehrenfeucht-Karpinski (1990).
- See Beaudrap and Herbert [2021] for other time/size/#H tradeoffs.
- Can we recognize G with $\langle 0^n \mid C_G \mid 0^n\rangle = 0$ more quickly still?



From Graphs to Polymatroids

From Graphs to Polymatroids

- A self-loop on node i becomes a Z-gate on qubit line i.

From Graphs to Polymatroids

- A self-loop on node i becomes a Z-gate on qubit line i.
- An S-gate on line i would then be a "half loop."

From Graphs to Polymatroids

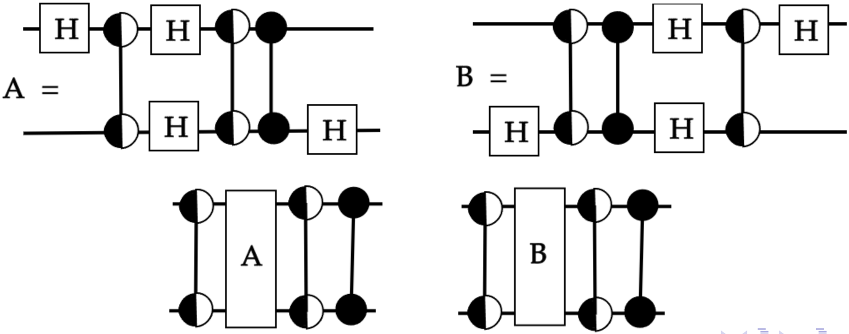
- A self-loop on node i becomes a Z-gate on qubit line i.
- An S-gate on line i would then be a "half loop."
- A CS gate would then be a "half edge."

From Graphs to Polymatroids

- A self-loop on node i becomes a Z-gate on qubit line i.
- An S-gate on line i would then be a “half loop.”
- A CS gate would then be a “half edge.”
- Formalizable as a **polymatroid** (PM). Into universal QC now.

From Graphs to Polymatroids

- A self-loop on node i becomes a Z-gate on qubit line i.
- An S-gate on line i would then be a "half loop."
- A CS gate would then be a "half edge."
- Formalizable as a **polymatroid** (PM). Into universal QC now.
- John Preskill's notes show that the following four widgets, together with their conjugations by $\mathsf{H} \otimes \mathsf{H}$, suffice:



New Heuristic Forms to Investigate

- Would be a "PM State Circuit"—except for all those H gates in the middle.

New Heuristic Forms to Investigate

- Would be a “PM State Circuit”—except for all those H gates in the middle.
- Can we move them to the sides, as with graph state circuits?

New Heuristic Forms to Investigate

- Would be a "PM State Circuit"—except for all those H gates in the middle.
- Can we move them to the sides, as with graph state circuits?
- If not, are there other useful canonical forms, a-la this?

New Heuristic Forms to Investigate

- Would be a “PM State Circuit”—except for all those H gates in the middle.
- Can we move them to the sides, as with graph state circuits?
- If not, are there other useful canonical forms, a-la this?
- How about the power of PM state circuits by themselves?

New Heuristic Forms to Investigate

- Would be a “PM State Circuit”—except for all those H gates in the middle.
- Can we move them to the sides, as with graph state circuits?
- If not, are there other useful canonical forms, a-la this?
- How about the power of PM state circuits by themselves?
- Are they more amenable to algebraic or logical model-counting heuristics than general quantum circuits?

New Heuristic Forms to Investigate

- Would be a “PM State Circuit”—except for all those H gates in the middle.
- Can we move them to the sides, as with graph state circuits?
- If not, are there other useful canonical forms, a-la this?
- How about the power of PM state circuits by themselves?
- Are they more amenable to algebraic or logical model-counting heuristics than general quantum circuits?
- Chaowen and I also considered graphs that can have:
  - Loops not attached to a vertex, called *circles*.
  - Numbered copies of the empty graph, called *wisps*.
  - Wisps of negative sign, called *negative isols*.

New Heuristic Forms to Investigate

- Would be a "PM State Circuit"—except for all those H gates in the middle.
- Can we move them to the sides, as with graph state circuits?
- If not, are there other useful canonical forms, a-la this?
- How about the power of PM state circuits by themselves?
- Are they more amenable to algebraic or logical model-counting heuristics than general quantum circuits?
- Chaowen and I also considered graphs that can have:
  - Loops not attached to a vertex, called *circles*.
  - Numbered copies of the empty graph, called *wisps*.
  - Wisps of negative sign, called *negative isols*.
- They can be formalized via (*graphical*) 2-polymatroids. Call them "(G)2PMs."

New Heuristic Forms to Investigate

- Would be a "PM State Circuit"—except for all those H gates in the middle.
- Can we move them to the sides, as with graph state circuits?
- If not, are there other useful canonical forms, a-la this?
- How about the power of PM state circuits by themselves?
- Are they more amenable to algebraic or logical model-counting heuristics than general quantum circuits?
- Chaowen and I also considered graphs that can have:
    - Loops not attached to a vertex, called *circles*.
    - Numbered copies of the empty graph, called *wisps*.
    - Wisps of negative sign, called *negative isols*.
- They can be formalized via (*graphical*) 2-polymatroids. Call them "(G)2PMs."
- We took them in a different direction.

III. New Generalized Tutte-Grothendieck Invariant

For any G2PM G, we define its **amplitude polynomial** $Q_G(x)$, of just one variable x, inductively like so:

III. New Generalized Tutte-Grothendieck Invariant

For any G2PM G, we define its **amplitude polynomial** $Q_G(x)$, of just one variable x, inductively like so:

- If G has ℓ isolated nodes, k circles, and any number of wisps or negative isols (i.e., no edges besides circles), then

$$Q_G(x) = (-1)^k x^\ell.$$

III. New Generalized Tutte-Grothendieck Invariant

For any G2PM G, we define its **amplitude polynomial** $Q_G(x)$, of just one variable x, inductively like so:

- If G has ℓ isolated nodes, k circles, and any number of wisps or negative isols (i.e., no edges besides circles), then

$$Q_G(x) = (-1)^k x^\ell.$$

- Else, if G has a loop e at some node, define

$$Q_G(x) = Q_{G\backslash e} - Q_{G\backslash\backslash e}.$$

III. New Generalized Tutte-Grothendieck Invariant

For any G2PM G, we define its **amplitude polynomial** $Q_G(x)$, of just one variable x, inductively like so:

- If G has ℓ isolated nodes, k circles, and any number of wisps or negative isols (i.e., no edges besides circles), then

$$Q_G(x) = (-1)^k x^\ell.$$

- Else, if G has a loop e at some node, define

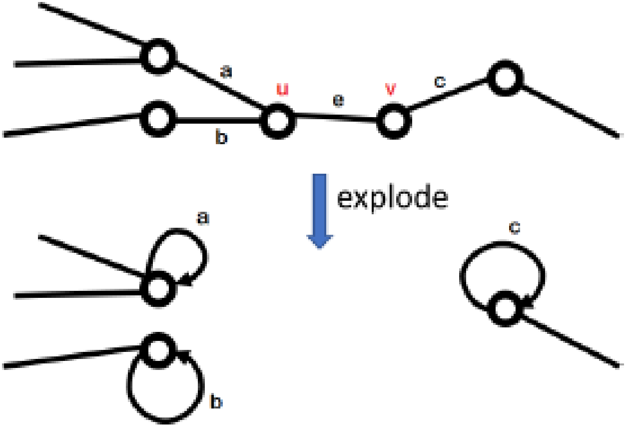
$$Q_G(x) = Q_{G\setminus e} - Q_{G\setminus\setminus e}.$$

- Else, if G has an edge e between two nodes, define

$$Q_G(x) = Q_{G\setminus e} - \frac{1}{2}Q_{G\setminus\setminus e}.$$

Here $G \setminus e$ means deleting edge e, but $G \setminus \setminus e$ means "**exploding**" e. The recursion is *confluent*—order of choosing e does not matter.

Exploding an Edge



Properties of the Amplitude Polynomial

We connect Q_G to the **rank-generating polynomial** S_G of J. Oxley and G. Whittle, and a variant form S'_G, by

Theorem

$$Q_G(x) = \left(\frac{1}{\alpha}\right)^n S'_G(\alpha x, -\alpha) = \left(\frac{1}{\alpha}\right)^n S_G(\alpha x, -\alpha)(\alpha x)^r,$$

where $\alpha = -i\sqrt{2}$ *and* r *is the number of isolated nodes of* G.

Drawing on their definition of a *generalized Tutte-Grothendieck invariant* (GTGI), we show:

Theorem

Q_G *is a GTGI of graphs* G *and belongs to the first of only two possible families of GTGIs that can arise from G2PMs*

Even More Speculative

Even More Speculative

- What are these good for? Many computational problems boil down to evaluating generative polynomials (Tutte, Jones, etc.) at specific points x_0. Classifying complexity of $Q_G(x_0)$ may channel simulation problems about QCs.

Even More Speculative

- What are these good for? Many computational problems boil down to evaluating generative polynomials (Tutte, Jones, etc.) at specific points x_0. Classifying complexity of $Q_G(x_0)$ may channel simulation problems about QCs.
- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and the effort needed to maintain *coherence* in universal QC.

Even More Speculative

- What are these good for? Many computational problems boil down to evaluating generative polynomials (Tutte, Jones, etc.) at specific points x_0. Classifying complexity of $Q_G(x_0)$ may channel simulation problems about QCs.
- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and the effort needed to maintain *coherence* in universal QC.
- Baur-Strassen showed that $\Omega(\log_2 \gamma(f))$ lower-bounds the arithmetical complexity of f, indeed the number of binary multiplication gates.

Even More Speculative

- What are these good for? Many computational problems boil down to evaluating generative polynomials (Tutte, Jones, etc.) at specific points x_0. Classifying complexity of $Q_G(x_0)$ may channel simulation problems about QCs.
- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and the effort needed to maintain *coherence* in universal QC.
- Baur-Strassen showed that $\Omega(\log_2 \gamma(f))$ lower-bounds the arithmetical complexity of f, indeed the number of binary multiplication gates.
- Yields $\Omega(n \log n)$ lower bound on circuits for $f = x_1^n + \cdots + x_n^n$.

Even More Speculative

- What are these good for? Many computational problems boil down to evaluating generative polynomials (Tutte, Jones, etc.) at specific points x_0. Classifying complexity of $Q_G(x_0)$ may channel simulation problems about QCs.
- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and the effort needed to maintain *coherence* in universal QC.
- Baur-Strassen showed that $\Omega(\log_2 \gamma(f))$ lower-bounds the arithmetical complexity of f, indeed the number of binary multiplication gates.
- Yields $\Omega(n \log n)$ lower bound on circuits for $f = x_1^n + \cdots + x_n^n$.
- Piddling, but it remains *the only super-linear lower bound known on any general measure of complexity*.

Even More Speculative

- What are these good for? Many computational problems boil down to evaluating generative polynomials (Tutte, Jones, etc.) at specific points x_0. Classifying complexity of $Q_G(x_0)$ may channel simulation problems about QCs.
- Invariants based on Strassen's *geometric degree* $\gamma(f)$ concept may help quantify both entanglement and the effort needed to maintain *coherence* in universal QC.
- Baur-Strassen showed that $\Omega(\log_2 \gamma(f))$ lower-bounds the arithmetical complexity of f, indeed the number of binary multiplication gates.
- Yields $\Omega(n \log n)$ lower bound on circuits for $f = x_1^n + \cdots + x_n^n$.
- Piddling, but it remains *the only super-linear lower bound known on any general measure of complexity*.
- Does $\gamma(P_C)$ witness a physical nonlinearity associated with operating quantum circuits C?

Other Web Sources

Other Web Sources

- https://rjlipton.com/2022/01/05/quantum-graph-theory/

Other Web Sources

- https://rjlipton.com/2022/01/05/quantum-graph-theory/
- https://rjlipton.com/2019/06/17/contraction-and-explosion/

Other Web Sources

- https://rjlipton.com/2022/01/05/quantum-graph-theory/
- https://rjlipton.com/2019/06/17/contraction-and-explosion/
- https://rjlipton.com/2019/08/26/a-matroid-quantum-connection/

Other Web Sources

- https://rjlipton.com/2022/01/05/quantum-graph-theory/
- https://rjlipton.com/2019/06/17/contraction-and-explosion/
- https://rjlipton.com/2019/08/26/a-matroid-quantum-connection/
- https://rjlipton.com/2021/11/01/quantum-trick-or-treat/ (chaos in quantum walks)

Other Web Sources

- https://rjlipton.com/2022/01/05/quantum-graph-theory/
- https://rjlipton.com/2019/06/17/contraction-and-explosion/
- https://rjlipton.com/2019/08/26/a-matroid-quantum-connection/
- https://rjlipton.com/2021/11/01/quantum-trick-or-treat/ (chaos in quantum walks)
- https://rjlipton.com/2019/06/10/net-zero-graphs/

Other Web Sources

- https://rjlipton.com/2022/01/05/quantum-graph-theory/
- https://rjlipton.com/2019/06/17/contraction-and-explosion/
- https://rjlipton.com/2019/08/26/a-matroid-quantum-connection/
- https://rjlipton.com/2021/11/01/quantum-trick-or-treat/ (chaos in quantum walks)
- https://rjlipton.com/2019/06/10/net-zero-graphs/
- https://rjlipton.com/2012/07/08/grilling-quantum-circuits/

Other Web Sources

- https://rjlipton.com/2022/01/05/quantum-graph-theory/
- https://rjlipton.com/2019/06/17/contraction-and-explosion/
- https://rjlipton.com/2019/08/26/a-matroid-quantum-connection/
- https://rjlipton.com/2021/11/01/quantum-trick-or-treat/ (chaos in quantum walks)
- https://rjlipton.com/2019/06/10/net-zero-graphs/
- https://rjlipton.com/2012/07/08/grilling-quantum-circuits/
- Last one has links to expanded geometric degree and Baur-Strassen discussion.

Other Web Sources

- https://rjlipton.com/2022/01/05/quantum-graph-theory/
- https://rjlipton.com/2019/06/17/contraction-and-explosion/
- https://rjlipton.com/2019/08/26/a-matroid-quantum-connection/
- https://rjlipton.com/2021/11/01/quantum-trick-or-treat/ (chaos in quantum walks)
- https://rjlipton.com/2019/06/10/net-zero-graphs/
- https://rjlipton.com/2012/07/08/grilling-quantum-circuits/
- Last one has links to expanded geometric degree and Baur-Strassen discussion.
- Thanks for listening. Q & A.