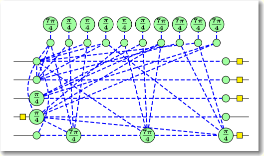
Symbolic synthesis of Clifford circuits **and beyond...**

Matt Amy[1], Owen Bennett-Gibbs[2], Neil Julien Ross[3]

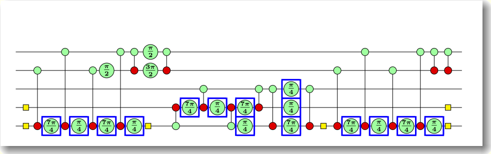
[1]Simon Fraser University
[2]McGill University
[2]Dalhousie University

Quantum Physics and Logic
Oxford, June 30, 2022

Extracting circuits from things



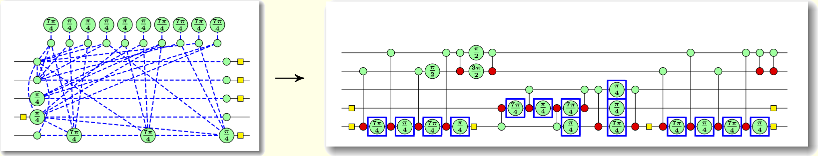
$\longrightarrow$



Circuit extraction from ZX/ZH diagrams (without gflow) is hard!

Kissinger & van de Wetering, Reducing T-count with the ZX-calculus. Phys. Rev. A (2020).

Extracting circuits from things



Circuit extraction from ZX/ZH diagrams (without gflow) is hard!

What if we extract from the sum-over-paths instead?

Kissinger & van de Wetering, Reducing T-count with the ZX-calculus. Phys. Rev. A (2020).

The sum-over-paths representation

A **path sum** is a symbolic expression of a linear operator $\Psi : \mathbb{C}^{2^m} \to \mathbb{C}^{2^n}$ as a sum indexed by binary variables:

$$\Psi|\mathrm{x}\rangle = |\Psi(\mathrm{x})\rangle = \mathcal{N} \sum_{\mathrm{y}\in\mathbb{Z}_2^k} e^{2\pi i P(\mathrm{x},\mathrm{y})} |f(\mathrm{x},\mathrm{y})\rangle,$$

The sum-over-paths representation

A **path sum** is a symbolic expression of a linear operator $\Psi : \mathbb{C}^{2^m} \to \mathbb{C}^{2^n}$ as a sum indexed by binary variables:

$$\Psi|\mathrm{x}\rangle = |\Psi(\mathrm{x})\rangle = \mathcal{N} \sum_{\mathrm{y} \in \mathbb{Z}_2^k} e^{2\pi i P(\mathrm{x},\mathrm{y})} |f(\mathrm{x},\mathrm{y})\rangle,$$

- $\mathcal{N} \in \mathbb{C} \setminus \{0\}$ is a normalization factor,

The sum-over-paths representation

A **path sum** is a symbolic expression of a linear operator $\Psi : \mathbb{C}^{2^m} \to \mathbb{C}^{2^n}$ as a sum indexed by binary variables:

$$\Psi|\mathrm{x}\rangle = |\Psi(\mathrm{x})\rangle = \mathcal{N} \sum_{\mathrm{y}\in\mathbb{Z}_2^k} e^{2\pi i P(\mathrm{x},\mathrm{y})}|f(\mathrm{x},\mathrm{y})\rangle,$$

- $\mathcal{N} \in \mathbb{C} \setminus \{0\}$ is a normalization factor,
- $P : \mathbb{Z}_2^m \times \mathbb{Z}_2^k \to \mathbb{R}$ is a real-valued multilinear polynomial, and

The sum-over-paths representation

A **path sum** is a symbolic expression of a linear operator $\Psi : \mathbb{C}^{2^m} \to \mathbb{C}^{2^n}$ as a sum indexed by binary variables:

$$\Psi|\mathsf{x}\rangle = |\Psi(\mathsf{x})\rangle = \mathcal{N} \sum_{\mathsf{y}\in\mathbb{Z}_2^k} e^{2\pi i P(\mathsf{x},\mathsf{y})}|f(\mathsf{x},\mathsf{y})\rangle,$$

- $\mathcal{N} \in \mathbb{C} \setminus \{0\}$ is a normalization factor,
- $P : \mathbb{Z}_2^m \times \mathbb{Z}_2^k \to \mathbb{R}$ is a real-valued multilinear polynomial, and
- $f : \mathbb{Z}_2^m \times \mathbb{Z}_2^k \to \mathbb{Z}_2^n$ is system of Boolean-valued multilinear polynomials

The sum-over-paths representation

A **path sum** is a symbolic expression of a linear operator $\Psi : \mathbb{C}^{2^m} \to \mathbb{C}^{2^n}$ as a sum indexed by binary variables:

$$\Psi|\mathrm{x}\rangle = |\Psi(\mathrm{x})\rangle = \mathcal{N} \sum_{\mathrm{y} \in \mathbb{Z}_2^k} e^{2\pi i P(\mathrm{x},\mathrm{y})} |f(\mathrm{x},\mathrm{y})\rangle,$$

- $\mathcal{N} \in \mathbb{C} \setminus \{0\}$ is a normalization factor,
- $P : \mathbb{Z}_2^m \times \mathbb{Z}_2^k \to \mathbb{R}$ is a real-valued multilinear polynomial, and
- $f : \mathbb{Z}_2^m \times \mathbb{Z}_2^k \to \mathbb{Z}_2^n$ is system of Boolean-valued multilinear polynomials

Examples

Phase & reversible gates:

$$S|x\rangle = i^x|x\rangle, \qquad T|x\rangle = \omega^x|x\rangle \text{ where } \omega = e^{2\pi i/8}$$

$$CNOT|x\rangle|y\rangle = |x\rangle|x \oplus y\rangle$$

Examples

Phase & reversible gates:

$$S|x\rangle = i^x|x\rangle, \qquad T|x\rangle = \omega^x|x\rangle \text{ where } \omega = e^{2\pi i/8}$$

$$CNOT|x\rangle|y\rangle = |x\rangle|x \oplus y\rangle$$

Branching gates:

$$H|x\rangle = \frac{1}{\sqrt{2}} \sum_y (-1)^{xy}|y\rangle$$

Examples

Phase & reversible gates:

$$S|x\rangle = i^x|x\rangle, \qquad T|x\rangle = \omega^x|x\rangle \text{ where } \omega = e^{2\pi i/8}$$

$$CNOT|x\rangle|y\rangle = |x\rangle|x \oplus y\rangle$$

Branching gates:

$$H|x\rangle = \frac{1}{\sqrt{2}}\sum\nolimits_y (-1)^{xy}|y\rangle$$

Cups & caps:

$$\subset = \sum\nolimits_y |y\rangle|y\rangle, \qquad \supset |x_1\rangle|x_2\rangle = \frac{1}{2}\sum\nolimits_y (-1)^{y(x_1+x_2)}$$

Examples

Phase & reversible gates:

$$S|x\rangle = i^x|x\rangle, \qquad T|x\rangle = \omega^x|x\rangle \text{ where } \omega = e^{2\pi i/8}$$

$$CNOT|x\rangle|y\rangle = |x\rangle|x \oplus y\rangle$$

Branching gates:

$$H|x\rangle = \frac{1}{\sqrt{2}}\sum\nolimits_y (-1)^{xy}|y\rangle$$

Cups & caps:

$$\subset \; = \sum\nolimits_y |y\rangle|y\rangle, \qquad \supset |x_1\rangle|x_2\rangle = \frac{1}{2}\sum\nolimits_y (-1)^{y(x_1+x_2)}$$

Gate composition:

$$TH|x\rangle = \frac{1}{\sqrt{2}}\sum\nolimits_y (-1)^{xy}(T|y\rangle) = \frac{1}{\sqrt{2}}\sum\nolimits_y (-1)^{xy}\omega^y|y\rangle$$

Recent work

Recent work on formalizing the sum-over-paths:

- Re-writing system and compositional model for circuits[1]
- Connections to graphical calculi[2,3] ($ZH \leftrightharpoons SOP$)
- Dagger compact structure[3]
- Complete re-write rules for ζ_2^k phases[4]

---

[1] Amy, Towards large-scale functional verification of universal quantum circuits. QPL 2018

[2] Lemmonier, Kissinger, van de Wetering, Hypergraph simplification: Linking the path-sum approach to the ZH-calculus QPL 2020.

[3] Vilmart, The Structure of Sum-Over-Paths, its Consequences, and Completeness for Clifford. FoSSaCs 2021.

[4] Vilmart, Completeness of Sum-Over-Paths for Toffoli-Hadamard and the Clifford Hierarchy. arXiv 2022.

Equational reasoning

Let Ψ be a path sum, f a Boolean polynomial, and assume $y \notin FV(\Psi)$ and $x, y \notin FV(f)$. Then the following equations hold.

$$\sum_{y} |\Psi\rangle = 2|\Psi\rangle \qquad \text{[E]}$$

$$\sum_{x,y} (-1)^{y(x+f)} |\Psi(x)\rangle = 2|\Psi(f)\rangle \qquad \text{[I]}$$

$$\sum_{y} i^{y}(-1)^{yf} |\Psi\rangle = \omega\sqrt{2}(-i)^{f} |\Psi\rangle \qquad \text{[U]}$$

$$\sum_{y} |\Psi(y)\rangle = \sum_{y} |\Psi(y+f)\rangle \qquad \text{[V]}$$

Proposition

The [E], [I], and [U] rules are complete for Stabilizer operations

Simplifications in SOP-land

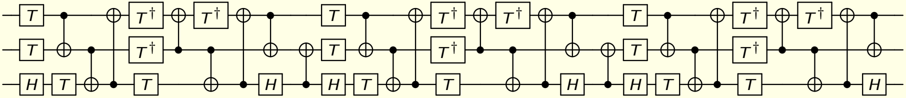
The original **phase polynomial optimization**[5] extracted a circuit from the simplified sum-over-paths

[5] Amy, Maslov, Mosca, *Polynomial-time T-depth Optimization of Clifford+T circuits via Matroid Partitioning*. IEEE TCAD 2014

Simplifications in SOP-land

The original **phase polynomial optimization**[5] extracted a circuit from the simplified sum-over-paths

More sophisticated re-writing can go further, but like in ZX-land circuit extraction gets harder!



$$\downarrow$$

$$|x_1 x_2 x_3\rangle \mapsto |x_1 (x_2 \oplus x_1 \cdot x_3) x_3\rangle$$

[5] Amy, Maslov, Mosca, *Polynomial-time T-depth Optimization of Clifford+T circuits via Matroid Partitioning*. IEEE TCAD 2014

Synthesis of the sum-over-paths

The synthesis problem

Given a promise that a path sum $|\Psi\rangle$ *represents a unitary transformation, synthesize/extract a circuit implementing* Ψ *over some gate set* $\mathcal{G}$

Synthesis of the sum-over-paths

The synthesis problem

Given a promise that a path sum $|\Psi\rangle$ represents a unitary transformation, synthesize/extract a circuit implementing Ψ over some gate set $\mathcal{G}$

In this work:

- Hardness of checking the unitarity condition
- Synthesis of Clifford path sums
- Synthesis of general path sums

The UNITARY problem

The UNITARY problem

The UNITARY problem

Given a sum-over-paths, does it represent a unitary transformation?

The UNITARY problem

The UNITARY problem

Given a sum-over-paths, does it represent a unitary transformation?

Theorem

The UNITARY problem is coNP*-hard.*

The UNITARY problem

The UNITARY problem

Given a sum-over-paths, does it represent a unitary transformation?

Theorem

The UNITARY problem is coNP*-hard.*

Proof sketch:

Reduce TAUT $\{=$ propositional tautologies$\}$ to UNITARY by constructing, for a propositional formula φ, a path sum

$$|x\rangle \mapsto \varphi(x)|x\rangle$$

The Tseytin transformation

For a Boolean **polynomial** p, there exists a direct encoding

$$p(\mathrm{x}) = 2^{-1} \sum_{y \in \mathbb{Z}_2} (-1)^{y(1+p(\mathrm{x}))}$$

However, writing a propositional formula φ as a polynomial, denoted $\overline{\varphi}$, may use exponential overhead.

We can instead use the **Tseytin transformation** to write φ as an equi-satisfiable conjunction of constant-size clauses

$$\begin{aligned}\mathcal{T}(x_1 \wedge (x_2 \vee (\neg x_3))) = &(z_1 \leftrightarrow \neg x_3) \wedge \\ &(z_2 \leftrightarrow (x_2 \vee z_1)) \wedge \\ &(z_3 \leftrightarrow (x_1 \wedge z_2)) \wedge \\ &z_3\end{aligned}$$

A polynomial size encoding

Given $\mathcal{T}(\varphi) = \wedge_i (z_i \leftrightarrow \varphi_i)$ encode $\mathcal{T}(\varphi)$ inductively as:

- $\Psi_{z\leftrightarrow\varphi} = 2^{-1} \sum_{y \in \mathbb{Z}_2} (-1)^{y(z+\overline{\varphi})}$
- $\Psi_{c_1 \wedge c_2} = \Psi_{c_1} \cdot \Psi_{c_2}$

By globally summing over all z_i we obtain a polynomial-size encoding over **only the free variables of φ**:

$$\varphi(\mathrm{x}) = 2^{-(k+1)} \sum_{y} \sum_{\mathrm{y} \in \mathbb{Z}_2^k} \sum_{\mathrm{z} \in \mathbb{Z}_2^k} (-1)^{y(1+z_1) + \sum_i y_i (z_i + \overline{\varphi_i}(\mathrm{x}))}$$

A polynomial size encoding

Given $\mathcal{T}(\varphi) = \wedge_i(z_i \leftrightarrow \varphi_i)$ encode $\mathcal{T}(\varphi)$ inductively as:

- $\Psi_{z\leftrightarrow\varphi} = 2^{-1}\sum_{y\in\mathbb{Z}_2}(-1)^{y(z+\overline{\varphi})}$
- $\Psi_{c_1\wedge c_2} = \Psi_{c_1}\cdot\Psi_{c_2}$

By globally summing over all z_i we obtain a polynomial-size encoding over **only the free variables of φ**:

$$\varphi(\mathrm{x}) = 2^{-(k+1)}\sum_{y}\sum_{\mathrm{y}\in\mathbb{Z}_2^k}\sum_{\mathrm{z}\in\mathbb{Z}_2^k}(-1)^{y(1+z_1)+\sum_i y_i(z_i+\overline{\varphi_i}(\mathrm{x}))}$$

Corollary

The UNITARY problem is coNP-*hard*

Synthesizing Clifford circuits

Clifford path sums

$$
\begin{aligned}
H &: |x\rangle \mapsto \sqrt{2}^{-1} \sum\nolimits_y (-1)^{xy} |y\rangle \\
S &: |x\rangle \mapsto i^x |x\rangle \\
CZ &: |x\rangle|y\rangle \mapsto (-1)^{xy} |x\rangle|y\rangle
\end{aligned}
$$

Path sums over Clifford gates have the form

$$
|x\rangle \mapsto \frac{1}{\sqrt{2^m}} \sum\nolimits_{\mathrm{y} \in \mathbb{Z}_2^m} i^{L(\mathrm{x},\mathrm{y})} (-1)^{Q(\mathrm{x},\mathrm{y})} |f(\mathrm{x},\mathrm{y})\rangle
$$

where

- L is linear,
- Q is pure quadratic, and
- f is affine.

Clifford normalization

Proposition (Affine normal form)

Any Clifford path sum can be re-written up to a permutation as

$$|\mathrm{x}\rangle \mapsto \frac{\omega^{l}}{\sqrt{2^k}} \sum_{\mathrm{y}\in\mathbb{Z}_2^k} i^{L(\mathrm{x},\mathrm{y})}(-1)^{Q(\mathrm{x},\mathrm{y})}|\mathrm{y}\rangle \otimes |f(\mathrm{x},\mathrm{y})\rangle$$

in polynomial time using the equations [E], [I], and [U].

Works by eliminating variables from the sum until a minimal spanning set for the affine subspace is obtained

Clifford normalization

Proposition (Affine normal form)

Any Clifford path sum can be re-written up to a permutation as

$$|\mathrm{x}\rangle \mapsto \frac{\omega^{l}}{\sqrt{2^k}} \sum_{\mathrm{y}\in\mathbb{Z}_2^k} i^{L(\mathrm{x},\mathrm{y})}(-1)^{Q(\mathrm{x},\mathrm{y})}|\mathrm{y}\rangle \otimes |f(\mathrm{x},\mathrm{y})\rangle$$

in polynomial time using the equations [E], [I], and [U].

Works by eliminating variables from the sum until a minimal spanning set for the affine subspace is obtained

Note: can be made **unique** with a tweak from Tommy & Miriam's work presented earlier this week

Decomposition into linear operators

Decomposing L, Q, and f into functions on **inputs** x, **affine basis variables** y, and x-y **cross terms**, the affine normal form factors into the following sequence of operators:

$$
\begin{aligned}
|\mathsf{x}\rangle &\mapsto \omega^{l} i^{L_x(\mathsf{x})}(-1)^{Q_x(\mathsf{x})}|\mathsf{x}\rangle && \{\mathrm{S}, \mathrm{CZ}\} \\
|\mathsf{x}\rangle &\mapsto |R(\mathsf{x})\rangle|f_x(\mathsf{x})\rangle && \{\mathrm{CNOT}\} \\
|R(\mathsf{x})\rangle|f_x(\mathsf{x})\rangle &\mapsto \frac{1}{\sqrt{2^k}}\sum\nolimits_{\mathsf{y}\in\mathbb{Z}_2^k}(-1)^{\sum_i y_i R_i(\mathsf{x})}|\mathsf{y}\rangle|f_x(\mathsf{x})\rangle && \{\mathrm{H}\} \\
|\mathsf{y}\rangle|f_x(\mathsf{x})\rangle &\mapsto |\mathsf{y}\rangle|f_x(\mathsf{x}) + f_y(\mathsf{y}) + \mathsf{b}\rangle && \{\mathrm{X}, \mathrm{CNOT}\} \\
|\mathsf{y}\rangle|f(\mathsf{x},\mathsf{y})\rangle &\mapsto i^{L_y(\mathsf{y})}(-1)^{Q_y(\mathsf{y})}|\mathsf{y}\rangle|f(\mathsf{x},\mathsf{y})\rangle && \{\mathrm{S}, \mathrm{CZ}\}
\end{aligned}
$$

A simple, constructive proof of the Bruhat decomposition

Any Clifford operator can be written in a **9** stage circuit[6]

$$\mathrm{S} \cdot \mathrm{CZ} \cdot \mathrm{X} \cdot \mathrm{CNOT} \cdot \mathrm{H} \cdot \mathrm{CNOT} \cdot \mathrm{X} \cdot \mathrm{CZ} \cdot \mathrm{S}$$

[6] Maslov, Roetteler, Shorter stabilizer circuits via Bruhat decomposition and quantum circuit transformations. IEEE TIT 2018.

A simple, constructive proof of the Bruhat decomposition

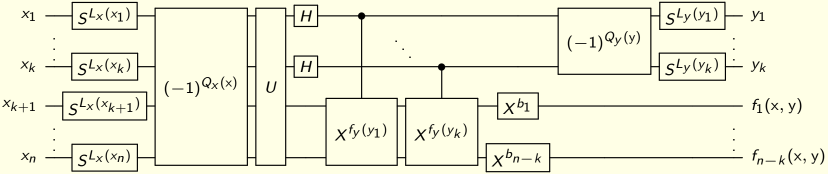
Any Clifford operator can be written in a **9** stage circuit[6]

$$\mathrm{S} \cdot \mathrm{CZ} \cdot \mathrm{X} \cdot \mathrm{CNOT} \cdot \mathrm{H} \cdot \mathrm{CNOT} \cdot \mathrm{X} \cdot \mathrm{CZ} \cdot \mathrm{S}$$

Theorem

Any Clifford operator can be synthesized in polynomial time over $\{\mathrm{CNOT}, \mathrm{X}, \mathrm{CZ}, \mathrm{S}, \mathrm{H}\}$ *as an* 8 *stage circuit of the form*

$$\mathrm{S} \cdot \mathrm{CZ} \cdot \mathrm{CNOT} \cdot \mathrm{H} \cdot \mathrm{CNOT} \cdot \mathrm{X} \cdot \mathrm{CZ} \cdot \mathrm{S}$$



[6] Maslov, Roetteler, Shorter stabilizer circuits via Bruhat decomposition and quantum circuit transformations. IEEE TIT 2018.

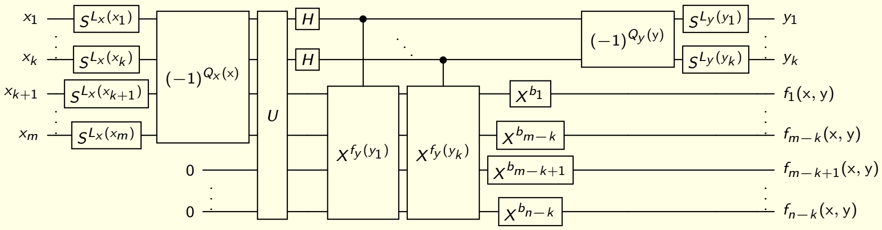
From Clifford circuits to stabilizer states

(and in between)

Corollary

An affine normal form $|\mathrm{x}\rangle \mapsto \frac{\omega^l}{\sqrt{2^k}} \sum_{\mathrm{y}\in\mathbb{Z}_2^k} i^{L(\mathrm{x},\mathrm{y})}(-1)^{Q(\mathrm{x},\mathrm{y})}|\mathrm{y}\rangle \otimes |f(\mathrm{x},\mathrm{y})\rangle$ *can be implemented with Clifford gates and ancillas initialized in the* $|0\rangle$ *state* if and only if

$$rank(\{R_i\} \cup \{(f_x)_i\}) = n$$



Synthesizing general circuits

Synthesizing more general circuits

Can we synthesize non-Clifford operators?

Synthesizing more general circuits

Can we synthesize non-Clifford operators?

By inverting the sum-over-paths, we can view **gates as reduction rules**, e.g.,

$$\Lambda_k(X) : |\mathsf{x}\rangle|y \oplus \prod_i x_i\rangle \mapsto |\mathsf{x}\rangle|y\rangle$$
$$\Lambda_k(\theta) : e^{2\pi i\theta \prod_i x_i}|\mathsf{x}\rangle \mapsto |\mathsf{x}\rangle$$
$$H : \frac{1}{\sqrt{2}} \sum_{x' \in \mathbb{Z}_2} (-1)^{xx'}|x'\rangle \mapsto |x\rangle$$

and synthesize by reducing to the identity!

Example: QFT_3 derivation

Derive the circuit by applying re-write rules!

$$QFT_3|x_1x_2x_3\rangle = \frac{1}{\sqrt{2^3}} \sum_{y_1,y_2,y_3} \omega^{x_3y_3} i^{x_3y_2+x_2y_3} (-1)^{x_3y_1+x_2y_2+x_1y_3} |y_1y_2y_3\rangle$$

Example: QFT$_3$ derivation

Derive the circuit by applying re-write rules!

$$QFT_3|x_1x_2x_3\rangle = \frac{1}{\sqrt{2^3}}\sum\nolimits_{y_1,y_2,y_3} \omega^{x_3y_3} i^{x_3y_2+x_2y_3}(-1)^{x_3y_1+x_2y_2+x_1y_3}|y_1y_2y_3\rangle$$

$$\xrightarrow{H_1} \frac{1}{\sqrt{2^2}}\sum\nolimits_{y_1,y_2} \omega^{x_3y_3} i^{x_3y_2+x_2y_3}(-1)^{x_2y_2+x_1y_3}|x_3y_2y_3\rangle$$

Example: QFT$_3$ derivation

Derive the circuit by applying re-write rules!

$$QFT_3|x_1x_2x_3\rangle = \frac{1}{\sqrt{2^3}} \sum_{y_1,y_2,y_3} \omega^{x_3y_3} i^{x_3y_2+x_2y_3}(-1)^{x_3y_1+x_2y_2+x_1y_3}|y_1y_2y_3\rangle$$

$$\xrightarrow{H_1} \frac{1}{\sqrt{2^2}} \sum_{y_1,y_2} \omega^{x_3y_3} i^{x_3y_2+x_2y_3}(-1)^{x_2y_2+x_1y_3}|x_3y_2y_3\rangle$$

$$\xrightarrow{cS^\dagger_{1,2}cT^\dagger_{1,3}} \frac{1}{\sqrt{2^2}} \sum_{y_2,y_3} i^{x_2y_3}(-1)^{x_2y_2+x_1y_3}|x_3y_2y_3\rangle$$

Example: QFT_3 derivation

Derive the circuit by applying re-write rules!

$$
\begin{aligned}
QFT_3|x_1x_2x_3\rangle &= \frac{1}{\sqrt{2^3}} \sum\nolimits_{y_1,y_2,y_3} \omega^{x_3y_3} i^{x_3y_2+x_2y_3} (-1)^{x_3y_1+x_2y_2+x_1y_3} |y_1y_2y_3\rangle \\
&\xrightarrow{H_1} \frac{1}{\sqrt{2^2}} \sum\nolimits_{y_1,y_2} \omega^{x_3y_3} i^{x_3y_2+x_2y_3} (-1)^{x_2y_2+x_1y_3} |x_3y_2y_3\rangle \\
&\xrightarrow{cS^\dagger_{1,2}cT^\dagger_{1,3}} \frac{1}{\sqrt{2^2}} \sum\nolimits_{y_2,y_3} i^{x_2y_3} (-1)^{x_2y_2+x_1y_3} |x_3y_2y_3\rangle \\
&\xrightarrow{H_2} \frac{1}{\sqrt{2}} \sum\nolimits_{y_3} i^{x_2y_3} (-1)^{x_1y_3} |x_3x_2y_3\rangle
\end{aligned}
$$

Example: QFT_3 derivation

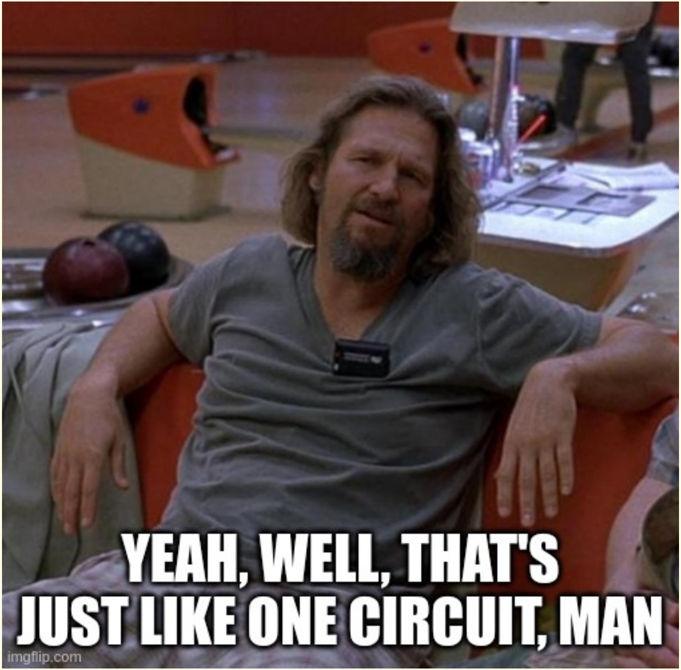
Derive the circuit by applying re-write rules!

$$
\begin{aligned}
QFT_3|x_1x_2x_3\rangle &= \frac{1}{\sqrt{2^3}} \sum_{y_1,y_2,y_3} \omega^{x_3y_3} i^{x_3y_2+x_2y_3} (-1)^{x_3y_1+x_2y_2+x_1y_3} |y_1y_2y_3\rangle \\
&\xrightarrow{H_1} \frac{1}{\sqrt{2^2}} \sum_{y_1,y_2} \omega^{x_3y_3} i^{x_3y_2+x_2y_3} (-1)^{x_2y_2+x_1y_3} |x_3y_2y_3\rangle \\
&\xrightarrow{cS^\dagger_{1,2}cT^\dagger_{1,3}} \frac{1}{\sqrt{2^2}} \sum_{y_2,y_3} i^{x_2y_3} (-1)^{x_2y_2+x_1y_3} |x_3y_2y_3\rangle \\
&\xrightarrow{H_2} \frac{1}{\sqrt{2}} \sum_{y_3} i^{x_2y_3} (-1)^{x_1y_3} |x_3x_2y_3\rangle \\
&\xrightarrow{cS^\dagger_{2,3}} \frac{1}{\sqrt{2}} \sum_{y_3} (-1)^{x_1y_3} |x_3x_2y_3\rangle
\end{aligned}
$$

Example: QFT_3 derivation

Derive the circuit by applying re-write rules!

$$
\begin{aligned}
QFT_3|x_1x_2x_3\rangle =& \frac{1}{\sqrt{2^3}}\sum_{y_1,y_2,y_3}\omega^{x_3y_3}i^{x_3y_2+x_2y_3}(-1)^{x_3y_1+x_2y_2+x_1y_3}|y_1y_2y_3\rangle \\
\xrightarrow{H_1}& \frac{1}{\sqrt{2^2}}\sum_{y_1,y_2}\omega^{x_3y_3}i^{x_3y_2+x_2y_3}(-1)^{x_2y_2+x_1y_3}|x_3y_2y_3\rangle \\
\xrightarrow{cS^\dagger_{1,2}cT^\dagger_{1,3}}& \frac{1}{\sqrt{2^2}}\sum_{y_2,y_3}i^{x_2y_3}(-1)^{x_2y_2+x_1y_3}|x_3y_2y_3\rangle \\
\xrightarrow{H_2}& \frac{1}{\sqrt{2}}\sum_{y_3}i^{x_2y_3}(-1)^{x_1y_3}|x_3x_2y_3\rangle \\
\xrightarrow{cS^\dagger_{2,3}}& \frac{1}{\sqrt{2}}\sum_{y_3}(-1)^{x_1y_3}|x_3x_2y_3\rangle \\
\xrightarrow{H_3}& |x_3x_2x_1\rangle
\end{aligned}
$$


YEAH, WELL, THAT'S
JUST LIKE ONE CIRCUIT, MAN
imgflip.com

Hadamards and permutations

A **generalized permutation** is a permutation matrix times a (unitary) diagonal matrix.

Proposition

Any unitary U can be written as a series of alternating stages of H gates and generalized permutations G

$$U = G_1 H_1 G_2 H_2 G_3 \cdots H_n G_n$$

[7] Kliuchnikov, Maslov, Mosca, *Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates*. QIC 2013.

[8] Giles, Selinger, *Exact synthesis of multiqubit Clifford+T circuits*. Phys. Rev. A 2013.

Hadamards and permutations

A **generalized permutation** is a permutation matrix times a (unitary) diagonal matrix.

Proposition

Any unitary U can be written as a series of alternating stages of H gates and generalized permutations G

$$U = G_1 H_1 G_2 H_2 G_3 \cdots H_n G_n$$

The **Number-Theoretic™** method[7][8] synthesizes unitary matrices by applying a generalized permutation so it can be **reduced** by a Hadamard gate

[7] Kliuchnikov, Maslov, Mosca, *Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates*. QIC 2013.

[8] Giles, Selinger, *Exact synthesis of multiqubit Clifford+T circuits*. Phys. Rev. A 2013.

Symbolic exact synthesis

Recall: A path sum Ψ is (Hadamard) **reducible** if we can apply the rule

$$H : \frac{1}{\sqrt{2}} \sum_{x' \in \mathbb{Z}_2} (-1)^{xx'} |x'\rangle \mapsto |x\rangle$$

Symbolic exact synthesis

Recall: A path sum Ψ is (Hadamard) **reducible** if we can apply the rule

$$H : \frac{1}{\sqrt{2}} \sum_{x' \in \mathbb{Z}_2} (-1)^{xx'} |x'\rangle \mapsto |x\rangle$$

Algorithm:

1. Simplify $|\Psi\rangle$ using [E], [I], [U]
2. If $\exists$ **generalized permutation** G s.t. $G^\dagger|\Psi\rangle$ is reducible,
   2.1 $|\Psi\rangle \leftarrow (\mathrm{H} \otimes \mathrm{I}_{n-1}) G^\dagger |\Psi\rangle$
   2.2 Go to 1.
3. If path variables remain or Ψ is non-unitary, fail

How do we find G? Does there always exist such a G?

A heuristic for generalized permutations

1. Apply affine simplifications to the output state $|f(\mathrm{x},\mathrm{y})\rangle$
2. Apply non-linear simplifications to the phase $e^{2\pi i P(\mathrm{x},\mathrm{y})}$
3. Apply non-linear simplifications to the output state $|f'(\mathrm{x},\mathrm{y})\rangle$
4. Apply non-linear simplifications to the phase $e^{2\pi i P'(\mathrm{x},\mathrm{y})}$
5. **If the path sum is still irreducible, find simultaneous variable substitutions to make it reducible**
    - Heuristic: degree reduction

See the paper for more details!

Performance

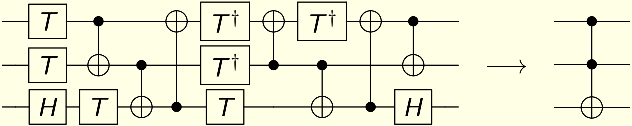
Re-synthesizing random Clifford and Clifford+T circuits:

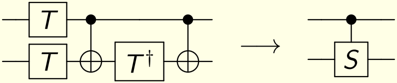
	n	# gates	# circuits	avg. time (s)	avg. change	success
Clifford	20	500	1000	0.137	+19.2%	–
	20	1000	1000	0.481	-12.9%	–
	50	500	1000	0.264	+90.7%	–
	50	1000	1000	1.518	+129.1%	–
Clifford+T	20	100	1000	0.010	+48.9%	99.9%
	20	200	1000	0.045	+93.7%	94.9%
	20	300	1000	0.097	+115.9%	74.7%
	50	100	1000	0.016	+33.5%	100.0%
	50	200	1000	0.044	+49.0%	100.0%
	50	300	1000	0.104	+79.4%	99.6%

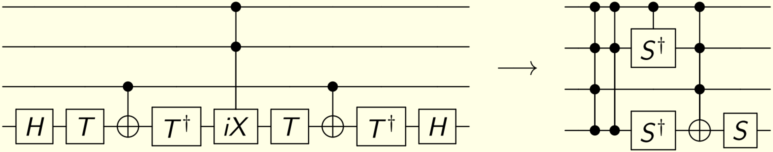
Clifford+T heuristic fails more as gate density increases

Applications

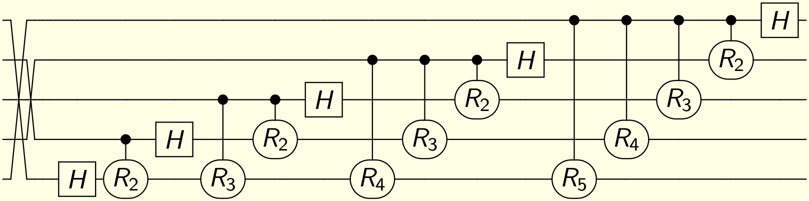
Decompiling from a low-level gate set (e.g. Clifford+T) to H + generalized permutations **often reveals high-level structure!**







QFT, synthesized



Compiled with Feynman: https://github.com/meamy/feynman

Conclusion

In this talk...

- Unitarity testing is **coNP**-hard
- Normal forms & extraction of an 8-stage Clifford circuit
- Partial heuristic for general circuit extraction

Future work

- Use completions of path sum re-writing to prove completeness of our synthesis framework
- Come up with complete procedure for finding a reducing generalized permutation
- Explore use in peephole re-synthesis

Thank you!