Symbolic representation of quantum computations

Matthew Amy

Simon Fraser University

Algebraic Structures in Quantum Computation
UBC, June 14, 2022

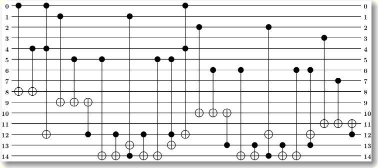
What is this talk about?

How can computers (and humans) express, reason *about, and* understand *quantum computations?*

Representations of quantum computations

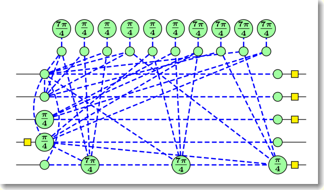
Standard representations are based around **composing linear operators**

- Circuits - composing unitary matrices (+ measurement)
- Quantum programming languages - complicated classical code describing a circuit
- ZX-calculus - tensor networks over a particular basis




```
%Array = type opaque
%Result = type opaque
%Qubit = type opaque

declare %Result* @__quantum__qis__mz(%Qubit* %0)
declare void @__quantum__qis__cnot(%Qubit* %0, %Qubit* %1)
declare void @__quantum__qis__h(%Qubit* %0)

declare %Array* @__quantum__rt__qubit_allocate_array(i64 %0)
declare void @__quantum__rt__qubit_release_array(%Array* %0)
declare i8* @__quantum__rt__array_get_element_ptr_1d(%Array* %0, i64 %1)

define i32 @main(i32 %0, i8** %1)  {
  %4 = call %Array* @__quantum__rt__qubit_allocate_array(i64 2)
  %5 = call i8* @__quantum__rt__array_get_element_ptr_1d(%Array* %4, i64 0)
  %6 = bitcast i8* %5 to %Qubit**
  %7 = load %Qubit*, %Qubit** %6, align 8
  call void @__quantum__qis__h(%Qubit* %7)
  %8 = call i8* @__quantum__rt__array_get_element_ptr_1d(%Array* %4, i64 1)
  %9 = bitcast i8* %8 to %Qubit**
  %10 = load %Qubit*, %Qubit** %9, align 8
  call void @__quantum__qis__cnot(%Qubit* %7, %Qubit* %10)
  %11 = call %Result* @__quantum__qis__mz(%Qubit* %7)
  %12 = call %Result* @__quantum__qis__mz(%Qubit* %10)
  call void @__quantum__rt__qubit_release_array(%Array* %4)
  ret i32 0
}
```

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[5];

ccx q[0],q[1],q[4];
ccx q[2],q[4],q[3];
ccx q[0],q[1],q[4];
```



Reasoning about quantum programs

Ideally, we would like a computer to know **what a quantum program does**

At the very least, it should know **when two programs are equivalent** for the purposes of

- Verification
- Compilation
- Optimization

```
%a1, %b1 = qs.CX %a0, %b0
%a2, %b2 = qs.CX %a1, %b1
%a3 = qs.H %a2
```

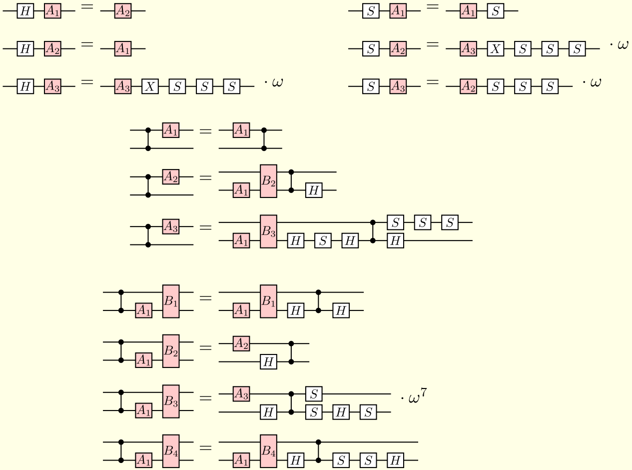
$\longrightarrow$

```


%a3 = qs.H %a0
```

Ittah et al., Enabling Dataflow Optimization for Quantum Programs. ACM Trans. Quant. Comput. (2022).

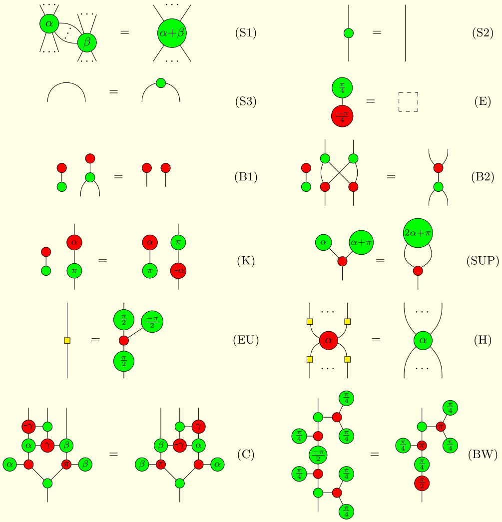
Circuit reasoning

$$\omega^8 = 1$$



Selinger, Generators and relations for n-qubit Clifford operators. LMCS 2015.

ZX reasoning



Jeandel, Perdrix and Vilmart, A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics. LICS 2018.

Models of classical computation

Combinatory logic

- Syntax: Point-free compositions of operators over a basis (e.g. **circuits**)
- Reasoning: Equational (e.g. by **re-write rules**)

Models of classical computation

Combinatory logic

- Syntax: Point-free compositions of operators over a basis (e.g. **circuits**)
- Reasoning: Equational (e.g. by **re-write rules**)

Lambda calculus

- Syntax: Functions of symbolic inputs
- Reasoning: Computational (e.g. by **reduction rules**)

Models of classical computation

Combinatory logic

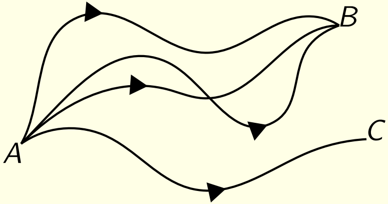
- Syntax: Point-free compositions of operators over a basis (e.g. **circuits**)
- Reasoning: Equational (e.g. by **re-write rules**)

Lambda calculus

- Syntax: Functions of symbolic inputs
- Reasoning: Computational (e.g. by **reduction rules**)

Is there a more computational model of QC?

Enter the symbolic sum-over-paths



A **symbolic representation** of discrete path integrals which is:

- Efficiently computable
- Universal for qubit quantum mechanics
- Re-writing has a **computational** interpretation as reducing or contracting sets of interfering paths
  - Highly automatable!
  - Implemented in FEYNMAN (`https://github.com/meamy/feynman`)

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

Overview

1. The sum-over-paths
2. Reasoning with symbolic sums
3. Applications

The sum-over-paths

The linear-algebraic view

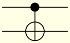
A (pure) state of n qubits is a unit vector in $\mathbb{C}^{2^n}$ which can be described as superposition of classical states

$$|\psi\rangle = \sum\nolimits_{\mathsf{x}\in\mathbb{Z}_2^n} \alpha_{\mathsf{x}}|\mathsf{x}\rangle, \qquad \mathsf{x} \in \{0,1\}^n = \mathbb{Z}_2^n$$

Computations change the state by applying unitary transformations to them

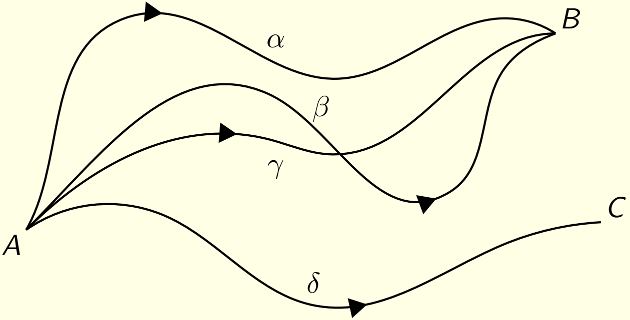
$$X = -\boxed{X}- = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad H = -\boxed{H}- = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$S = -\boxed{S}- = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad T = -\boxed{T}- = \begin{bmatrix} 1 & 0 \\ 0 & \omega = e^{i\frac{\pi}{4}} \end{bmatrix}$$

CNOT =  $= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

The path integral view

The amplitude of a classical state is the **sum-over-all-paths leading to it**

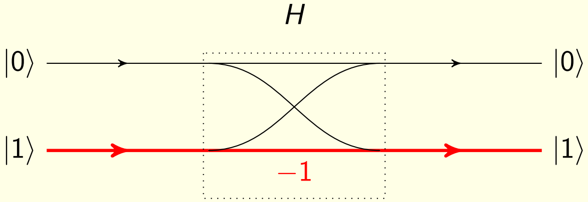


Computations change the state by sending classical states along various paths to new states

$$A \mapsto (\alpha + \beta + \gamma)B + \delta C$$

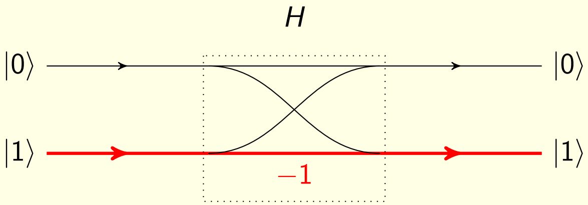
Gates as symbolic sums

The hadamard gate H **branches** on a **classical** value in superposition with equal weight $\frac{1}{\sqrt{2}}$ and varying phase



Gates as symbolic sums

The hadamard gate H **branches** on a **classical** value in superposition with equal weight $\frac{1}{\sqrt{2}}$ and varying phase

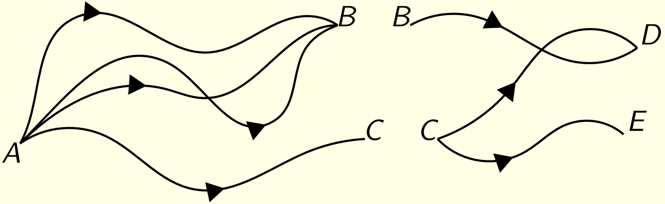


We can write this action symbolically as a **sum-over-paths**:

$$H : |x\rangle \mapsto \frac{1}{\sqrt{2}} \sum_{y \in \mathbb{Z}_2} (-1)^{xy} |y\rangle \text{ for } x \in \mathbb{Z}_2$$

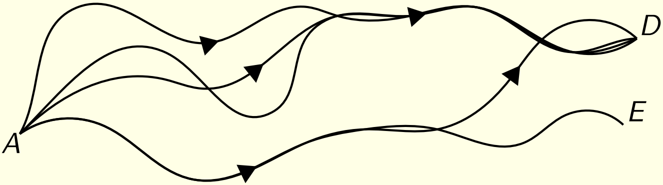
Composition

Computations can be composed by composing paths through the same intermediate states



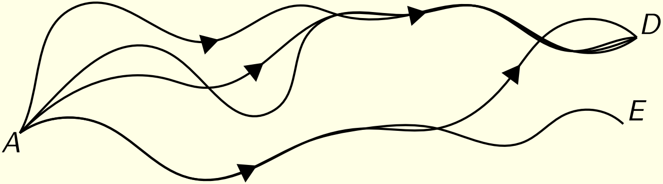
Composition

Computations can be composed by composing paths through the same intermediate states



Composition

Computations can be composed by composing paths through the same intermediate states



Symbolically, corresponds to an **encoding of matrix multiplication**

$$
\begin{aligned}
HH : |x\rangle &\mapsto \frac{1}{\sqrt{2}} \sum\nolimits_{y \in \mathbb{Z}_2} (-1)^{xy} H|y\rangle \\
&\mapsto \frac{1}{\sqrt{2}} \sum\nolimits_{y \in \mathbb{Z}_2} (-1)^{xy} \left(\frac{1}{\sqrt{2}} \sum\nolimits_{z \in \mathbb{Z}_2} (-1)^{yz} |z\rangle \right) \\
&\mapsto \frac{1}{2} \sum\nolimits_{y,z} (-1)^{xy+yz} |z\rangle
\end{aligned}
$$

Historical complexity applications

$$H : |x\rangle \mapsto \frac{1}{\sqrt{2}} \sum_{y \in \mathbb{Z}_2} (-1)^{xy} |y\rangle \qquad CCZ : |xyz\rangle \mapsto (-1)^{xyz} |xyz\rangle$$

The general form of a path sum over $\{H, CCZ\}$

$$|\mathrm{x}\rangle \mapsto \frac{1}{\sqrt{2}^{k+n}} \sum_{\mathrm{y} \in \mathbb{Z}_2^k} \sum_{\mathrm{x}' \in \mathbb{Z}_2^n} (-1)^{f(\mathrm{x},\mathrm{y},\mathrm{x}')} |\mathrm{x}'\rangle$$

where $\deg(f) \leq 3$.

Theorem (Ehrenfeucht & Karpinski)

Counting the 0's of $f \in \mathbb{Z}_2[x_1, \ldots]$ with degree ≥ 3 is #P-hard

Corollary: $\mathbf{BQP} \subseteq \mathbf{P}^{\#\mathbf{P}}$

Montanaro, Quantum circuits and low-degree polynomials over $\mathbb{Z}_2$. J Phys A: Math Theor, 2017.

Historical complexity applications

$$H : |x\rangle \mapsto \frac{1}{\sqrt{2}} \sum_{y \in \mathbb{Z}_2} (-1)^{xy} |y\rangle \qquad CZ : |xy\rangle \mapsto (-1)^{xy} |xy\rangle$$

The general form of a path sum over $\{H, CZ\}$ is

$$|\mathrm{x}\rangle \mapsto \frac{1}{\sqrt{2}^{k+n}} \sum_{\mathrm{y} \in \mathbb{Z}_2^k} \sum_{\mathrm{x}' \in \mathbb{Z}_2^n} (-1)^{f(\mathrm{x},\mathrm{y},\mathrm{x}')} |\mathrm{x}'\rangle$$

where $\deg(f) \leq 2$.

Theorem (Ehrenfeucht & Karpinski)

Counting the 0*'s of* $f \in \mathbb{Z}_2[x_1, \ldots]$ *with degree* ≤ 2 *is in P*

Corollary: $\{H, CZ\}$ **can be simulated in polynomial time**

Montanaro, Quantum circuits and low-degree polynomials over $\mathbb{Z}_2$. J Phys A: Math Theor, 2017.

Formalizing the sum-over-paths

Definition

A **(balanced) sum-over-paths** from $\mathbb{C}^{2^n}$ to $\mathbb{C}^{2^m}$ is a map

$$|x_1 \cdots x_n\rangle \mapsto \mathcal{N} \sum_{y_1,\ldots,y_k \in \mathbb{Z}_2} e^{2\pi i P(\mathrm{x},\mathrm{y})} |f_1(\mathrm{x},\mathrm{y}) \cdots f_m(\mathrm{x},\mathrm{y})\rangle$$

defined by

- a scalar $\mathcal{N} \in \mathbb{C}$,
- a **phase polynomial** $P \in \mathbb{R}[x_1, \ldots, x_n, y_1, \ldots y_k]$, and
- outputs $f_i \in \mathbb{Z}_2[x_1, \ldots, x_n, y_1, \ldots y_k]$

Formalizing the sum-over-paths

Definition

A **(balanced) sum-over-paths** from $\mathbb{C}^{2^n}$ to $\mathbb{C}^{2^m}$ is a map

$$|x_1 \cdots x_n\rangle \mapsto \mathcal{N} \sum_{y_1,\ldots,y_k \in \mathbb{Z}_2} e^{2\pi i P(\mathrm{x},\mathrm{y})} |f_1(\mathrm{x},\mathrm{y}) \cdots f_m(\mathrm{x},\mathrm{y})\rangle$$

defined by

- a scalar $\mathcal{N} \in \mathbb{C}$,
- a **phase polynomial** $P \in \mathbb{R}[x_1, \ldots, x_n, y_1, \ldots y_k]$, and
- outputs $f_i \in \mathbb{Z}_2[x_1, \ldots, x_n, y_1, \ldots y_k]$

We can also consider **unbalanced** sums of the form

$$|x_1 \cdots x_n\rangle \mapsto \mathcal{N} \sum_{y_1,\ldots,y_k \in \mathbb{Z}_2} \alpha_1^{P_1(\mathrm{x},\mathrm{y})} \alpha_2^{P_2(\mathrm{x},\mathrm{y})} \cdots |f_1(\mathrm{x},\mathrm{y}) \cdots f_m(\mathrm{x},\mathrm{y})\rangle$$

The sum-over-paths as a model for circuits

Balanced sums form a **symmetric monoidal category**, so we can define the circuit SOP $[\![C]\!]$ **compositionally** by giving interpretations of each basis gate:

$$[\![H]\!] = |x\rangle \mapsto \frac{1}{\sqrt{2}}\sum\nolimits_y (-1)^{xy}|y\rangle$$
$$[\![T]\!] = |x\rangle \mapsto \omega^x|x\rangle$$
$$[\![\mathrm{CNOT}]\!] = |x_1x_2\rangle \mapsto |x_1(x_1 \oplus x_2)\rangle$$

$$[\![U_2U_1]\!] = [\![U_2]\!] \circ [\![U_1]\!]$$
$$[\![U_1 \otimes U_2]\!] = [\![U_1]\!] \otimes [\![U_2]\!]$$

Proposition

For any fixed k, the discrete path integral of a circuit over Clifford $+ R_k := diag(1, e^{2\pi i/2^k})$ is poly-time and poly-space computable

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

Universality

Balanced sums can be further equipped with the structure of a **dagger compact category** via a unit, counit, and dagger

$$[\![\eta]\!] = \sum_y |yy\rangle$$

$$[\![\epsilon]\!] = |x_1 x_2\rangle \mapsto \sum_y (-1)^{x_1 y + y x_2}$$

$$[\![U^\dagger]\!] = [\![U]\!]^\dagger$$

Theorem

Any linear operator between even-power dimensional complex vector spaces can be represented as a balanced sum-over-paths

- $\langle\psi| := |\psi\rangle^\dagger$
- $tr_A(U) := (\epsilon \otimes I) \circ (I \otimes U) \circ (\eta \otimes I)$
- $meas := (SWAP \otimes \epsilon) \circ (I \otimes \chi_{meas} \otimes I) \circ (\eta \otimes SWAP)$

Vilmart, The Structure of Sum-Over-Paths, its Consequences, and Completeness for Clifford. FoSSaCS 2021.

Reasoning with symbolic sums

Rewriting sums

The sum-over-paths representation **encodes** matrix multiplication **symbolically**

$$[\![I]\!] = |x\rangle \mapsto |x\rangle$$
$$[\![HH]\!] = |x\rangle \mapsto \frac{1}{2}\sum_{y,z\in\mathbb{Z}_2}(-1)^{xy+yz}|z\rangle$$

This allows the sum to be **simplified** without explicit evaluation!

Interference patterns

To see how the *HH* sum

$$|x\rangle \mapsto \frac{1}{2} \sum_{y,z \in \mathbb{Z}_2} (-1)^{xy+yz} |z\rangle$$

interferes, we can expand it out.

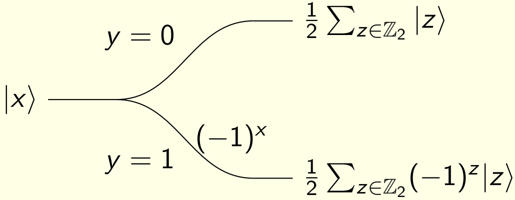
$|x\rangle$ ———

Interference patterns

To see how the HH sum

$$|x\rangle \mapsto \frac{1}{2}\sum_{y,z\in\mathbb{Z}_2}(-1)^{xy+yz}|z\rangle$$

interferes, we can expand it out.

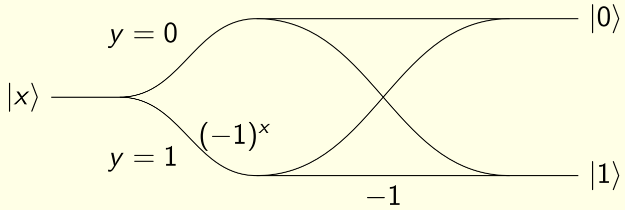


Interference patterns

To see how the HH sum

$$|x\rangle \mapsto \frac{1}{2}\sum_{y,z\in\mathbb{Z}_2}(-1)^{xy+yz}|z\rangle$$

interferes, we can expand it out.

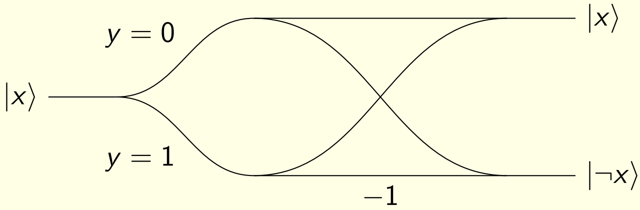


Interference patterns

If we sum over $z \in \{x, \neg x\} = \mathbb{Z}_2$ instead,

$$|x\rangle \mapsto \frac{1}{2} \sum_{y \in \mathbb{Z}_2, z \in \{x, \neg x\}} (-1)^{xy+yz} |z\rangle$$

we get a simple pattern

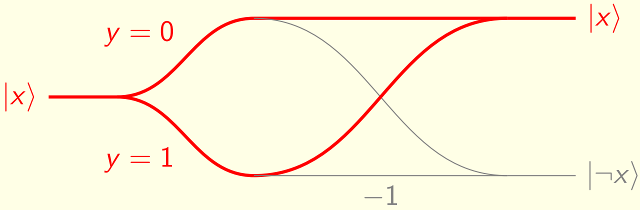


Interference patterns

If we sum over $z \in \{x, \neg x\} = \mathbb{Z}_2$ instead,

$$|x\rangle \mapsto \frac{1}{2} \sum_{y \in \mathbb{Z}_2, z \in \{x, \neg x\}} (-1)^{xy+yz} |z\rangle$$

we get a simple pattern

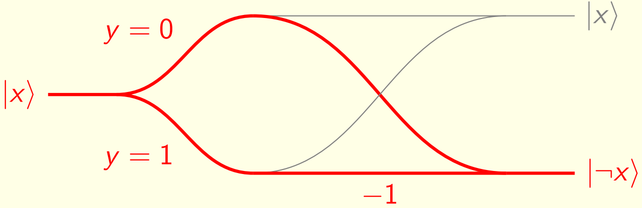


Interference patterns

If we sum over $z \in \{x, \neg x\} = \mathbb{Z}_2$ instead,

$$|x\rangle \mapsto \frac{1}{2} \sum\nolimits_{y \in \mathbb{Z}_2, z \in \{x, \neg x\}} (-1)^{xy+yz} |z\rangle$$

we get a simple pattern



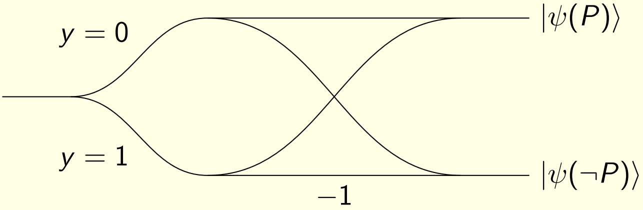
Generalization

Lemma

For any Boolean-valued expression P

$$\sum_{y,z}(-1)^{zy+yP}|\psi(z)\rangle = 2|\psi(P)\rangle$$

In particular **only the paths where $z = P$ survive**



Reduction rules for sum-over-paths

$$\sum_y |\psi\rangle \quad\longleftrightarrow\quad 2|\psi\rangle \qquad [E]$$

$$\sum_{y,z}(-1)^{zy+yP}|\psi(z)\rangle \quad\longleftrightarrow\quad 2|\psi(P)\rangle \qquad [H]$$

$$\sum_y i^y(-1)^{yP}|\psi\rangle \quad\longleftrightarrow\quad \sqrt{2}\omega i^{3P}|\psi\rangle \qquad [\omega]$$

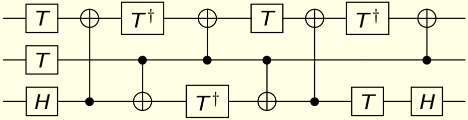
$$\sum_{y,z}\alpha^{xP(y)}\beta^{\neg xQ(z)}|\psi\rangle \quad\longleftrightarrow\quad \sum_y \alpha^{xP(y)}\beta^{\neg xQ(y)}|\psi\rangle \qquad [O]$$

$$\sum_y(\alpha^y\beta^{\neg y})^P|\psi\rangle \quad\longleftrightarrow\quad 2(\frac{\alpha+\beta}{2})^P|\psi\rangle \qquad [A]$$

- **[E], [H], [ω]** complete (and poly-time) for **Clifford** circuits
- **[E], [H], [ω], [O], [A]** complete for arbitrary linear operators
- Vilmart 2022: a complete re-write system for Clifford+R_k that **stays in the balanced fragment**

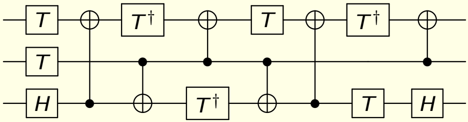
Reasoning symbolically

What computation does this circuit perform?



Reasoning symbolically

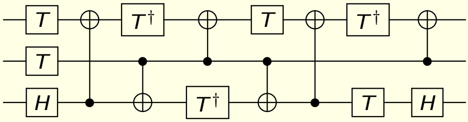
What computation does this circuit perform?



$$|x_1 x_2 x_3\rangle \mapsto \frac{1}{2} \sum_{y_1, y_2 \in \mathbb{Z}_2} (-1)^{x_3 y_1 + x_1 x_2 y_1 + y_1 y_2} |x_1 x_2 y_2\rangle$$

Reasoning symbolically

What computation does this circuit perform?

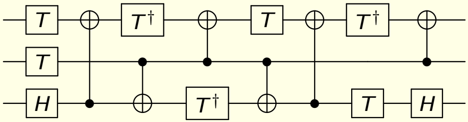


$$|x_1x_2x_3\rangle \mapsto \frac{1}{2}\sum_{y_1,y_2\in\mathbb{Z}_2}(-1)^{x_3y_1+x_1x_2y_1+y_1y_2}|x_1x_2y_2\rangle$$

$$\mapsto \frac{1}{2}\sum_{y_1,y_2\in\mathbb{Z}_2}(-1)^{y_2y_1+y_1(x_3+x_1x_2)}|x_1x_2y_2\rangle$$

Reasoning symbolically

What computation does this circuit perform?



$$|x_1x_2x_3\rangle \mapsto \frac{1}{2}\sum_{y_1,y_2\in\mathbb{Z}_2}(-1)^{x_3y_1+x_1x_2y_1+y_1y_2}|x_1x_2y_2\rangle$$

$$\mapsto \frac{1}{2}\sum_{y_1,y_2\in\mathbb{Z}_2}(-1)^{y_2y_1+y_1(x_3+x_1x_2)}|x_1x_2y_2\rangle$$

$$\mapsto |x_1x_2(x_3\oplus x_1x_2)\rangle \qquad [\mathrm{H},\ y_2\leftarrow x_3\oplus x_1x_2]$$

Applications: Verification & simulation

Why verify?

- **High** degree of uncertainty about the correctness of estimates
- Bugs!

[a]Our simulation found an error in the circuit optimized by T-par. Specifically, the circuit maps $|1024\rangle \mapsto \frac{|1025\rangle+|1030\rangle+|1161\rangle+|1166\rangle}{2}$ whereas it is supposed to perform the mapping $|1024\rangle \mapsto |1088\rangle$.

[b]Note that our software reduced the T-count of the original pre-optimization circuit used by T-par to 0. It turned out that the circuit used by T-par is incorrect. In our optimization reported in this table, we used the correct original circuit [41, Figure 5].

qcla-mod$_7$ [27]	26	413	235[a]	NRSCM	237	**216**
rc-adder$_6$ [27]	14	77	47	RM$_{m\&r}$	47	47
vbe-adder$_3$ [27]	10	70	24	Tpar	24	24

Table 1: Benchmark circuits. The columns n and T contain the amount of qubits and T gates in the original circuit. *Best prev.* is the previous best-known ancilla-free T-count for that circuit and *Method* specifies which method was used: *RM$_m$* and *RM$_r$* refer to the *maximum* and *recursive* Reed-Muller decoder of Ref. [8], *Tpar* is Ref. [6], *TODD* is Ref. [21] and *NRSCM* refers to Ref. [28]. *PyZX* and *PyZX + TODD* specify the T-counts produced by the procedures outlined in the Methods section. Numbers shown in bold are better than previous best, and italics are worse. The superscript (a) indicates an error in a previously reported T-count. The error was found using Amy's Feynman tool [4].

[4]It was reported in [2] that 4 separate optimized benchmark circuits (CSLA-MUX_3, Adder_8, Mod-Mult_55 and GF(2^{32})) provably contained errors. These errors have since been fixed and these circuits now pass verification.

Nam, Ross, Su, Childs, Maslov, *Automated optimization of large quantum circuits with continuous parameters*. npj:Quantum Information 4, 23 (2018)

Amy, Azimzadeh, Mosca, *On the CNOT-complexity of CNOT-PHASE circuits*. Quantum Science and Technology 4(1), 2018

van de Wetering, Kissinger, *Reducing T-count with the ZX-calculus*. Phys. Rev. A 102, 022406 (2020)

Automated verification in practice

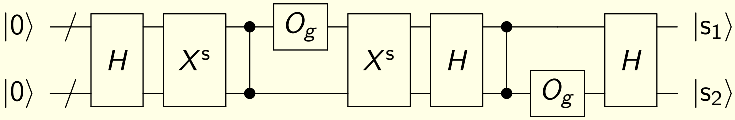
Through **purely automated re-writing**

- Circuits verified against **logical specifications** [A18] (below)
- Optimization results verified in academic papers
  - [AAM18, dBBW19, dBBW20, vdWK20, AR21]
  - Bugs also found in these!
- State & T gate teleportation channels verified
- Measurement-assisted uncomputation circuits verified for up to 128 qubits in [AR12]

Circuit	Qubits	Variables	Gates	Time (s)
Toffoli_{50}	97	190	1520	1.078
Toffoli_{100}	**197**	390	3120	**5.346**
Maslov_{50}	74	192	865	0.759
Maslov_{100}	149	392	1765	3.937
Adder_{8}	40	56	530	0.142
Adder_{16}	80	120	1130	26.151
QFT_{16}	16	16	616	1.250
QFT_{31}	31	31	2356	16.929

Hidden shift algorithm

Simulation of **the entire output distribution** for the popular Maiorana-McFarland Hidden Shift benchmark circuits via only automated re-writing



- **5s** vs **28h** for stabilizer decompositions[1]
- **1m** on a **tablet computer** for 1400 T-count, $> 10,000$ gate instances used recently in graphical simulation[2]

[1]S. Bravyi et al., *Simulation of quantum circuits by low-rank stabilizer decompositions.* Quantum 3, 181 (2019).

[2]A. Kissinger, J. van de Wetering, R. Vilmart, Classical simulation of quantum circuits with partial and graphical stabiliser decompositions. QPL 2022.

Applications:
Synthesis & Optimization

Clifford circuits

Over $\{H, CNOT, S\}$, path sums have the form

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

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

Amy, Bennett-Gibbs, Ross, Symbolic synthesis of Clifford circuits and beyond. QPL 2022.

Clifford circuits

Over $\{H, CNOT, S\}$, path sums have the form

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

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

Proposition (Affine normal form)

Any Clifford circuit can be written, up to a re-ordering of the qubits, using the re-write rules [E], [H], [ω] in polynomial time as

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

Corollary: **Efficient simulation of Clifford circuits**

Amy, Bennett-Gibbs, Ross, Symbolic synthesis of Clifford circuits and beyond. QPL 2022.

Decomposition into linear operators

The affine normal form

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

decomposes into the following sequence of linear operators

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

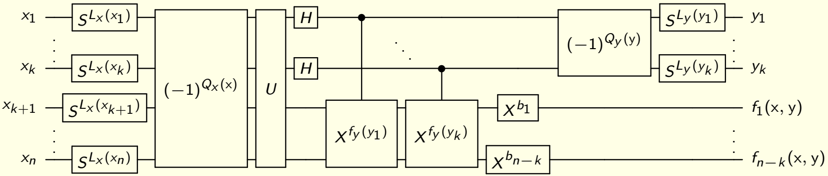
Amy, Bennett-Gibbs, Ross, Symbolic synthesis of Clifford circuits and beyond. QPL 2022.

A constructive proof of the Bruhat decomposition

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*

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



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

Amy, Bennett-Gibbs, Ross, Symbolic synthesis of Clifford circuits and beyond. QPL 2022.

Synthesizing more general circuits

Can we synthesize non-Clifford operators?

Amy, Bennett-Gibbs, Ross, Symbolic synthesis of Clifford circuits and beyond. QPL 2022.

Synthesizing more general circuits

Can we synthesize non-Clifford operators?

By **inverting** the sum-over-paths actions, we get an augmented re-write system with side effects

$$\Lambda_k(X) : |\mathsf{x}\rangle|y \oplus \prod_i x_i\rangle \mapsto |\mathsf{x}\rangle|y\rangle$$

$$\Lambda_k(\theta) : e^{2\pi i\theta \prod_i x_i}|\mathsf{x}\rangle \mapsto |\mathsf{x}\rangle$$

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

Synthesize by reducing to the identity!

Amy, Bennett-Gibbs, Ross, Symbolic synthesis of Clifford circuits and beyond. QPL 2022.

Synthesizing the QFT

The n-bit **QFT** is given by the following matrix

$$\frac{1}{\sqrt{2^n}}\begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_{2^n} & \omega_{2^n}^2 & \omega_{2^n}^3 & \cdots & \omega_{2^n}^{2^n-1} \\ 1 & \omega_{2^n}^2 & \omega_{2^n}^4 & \omega_{2^n}^6 & \cdots & \omega_{2^n}^{2(2^n-1)} \\ 1 & \omega_{2^n}^3 & \omega_{2^n}^6 & \omega_{2^n}^9 & \cdots & \omega_{2^n}^{3(2^n-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_{2^n}^{2^n-1} & \omega_{2^n}^{2(2^n-1)} & \omega_{2^n}^{3(2^n-1)} & \cdots & \omega_{2^n}^{(2^n-1)(2^n-1)} \end{bmatrix}$$

As a sum-over-paths,

$$QFT_n : |\mathsf{x}\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{\mathsf{y}\in\mathbb{Z}_2^n} \omega_{2^n}^{\mathsf{x}\cdot\mathsf{y}} |\mathsf{y}\rangle$$

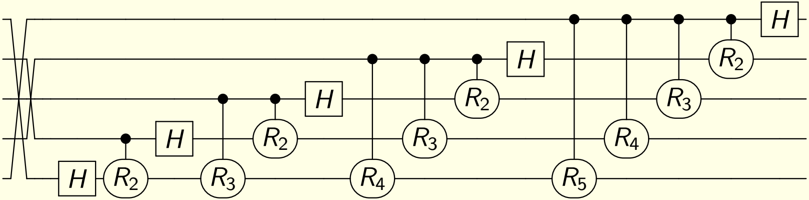
QFT_3 derivation

Completely automated circuit derivation

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

QFT, synthesized

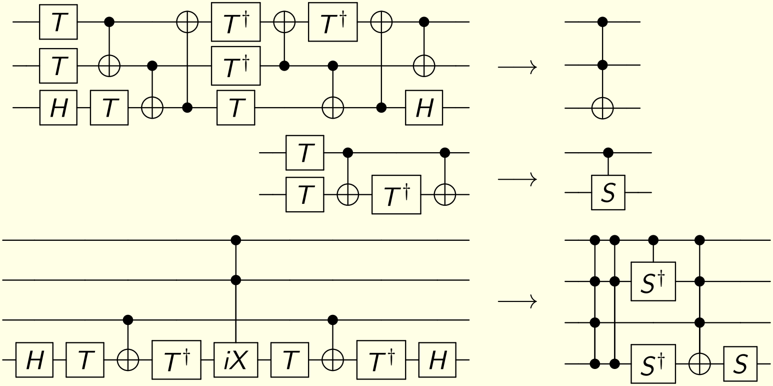
Compiled with FEYNMAN:



https://github.com/meamy/feynman

Decompilation

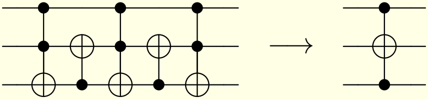
Sum-over-paths synthesis can be used to **decompile** from a low-level gate set to a high-level one

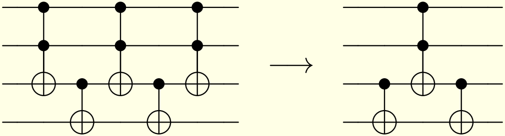


Reveals useful semantic information about a circuit!

Circuit optimization

(Future work) Can re-synthesis be used to optimize quantum circuits?





Conclusion

In this talk...

- Symbolic sums as a representation for quantum circuits, channels, and generally mechanics
- Reasoning with symbolic sums
- Applications to simulation, verification, synthesis and circuit optimization

Thank you!