Quantum computation and compilation

Matthew Amy

Simon Fraser University

CS Undergraduate Research Symposium
April 11, 2022
Computation

Computation is a physical process

We use abstractions to describe and model computation
- 0 for low voltage, 1 for high voltage
- Turing machines

The (extended) Church–Turing thesis:
A probabilistic Turing machine can efficiently simulate any physical model of computation.
Quantum computation

Classical models of computation are based on classical physics.

Richard Feynman, 1982:

A classical computer cannot efficiently simulate a quantum mechanical system.

Subsequent algorithms using quantum effects for speed-ups:

- (Shor, 1994) Integer factorization
- (Lloyd, 1996) Simulation of quantum systems
- (Grover, 1996) Unstructured search
- Discrete logarithms, linear systems, knot invariants, ...
Beam-splitters

A beam-splitter acts as a classical **coin flip**: a photon traveling through it will either
- continue straight through, or
- be reflected

with **equal probability**.
The beam-splitter experiment

Where will a single photon be detected?

- Classical intuition says equal probability at either location
The beam-splitter experiment

Experimentally it will always appear at the lower detector

- Intuition is that the photon took *both paths simultaneously*
- **Interference** causes paths to the upper detector to cancel
How do we model this abstractly?
The linear algebraic model

We map the classical states to a basis of $\mathbb{C}^2$

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The state of a qubit is a unit vector $|\psi\rangle \in \mathbb{C}^2$, corresponding to a superposition of the classical 0 and 1 states

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1$$
Transformations on a quantum state are **unitary** operators $U : \mathbb{C}^2 \rightarrow \mathbb{C}^2$ called **gates**

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Gates transform states via matrix multiplication

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{-1}{\sqrt{2}}|1\rangle$$
Quantum circuits

Large unitaries are built by composing gates in circuits

\[
\begin{align*}
T & \quad T^\dagger & \quad T & \quad T^\dagger \\
T & \quad T^\dagger & \quad H & \quad T^\dagger \\
H & \quad T^\dagger & \quad T & \quad H
\end{align*}
\]

Time

Common gates:

\[
S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad 
H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad 
CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad 
T = \begin{bmatrix} 1 & 0 & \omega := e^{i\pi/4} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]
Measurement

When we measure a qubit in a superposition

\[ |\psi\rangle = \alpha |0\rangle + \beta |1\rangle \]

the state collapses to

- \[|0\rangle\] with probability \(|\alpha|^2\)
- \[|1\rangle\] with probability \(|\beta|^2\)

The measurement probabilities form a **probability distribution**, forcing \(|\psi\rangle\) to be a unit vector:

\[ |\alpha|^2 + |\beta|^2 = 1 \]
Quantum programs

- States: $x \in \{0, 1\}^n = \mathbb{Z}_2^n$
- Functions: $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$
- States: $|\psi\rangle \in \mathbb{C}^{2^n}$
- Functions: $U : \mathbb{C}^{2^n} \rightarrow \mathbb{C}^{2^n}$

Typical quantum program:

1. Apply some circuit to the $|00\ldots0\rangle$ state
2. Measure some or all of the qubits
3. Process the results on a classical computer
How do we program & compile circuits?
Quantum compilation

Circuit over high-level gates & functions

Circuit over mid-level logical gates

Circuit over error corrected logical gates

Circuit over physical gates

Compile by replacing gates with lower-level circuits, e.g.,

\[
\begin{align*}
S & = T \\
& \quad \quad T \quad T^\dagger
\end{align*}
\]
Gate sets

A set of gates is **universal** if it can **approximate** any unitary up to arbitrary accuracy.

**Theorem (The Solovay-Kitaev theorem)**

*Given a set $G$ of gates which is dense in $SU(2)$, any $U \in SU(2)$ can be approximated to within $\epsilon$ error using a poly-logarithmic number of gates taken from $G$.*

$\{H, \text{CNOT}, T\}$ **is the standard error corrected universal set**
Algebraic compilation = compile using algebraic characterizations.

The **number-theoretic method:**
Let $\mathbb{D} = \{a/2^k | a, k \in \mathbb{Z}\}$ be the ring of Dyadic fractions and let

$$\mathbb{D}[\omega] = \{a\omega^3 + b\omega^2 + c\omega + d \mid a, b, c, d \in \mathbb{D}\}$$

where $\omega = e^{\pi i/4}$.

*(Kliuchnikov et al. 2013, Giles & Selinger 2013):*  
A $2^n \times 2^n$ unitary matrix $U$ can be written as a product of \{H, CNOT, T\} gates if and only if $U$ has entries in $\mathbb{D}[\omega]$.

*(Amy, Glaudell, & Ross 2020):*  
Similar characterizations for $\mathbb{D}, \mathbb{D}[\sqrt{2}], \mathbb{D}[i\sqrt{2}], \text{and } \mathbb{D}[i]$
Let $\mathcal{R}$ be a ring and $\mathcal{R}[\alpha]$ be an **algebraic** extension of $\mathcal{R}$.

**(Amy, Glaudell, Ross, et al. 2022):**

*If there exists a pseudo-companion matrix $\Gamma \in M_{k \times k}(\mathcal{R})$ for $\alpha$, then any $n \times n$ unitary $U \in M_{k \times k}(\mathcal{R}[\alpha])$ can be embedded over $\mathcal{R}$ with a suitable resource state.*

```
\begin{array}{c}
\vdots \\
| \lambda \rangle \\
\vdots \\
\end{array}
\begin{array}{c}
U \\
\phi(U) \\
\end{array}
\begin{array}{c}
\vdots \\
| \lambda \rangle \\
\vdots \\
\end{array}
= \\
\begin{array}{c}
\vdots \\
| \lambda \rangle \\
\vdots \\
\end{array}
```
Circuits over restricted or non-universal gate sets are often easier to efficiently characterize & compile.

(Amy, Maslov, & Mosca, 2014)

Any n-qubit circuit over \{CNOT, X, T\} can be written as

$$U|x\rangle = \omega^{P(x)}|Ax\rangle$$

where $$A \in \mathcal{M}_{n \times n}(\mathbb{Z}_2)$$ and $$P$$ is a phase polynomial:

$$P(x) = \sum_{y \in \mathbb{Z}_2^n} a_y(x_1y_1 \oplus x_1y_2 \oplus \cdots \oplus x_ny_n)$$
Phase polynomial synthesis problems

Given a phase polynomial

\[ P(x) = \sum_{y \in \mathbb{Z}_2^n} a_y (x_1 y_1 \oplus x_1 y_2 \oplus \cdots \oplus x_n y_n) \]

can we synthesize with the minimal...

(Amy, Maslov, & Mosca 2014) \( T \)-depth
Poly-time via reduction to Matroid partitioning.

(Amy, Azimzadeh, & Mosca 2018) CNOT gates
Unique combinatorial problem, \( NP \)-hard in some cases.

(Amy & Mosca 2019) \( T \) gates
Poly-time equivalent to decoding \( RM(n - 4, n) \).
Just the tip of the iceberg...

- Near term/non-fault-tolerant computers
- Compilation & error correction co-design
- **Symbolic synthesis**
- ZX-calculus compilation
- **Cost lower bounds**
- Optimal reversible circuit synthesis
- **Relative-phase implementations**
- Measurement-assisted circuits
- Pebbling strategies
- **Applications of number-theoretic embeddings**
- Algorithm-specific compilation problems
- Pauli-based computing
- ???
Thank you!

I’m looking for students!