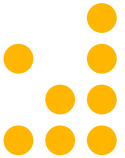
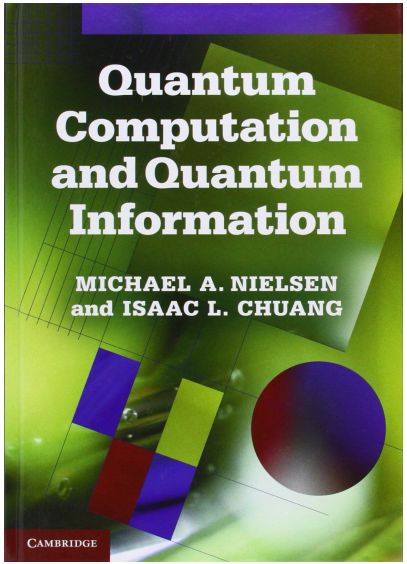


CMPT 478/981 Spring 2025 Quantum Circuits & Compilation

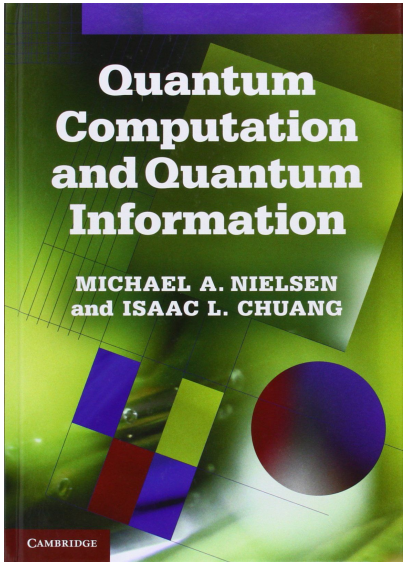
Matt Amy

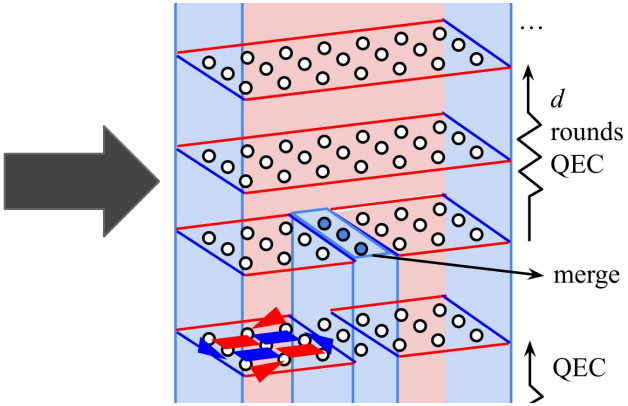
What is this course about?

CMPT 476:

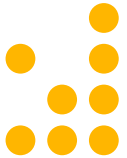


CMPT 478:





What is this course about?



"Quantum software"

1. Software that runs on a quantum computer (**circuits**)
2. Software that helps produce circuits (**compilers**)
    a. Circuit synthesis
    b. Circuit optimization
    c. Fault tolerance
    d. Hardware routing
    e. Programming languages

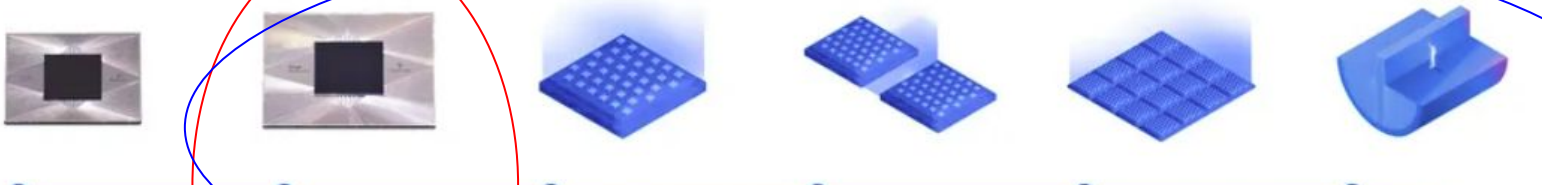
See also the Oxford course https://www.cs.ox.ac.uk/teaching/courses/2023-2024/qsoft/

Why quantum software?

Our quantum computing roadmap

Our focus is to unlock the full potential of quantum computing by developing a large-scale computer capable of complex, error-corrected computations. We're guided by a roadmap featuring six milestones that will lead us toward top-quality quantum computing hardware and software for meaningful applications.

Need software for these



Milestone 1
Beyond classical

Physical Qubits: 54
Logical Qubit Error Rate: –

Milestone 2
Quantum error correction

Physical Qubits: 10^1
Logical Qubit Error Rate: 10^{-6}

Milestone 3
Building a long-lived logical qubit

Physical Qubits: 10^1
Logical Qubit Error Rate: 10^{-6}

Milestone 4
Creating a logical gate

Physical Qubits: 10^2
Logical Qubit Error Rate: 10^{-6}

Milestone 5
Engineering scale up

Physical Qubits: 10^4
Logical Qubit Error Rate: 10^{-6}

Milestone 6
Large error-corrected quantum computer

Physical Qubits: 10^6
Logical Qubit Error Rate: 10^{-9}

We are here

Why quantum software?

NISQ (**near-term noisy intermediate scale QC**)

- Hybrid quantum-classical optimization algs
- Software necessary for
  - **Routing**
  - Error mitigation (NOT error correction)
  - Compiling ansatz (variational circuit template)

But why quantum software?

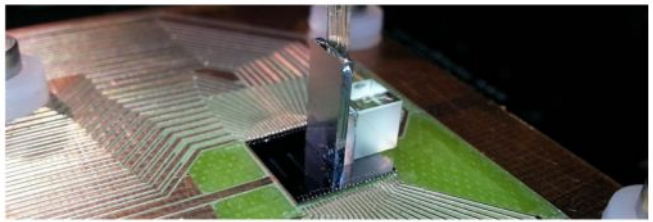
Expectation:

QUANTUM COMPUTING KILLS ENCRYPTION

by: Elliot Williams

78 Comments

September 29, 2015



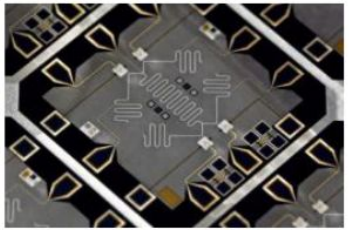
Imagine a world where the most widely-used cryptographic methods turn out to be broken: quantum computers allow encrypted Internet data transactions to become readable by anyone who happened to be listening. No more HTTPS, no more PGP. It sounds a little bit sci-fi, but that's exactly the scenario that cryptographers interested in post-quantum crypto are working to save us from. And although the (potential) threat of quantum computing to cryptography is already well-known, this summer has seen a flurry of activity in the field, so we felt it was time for a recap.

HOW BAD IS IT?

Reality:

Physicists demonstrate that 15=3x5 about half of the time

19 August 2012



The device in the photomicrograph was used to run the first solid-state demonstration of Shor's algorithm. It is made up of four phase qubits and five superconducting resonators, for a total of nine engineered quantum elements. The quantum processor measures one-quarter inch square. Credit: UCSB

never been done before," said Erik Lucero, the paper's lead author. Now a postdoctoral researcher in experimental quantum computing at IBM, Lucero was a doctoral student in physics at UCSB when the research was conducted and the paper was written.

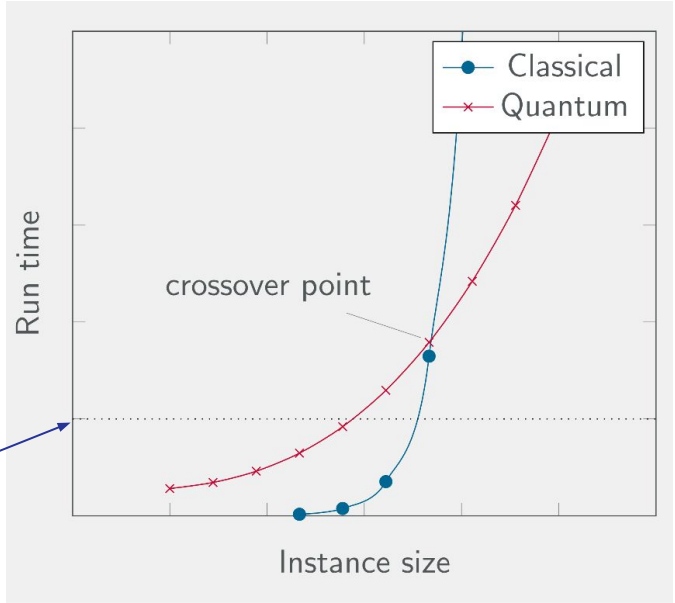
"What is important is that the concepts used in factoring this small number remain the same when factoring much larger numbers," said Andrew Cleland, a professor of physics at UCSB and a collaborator on the experiment. "We just need to scale up the size of this processor to something much larger. This won't be easy, but the path forward is clear."

Practical applications motivated the research, according to Lucero, who explained that factoring very large numbers is at the heart of cybersecurity protocols, such as the most common form of

Resource estimation

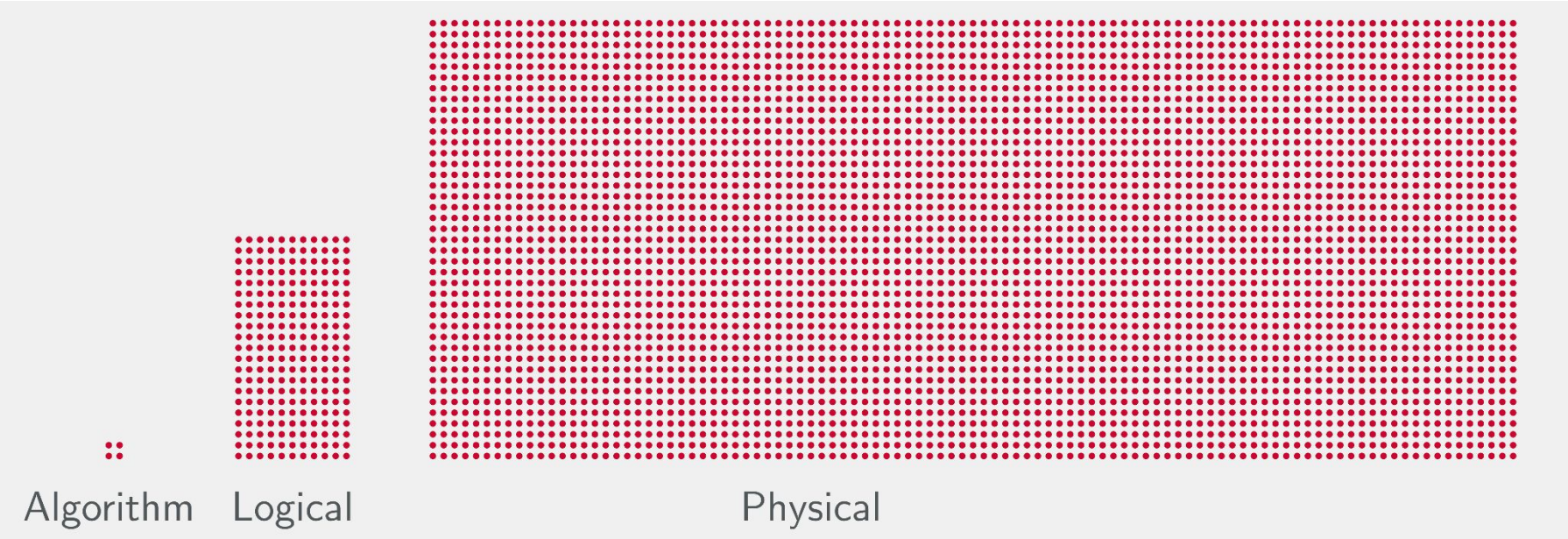
- 2010's: research shifted from showing theoretical to practical advantage

- Goal is to find crossover points where a quantum algorithm will outperform classical

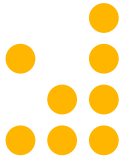


Found it was actually hard to compile algorithms in practice, and the results were extremely pessimistic

Quantum overheads



Goals for quantum computer scientists



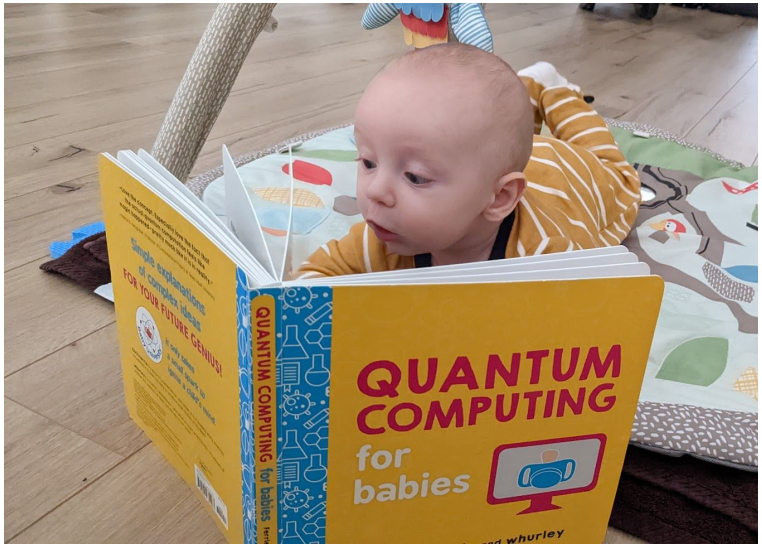
- Reduce the resources necessary to run algorithms (compilation/optimization)
- Reduce difficult of estimation (programming languages/software tooling)
- Reduce uncertainty in results (verification)
- **Figure out how to run the damn things on hardware**

Last ~15 years have seen significant progress in all of the above!

(and that's where this course begins…)

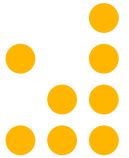
Course history

2022:



2025:

Format of the course

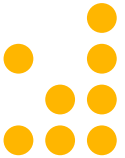


Website:
https://www.cs.sfu.ca/~meamy/Teaching/cmpt981/index.html

- Seminar style
  - Each week read 2-3 papers & discuss
  - I'll ask one of you to summarize & kick-off discussion for each paper
  - Lectures as needed to introduce topics & background
- Topics (tentative)
  - Circuit theory (background, 1-2 weeks)
  - Circuit compilation (2-3 weeks)
  - Circuit routing (1 week)
  - Circuit optimization (2-3 weeks + background on circuit representations)
  - Verification & simulation (1-2 weeks)
  - Programming languages

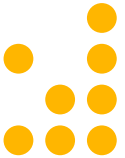
I'm flexible: let me know what you're most interested in!!!

Grading

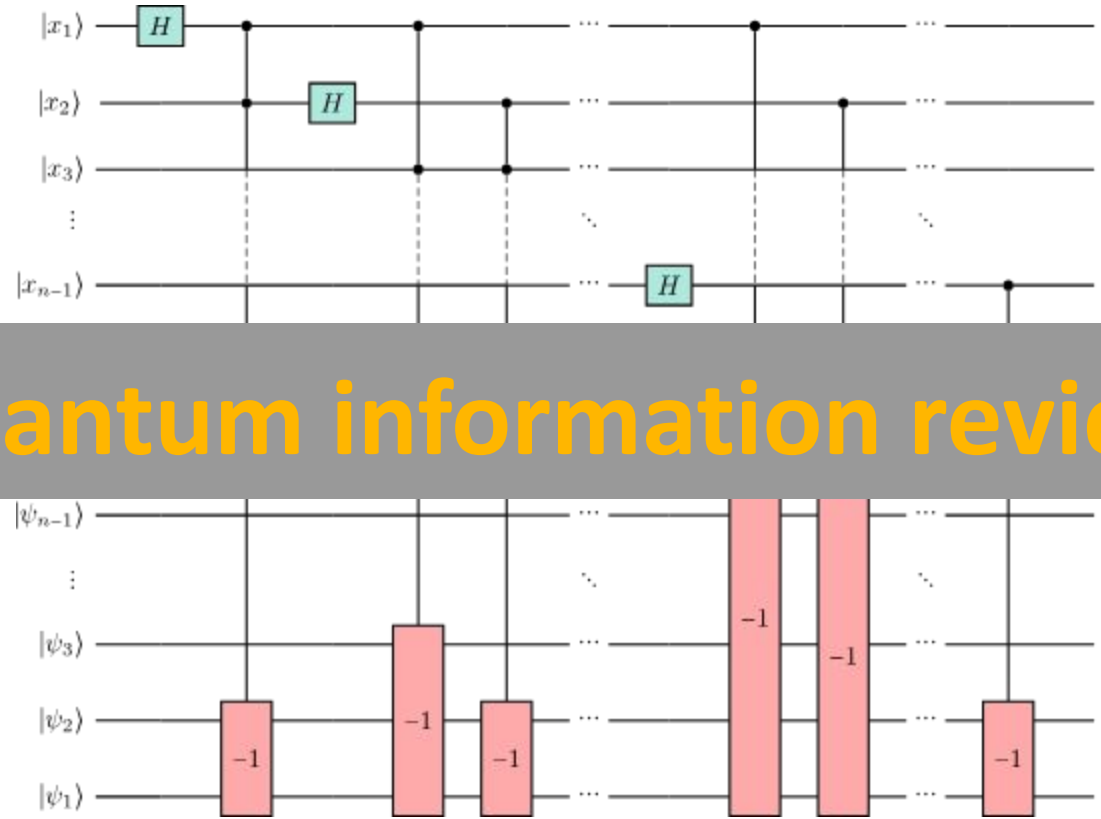


- 40% project
  - Research project related to the course (can be in small groups)
  - Tentative milestones:
    - Feb. 27th: Propose your project
    - Apr. 11th: Project is due
- 10% class participation
  - Discussion & summaries
- 25% homework
  - 1-2 assignments which will involve some coding
  - Short (1-2 paragraph) paper reviews
- 25% presentation
  - Presenting a research paper of your choice towards the end of the term

Today’s agenda

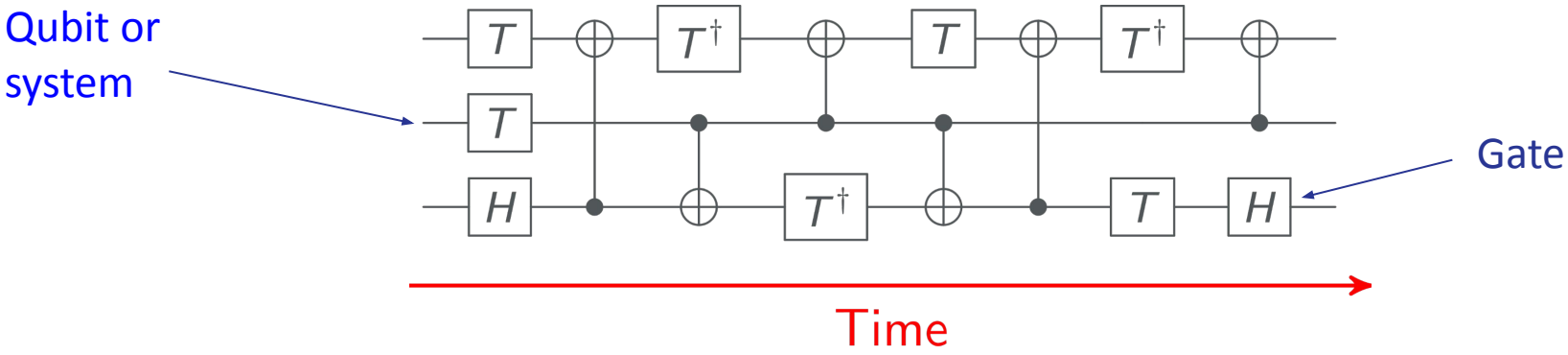


- Review of quantum information
- Review of the circuit model & circuit diagrams
- Overview of quantum compilation



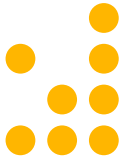
Quantum information review

What are quantum circuits?



1. “Assembly language” for quantum processors?
2. Theoretical model of (quantum) computation?
3. Diagrammatic language for quantum theory?
4. Particular factorizations of a matrix?
5. All of the above? ✅

Quantum theory = S.C.U.M.



S - States are unit vectors in a Hilbert space

C - Compound systems are described by the tensor product

U - Time evolution is unitary

M - Measurement probabilities follow the Born rule

Quantum circuits = diagrammatic language of quantum theory

States

A **Hilbert space** is a (finite dimensional for our purposes) Complex vector space

$$\mathcal{H} = \mathbb{C}^d$$

Write a state as a **ket**

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_d \end{bmatrix} = |\psi\rangle \in \mathbb{C}^d$$

Inner products

Standard inner product on a Hilbert space is

Complex conjugate

$$\langle \psi, \phi \rangle = \sum_i a_i b_i^*$$

A **bra** $\langle \psi |$ denotes the linear functional

$$\langle \psi | : |\phi\rangle \mapsto \langle \psi | \phi \rangle = \langle \psi, \phi \rangle$$

Concretely, it is represented by the **conjugate transpose (dagger)**

$$\begin{bmatrix} a_1^* & a_2^* & \cdots & a_d^* \end{bmatrix} = \langle \psi | = (|\psi\rangle)^\dagger$$

Bases

- For C^d, denote the elementary basis vectors by

$$|0\rangle = e_0 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad |1\rangle = e_1 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \qquad \cdots \qquad |d-1\rangle = e_{d-1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

- Vectors in a **particular** orthonormal basis are **classical states**
- A quantum state is viewed as a **superposition** of classical states

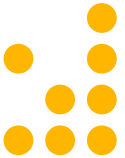
$$|\psi\rangle = \sum_i a_i |i\rangle$$

Amplitude of state |i>

- Useful to remember:

$$\langle i|j\rangle = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{otherwise} \end{cases}$$

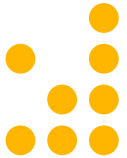
State postulate



QC Postulate #1

The state of an isolated system is described by a unit vector in a Hilbert space

Tensor product



Given vector spaces V & W, the **tensor product** V⊗W is a vector space with a **bi-linear map**

$$\otimes : V \times W \rightarrow V \otimes W$$

$$v \in V, w \in W \implies v \otimes w \in V \otimes W$$

If V and W have orthonormal bases $\{|e_i\rangle\}$, $\{|f_j\rangle\}$, then V⊗W has an orthonormal basis $\{|e_i\rangle \otimes |f_j\rangle\}$

Properties of tensor products

$$(s \cdot a) \otimes b = s(a \otimes b) \quad \text{(scalar mult)}$$

$$(a \otimes b) \otimes c = a \otimes (b \otimes c) \quad \text{(associative)}$$

$$(a + b) \otimes c = a \otimes c + b \otimes c \quad \text{(left linear)}$$

$$a \otimes (b + c) = a \otimes b + a \otimes c \quad \text{(right linear)}$$

Recall: inner products are similarly bi-linear

Given two states, their tensor product is

$$(\sum_i a_i |e_i\rangle) \otimes (\sum_j b_j |f_j\rangle) = \sum_{ij} a_i b_j |e_i\rangle \otimes |f_j\rangle$$

Note: The below are equivalent

$$|i\rangle \otimes |j\rangle = |i, j\rangle = |ij\rangle$$

Compound system postulate

QC Postulate #2

The combined state of systems with state spaces V & W is a unit vector in V⊗W

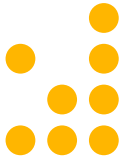
Qubits: systems with 2-dimensional states

- n qubits has state space $\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 \simeq \mathbb{C}^{2^n}$
- "Everything works out":

$$|1\rangle \otimes |0\rangle = |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = |2\rangle$$

= 10 in binary

Linear operators



A linear operator from V to W is an operator which is linear…

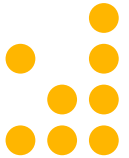
$$T(\alpha|\psi\rangle + \beta|\phi\rangle) = \alpha T|\psi\rangle + \beta T|\phi\rangle$$

Since we work in finite dimensions, typically define a linear operator uniquely by its **action** on a basis of V:

$$X : \mathbb{C}^2 \to \mathbb{C}^2$$
$$|0\rangle \mapsto |1\rangle$$
$$|1\rangle \mapsto |0\rangle$$

$$Z : \mathbb{C}^2 \to \mathbb{C}^2$$
$$|0\rangle \mapsto |0\rangle$$
$$|1\rangle \mapsto -|1\rangle$$

Outer products



The **outer product** $|\psi\rangle\langle\phi| : |\chi\rangle \mapsto (\langle\phi|\chi\rangle)|\psi\rangle$ is convenient for building and decomposing linear operators

Fact:

given two VS V, W with bases $\{|e_i\rangle\}$, $\{|f_j\rangle\}$, a linear operator T:V -> W can be represented by its **matrix** over those bases,

$$T = \sum_{ij} T_{ij} |f_j\rangle\langle e_i|$$

(Some) Classes of operators

- Normal: $AA^\dagger = A^\dagger A$
- Unitary: $UU^\dagger = I = U^\dagger U$
- Hermitian: $H^\dagger = H$
- Projector: $P^2 = P$

Observe that a unitary operator is invertible…

Recall: $$A^\dagger = (A^*)^t$$

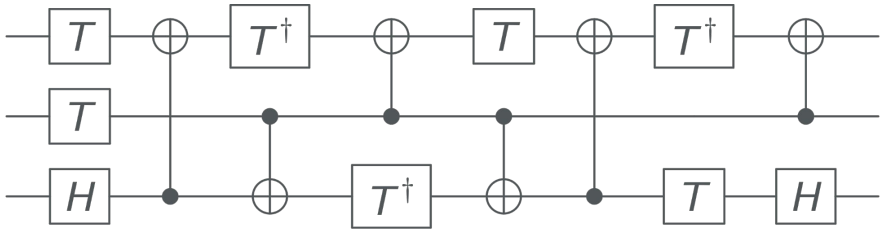
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^\dagger = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}$$

Time evolution postulate

QC Postulate #3

The physical evolution of a closed system is described by a unitary operator acting on its Hilbert space

Write a unitary operator as a circuit built from smaller operators



How do we build circuits?

Circuits (operators) can be composed **sequentially** or **in parallel**

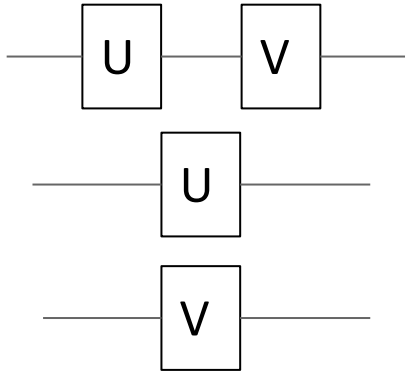
For $U : X \to Y$, $V : Y \to Z$

(Sequential) $VU : X \to Z$

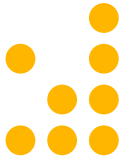
(Parallel) $U \otimes V : X \otimes Y \to Y \otimes Z$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \otimes \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} A \otimes \begin{bmatrix} E & F \\ G & H \end{bmatrix} & B \otimes \begin{bmatrix} E & F \\ G & H \end{bmatrix} \\ C \otimes \begin{bmatrix} E & F \\ G & H \end{bmatrix} & D \otimes \begin{bmatrix} E & F \\ G & H \end{bmatrix} \end{bmatrix}$$

Diagrammatically,



(Unitary) circuit over a gate set



Algebraic view:

given an inverse-closed set G of **unitary gates** $g_i : V_i \to V_i$, a **circuit over G** is a well-formed term over the signature $(G, I, \cdot, \otimes, \dagger)$

Notes:

$(G, I, \cdot, \dagger)$ — is a group(**-oid**, since $\cdot$ is partial)

$(G, I, \otimes)$ — is a monoid ($\otimes$ is total here)

Often fix V_i = C^{2^n} for all i and view $(G, I, \cdot, \dagger)$ as a group

Examples

Pauli

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

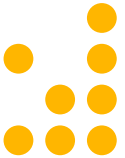
Clifford

$$S = \sqrt{Z} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

non-Clifford

$$T = \sqrt{S} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

Gate sets as groups

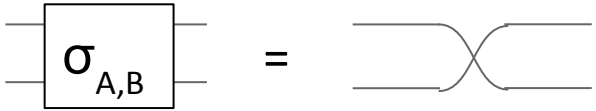


Aside: symmetries

The difference between

is usually not important, so we often assume a **symmetric** monoidal structure, that is the order of Hilbert spaces can be rearranged freely via SWAPs

$$\sigma_{A,B} : A \otimes B \to B \otimes A$$



Aside: Dagger symmetric monoidal categories

= well formed terms over the signature $(G, I, \sigma, \cdot, \otimes, \dagger)$

- Prototypical **model** is finite dimensional Hilbert spaces (**FdHilb**)
  - But we'll see other models which serve to interpret circuits!
- Graphical language is the language of quantum circuits
- Axiomatizes properties of the kronecker product & hermitian adjoint, e.g.

$$(A^\dagger)^\dagger = A$$
$$(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$$
$$(AB)^\dagger = B^\dagger A^\dagger$$

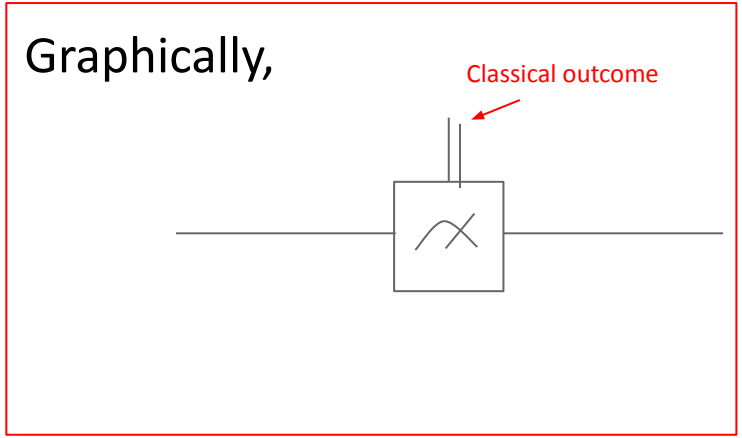
Measurement postulate pt 1

Given a system in the state $|\psi\rangle = \sum_i a_i |e_i\rangle$, a **complete measurement** in basis $\{|e_i\rangle\}$ produces two things:

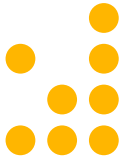
- Result "i"
- The state $|e_i\rangle$

with probability $|\langle e_i|\psi\rangle|^2 = |a_i|^2$

Graphically,



Measurement postulate pt 2



Given a set of **measurement operators** $\{M_i\}$ such that

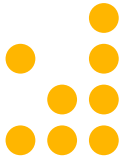
$$\sum_i M_i^\dagger M_i = I$$

the measurement (incomplete if not every $M_i^\dagger M_i$ is rank 1) of $|\psi\rangle$ produces

- Result i
- State $\frac{M_i|\psi\rangle}{\sqrt{p(i)}}$

with probability $p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle$

Projective measurement



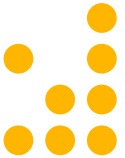
- Given a basis $\{|e_i\rangle\}$, set $M_i = |e_i\rangle\langle e_i|$
- Then each M_i is Hermitian & a projector with $\sum_i M_i^\dagger M_i = I$
- Resulting measurement projects & normalizes onto the state $|e_i\rangle$ w/ prob

$$\langle\psi|M_i^\dagger M_i|\psi\rangle = \langle\psi|e_i\rangle\langle e_i|\psi\rangle = |\langle e_i|\psi\rangle|^2$$

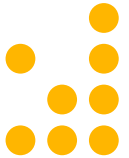
Useful fact:

Computational (i.e. $\{|i\rangle\}$) basis measurements + unitary transformations suffice to implement any projective measurement

Example: partial measurement



Aside: global phase invariance

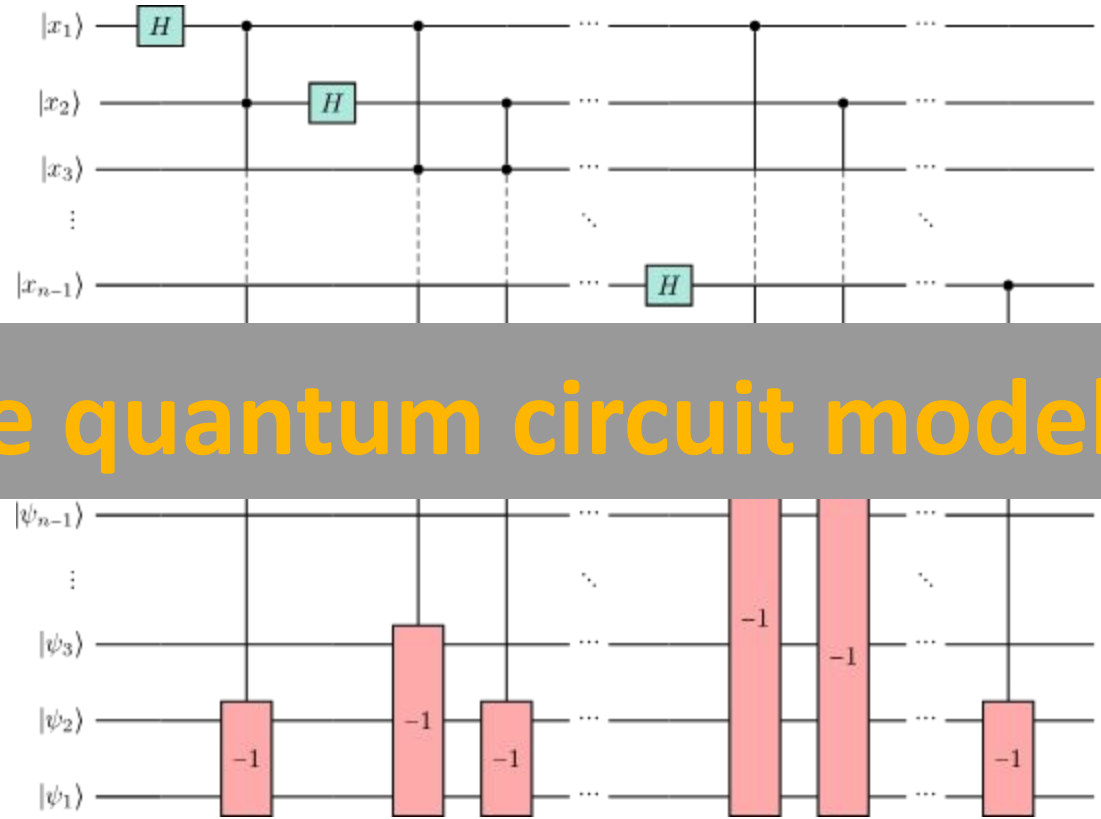


- Two states which differ by a **global phase** are indistinguishable, i.e.

$$|\psi\rangle = e^{i\theta}|\phi\rangle$$

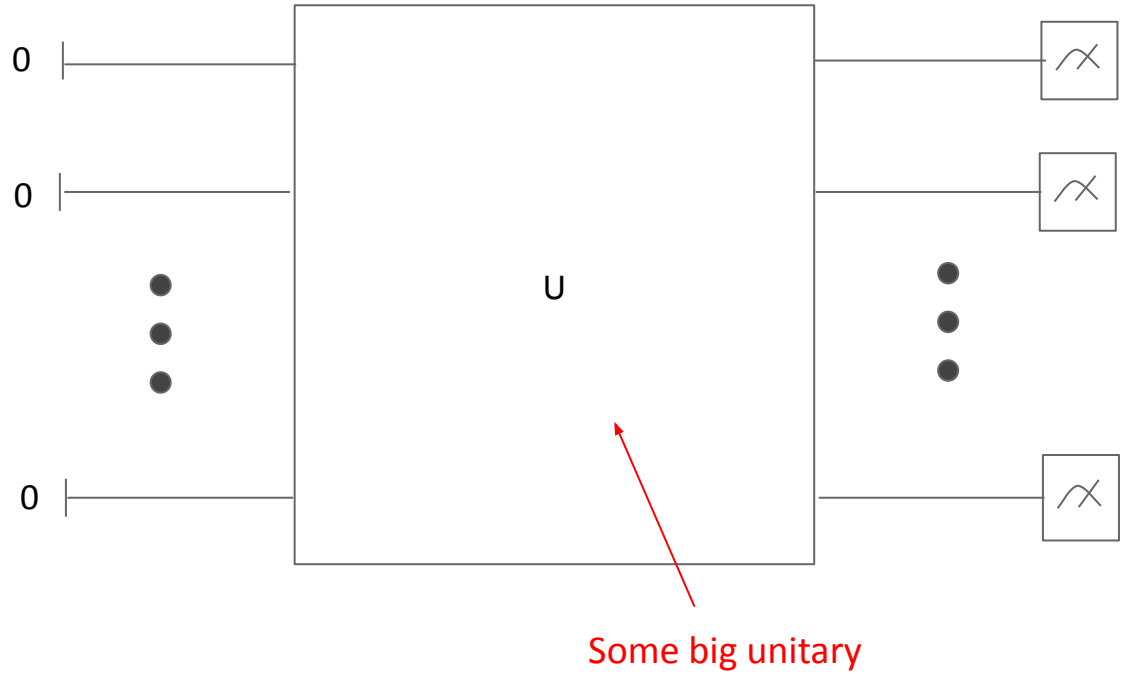
- Two states which differ by a **relative phase** are distinguishable, i.e.

$$\langle 0|\psi\rangle = \langle 0|\phi\rangle, \qquad \langle 1|\psi\rangle = e^{i\theta}\langle 1|\phi\rangle$$



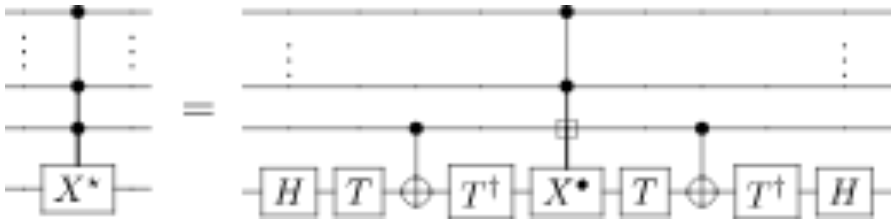
The quantum circuit model

The quantum circuit model



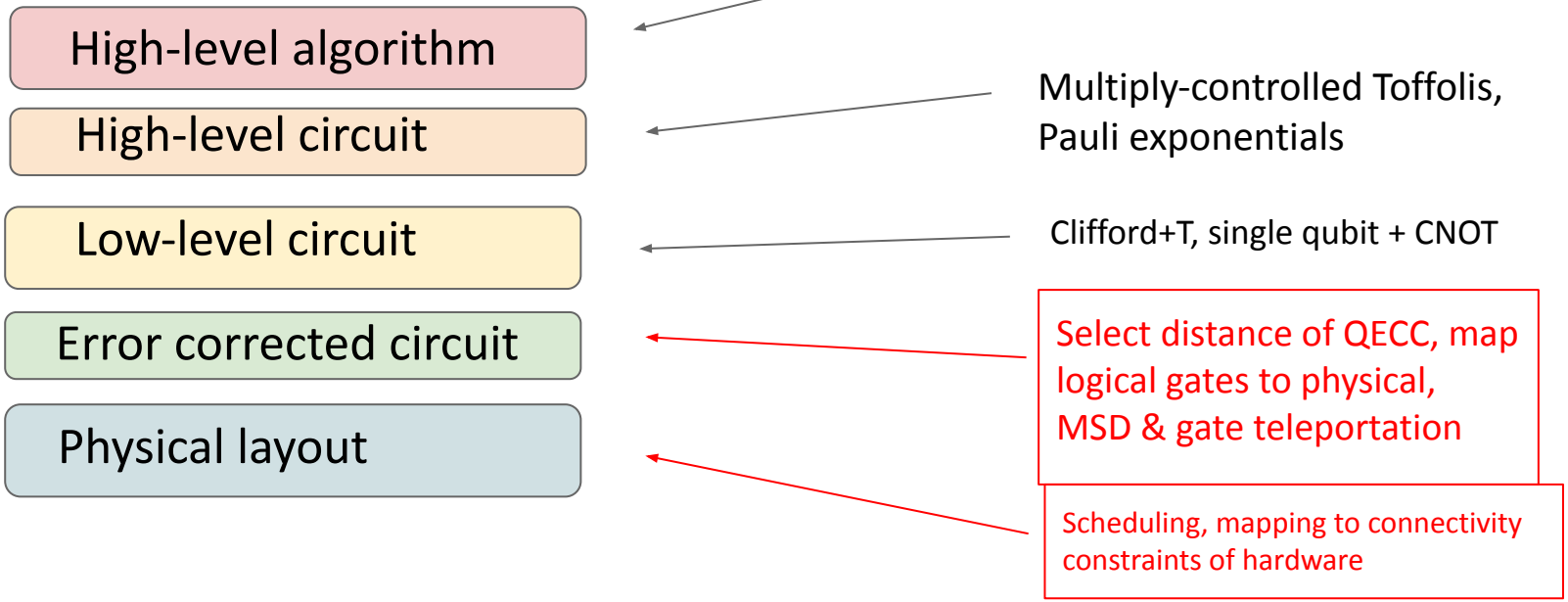
Compilation

- Simplest form: take the big unitary U (itself possibly a circuit) and write it over an intended gate set

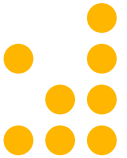


A bit more accurate

The layered reality:



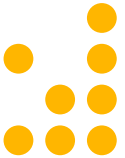
What makes a gate set?



1. Should be **universal**
2. Should be **efficient**
3. Should be **implementable**
   a. Hardware layer ⇒ implementable directly on the physical media, usually by some control pulses
   b. Logical layer ⇒ implementable directly on encoded data (more on this later)

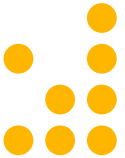
Hardware	Logical
$\exp(-i\frac{\theta}{2}Z\otimes Z) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 & 0 & 0 \\ 0 & e^{i\frac{\theta}{2}} & 0 & 0 \\ 0 & 0 & e^{i\frac{\theta}{2}} & 0 \\ 0 & 0 & 0 & e^{-i\frac{\theta}{2}} \end{bmatrix}$	$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$

The classical case



- State of a bit is 0 or 1
- State of n bits is an n-bit string x1x2…xn
- Function is a map from n-bit strings to m-bit strings
- Classical gates:

Classical universality

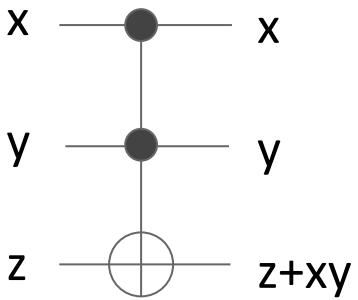


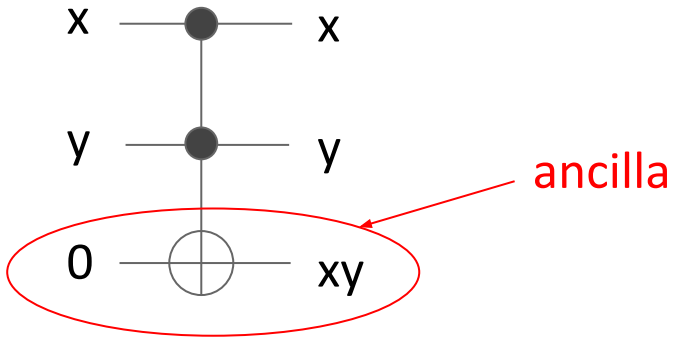
A gate set G is **universal** for classical computing if any function
$f:\{0,1\}^n \rightarrow \{0,1\}^m$ from n bits to m bits can be implemented by a circuit over G

Theorem: {AND, XOR, NOT, FANOUT} is universal

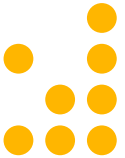
The reversible case

- Quantum computing includes (**and relies on**) classical computation
- But unitary operators are invertible
  - Problem: AND gate is not invertible!
  - Dissipating 1 bit of information dissipates $K_B T \ln 2$ J of energy (Landauer's principle)
- Instead use a reversible embedding of the AND (Toffoli) + **ancillas**





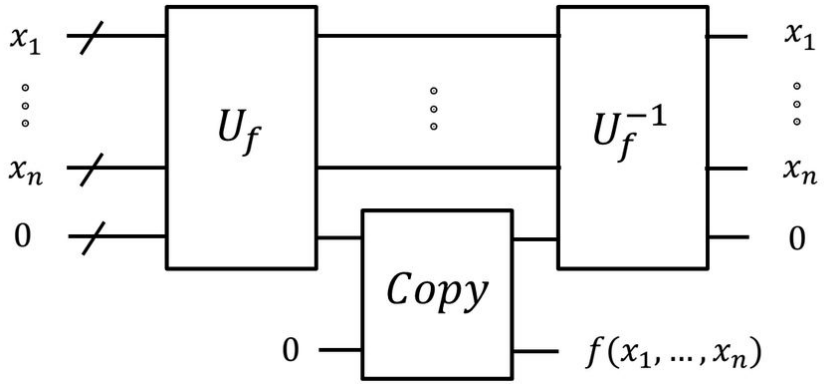
Classical universality for QC



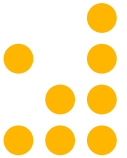
Theorem: {TOFFOLI + ancillas} is universal for reversible computation

Resource usage

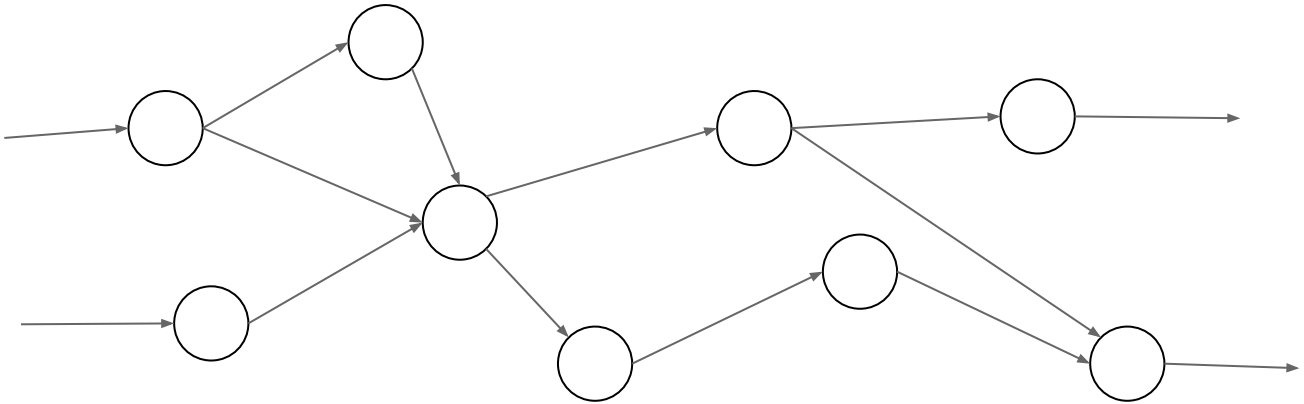
- For a classical circuit using space S & time T, reversible version uses space O(S + T) and time O(T)
- How to reclaim space at the end? Bennett trick



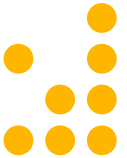
Pebble games (1st real compilation “technique”)



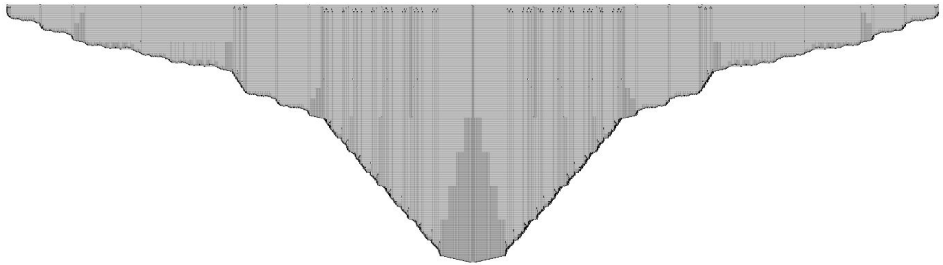
- Classic problem in computer science
- Given a DAG, try to place pebbles on out-going nodes **using the minimum number of intermediate pebbles**
- A node can only be pebbled if its predecessors are pebbled



Time vs. space trade-off in RC

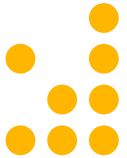


- No consideration for pebbling results in this:



- Other extreme uses exponential time

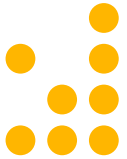
Bennett's pebble games



Theorem (Bennett 1989): a classical computation with space S and time T can be implemented reversibly with

- Space $O(S \log T)$ & time $O(T^{1+e})$, or
- Space $O(ST^{e})$ & time $O(T)$

A note on reversible vs quantum compilation



- Reversible computation forms bulk of most algorithm implementations
  - Notable exceptions: NISQ algorithms & Trotterization-based Hamiltonian simulation
- Most algorithms rely on a classical sub-routine performed in superposition
  - Shor: Modular exponentiation
  - Search/Grover: Evaluation of the classical search function
  - QSP/LCU-based Hamiltonian simulation: Giant multiplexor
  - Quantum walks: Adjacency computation on a graph

Optimizing & compiling classical computation
is the main job for most high-level compilers