CSCI 435: Algorithms and Complexity
10. Quantum Computing

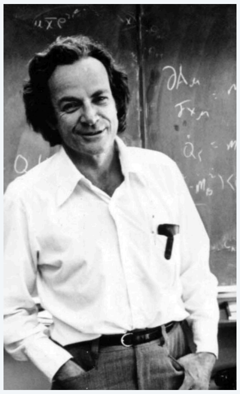
- qubits
- quantum circuits
- Deutsch's problem
- more quantum algorithms
- quantum complexity theory

Quantum computing

Richard Feynman, 1981.

"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem because it doesn't look so easy."

"Can you do it with a new kind of computer — a quantum computer? Now it turns out, as far as I can tell, that you can simulate this with a quantum system, with quantum computer elements. It's not a Turing machine, but a machine of a different kind."



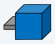
CSCI 435: Algorithms and Complexity
10. Quantum Computing

- qubits
- quantum circuits
- Deutsch's problem
- more quantum algorithms
- quantum complexity theory

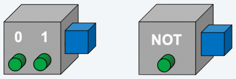
Image references: Richard Cleve

Classical information

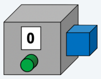
Bits. A bit is like a device that stores a single binary piece of information.



We can set the value of this bit to either 0 or 1, and we can flip the value.

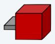


We can also read the value of a bit.

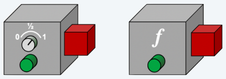


Classical information

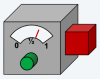
“Analog” bits. Consider an “analog” bit that can store any value [0,1].



We can set the value of this “analog” bit, and we can apply analog transformations $f\colon [0,1] \to [0,1]$.

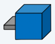


We can also read the value of an “analog” bit.

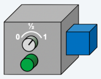


Probabilistic information

Probabilistic bits. Let’s see what happens when we plug our classical bit into our “analog” bit “setting machine”.



When we set the value of a classical bit, it gets set to 1 with some probability between 0 and 1, and it gets set to 0 otherwise.



If we know how the dial was set, we can describe the state of this system by a probability vector $\begin{bmatrix}p_0\\p_1\end{bmatrix}$.

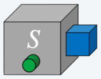
The actual state is either 0 or 1, but we don’t know which until we read it. We can’t retrieve p_0 or p_1 if we didn’t know how the dial was set.

Probabilistic information

Transformations. We can also apply transformations to a probabilistic bit. A transformation is a 2×2 stochastic matrix

$$S = \begin{bmatrix} s_{00} & s_{01} \\ s_{10} & s_{11} \end{bmatrix}$$

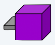
where $s_{00}, s_{01}, s_{10}, s_{11} \geq 0$, $s_{00} + s_{10} = 1$, and $s_{10} + s_{11} = 1$.



Applying a transformation to a probabilistic bit changes the probability vector to $S\begin{bmatrix} p_0 \\ p_1 \end{bmatrix}$.

Quantum information

Qubits. A quantum bit is neither classical nor probabilistic, but it is most similar to a probabilistic bit.



Qubits have probability amplitudes associated with the values 0 and 1.

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \in \mathbb{C}^2$$

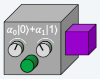
Euclidean distance

Every probability amplitude has the property that $\sqrt{|\alpha_0|^2 + |\alpha_1|^2} = 1$.

Key difference. The explicit state of a qubit is not 0 or 1. The explicit state is the amplitude vector $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$.

Quantum information

Setting state. We can set the state of a qubit by setting its amplitude vector.

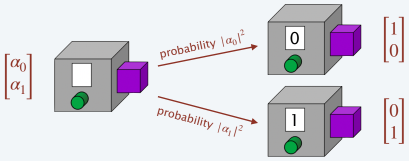


There are two degrees of freedom to set: the amplitudes α_0 and α_1. These can be expressed in polar form:

$$\alpha_0 = \sin(\theta)$$
$$\alpha_1 = e^{i\phi}\cos(\theta)$$

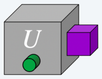
Quantum information

Measuring state. We can measure a qubit, but this collapses its state.


$\begin{bmatrix}\alpha_0\\ \alpha_1\end{bmatrix}$
probability $|\alpha_0|^2$
0
$\begin{bmatrix}1\\ 0\end{bmatrix}$
probability $|\alpha_1|^2$
1
$\begin{bmatrix}0\\ 1\end{bmatrix}$

Quantum information

Transformations. We can apply transformations to amplitude vectors. A transformation is a 2×2 unitary matrix.



Unitary transformations preserve probability amplitudes: we can transform a vector of length 1 and it remains a vector of length 1.

Foundations of quantum information. Setting state, measuring state, and applying unitary transformations.

Examples of unitary transformations

Rotation by θ.

$$R_\theta = \begin{bmatrix}\cos(\theta) & -\sin(\theta)\\ \sin(\theta) & \cos(\theta)\end{bmatrix}$$

Hadamard transform.

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

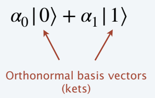
Pauli matrices.

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

"bit flip" "phase flip"

Qubits

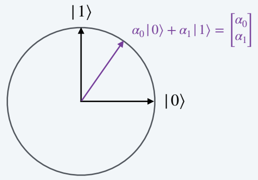
Notation. We can represent an amplitude vector $\begin{bmatrix}\alpha_0\\ \alpha_1\end{bmatrix} \in \mathbb{C}^2$ using the bra-ket notation (or Dirac notation).



$$\alpha_0 |0\rangle + \alpha_1 |1\rangle$$

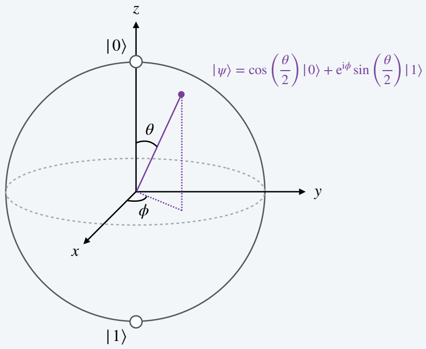
Orthonormal basis vectors (kets)

Geometric perspective.


$|1\rangle$
$\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{bmatrix}\alpha_0\\ \alpha_1\end{bmatrix}$
$|0\rangle$

Bloch sphere

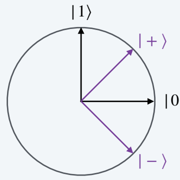
From $\mathbb{R}^2$ to $\mathbb{C}^2$. The "true" geometric perspective of a qubit looks like a sphere instead of a circle.


z
$|0\rangle$
$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$
θ
y
ϕ
x
$|1\rangle$

Superposition

Plus and minus qubits. In addition to $|0\rangle$ and $|1\rangle$, we also have

$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \quad \text{and} \quad |-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$



Distinguishing between qubits. We can distinguish between $|+\rangle$ and $|-\rangle$ by applying a 45° rotation and measuring the qubit.
But we can't perfectly distinguish between $|+\rangle$ and $|0\rangle$!

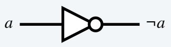
CSCI 435: Algorithms and Complexity
10. Quantum Computing

- qubits
- *quantum circuits*
- Deutsch's problem
- more quantum algorithms
- quantum complexity theory

Classical logic gates

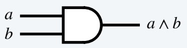
NOT.

a	$\neg a$
0	**1**
1	**0**



AND.

a	b	$a \wedge b$
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**



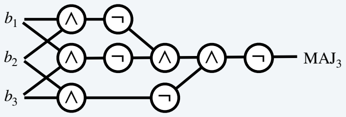
OR and XOR. Not necessary; we can simulate these using NOT and AND.

Computation using classical circuits

Models of computation. Turing machines, word RAM, Boolean circuits.

Majority. Suppose we have a 3-bit string and we want to determine whether the string contains more 0s than 1s (or vice versa).

b_1	b_2	b_3	MAJ_3
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

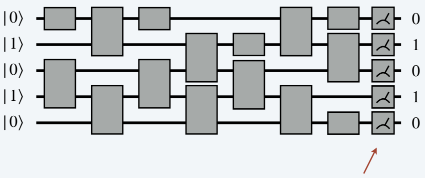


$$\mathrm{MAJ}_3(b_1, b_2, b_3) = (b_1 \wedge b_2) \vee (b_1 \wedge b_3) \vee (b_2 \wedge b_3)$$

Computation using quantum circuits

Quantum circuits.

- Input data enters as a computational basis state.
- Data flows from left to right through unitary gates.
- Data flows into measurement gates and exits as output.

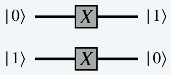


measurement gates

Quantum logic gates

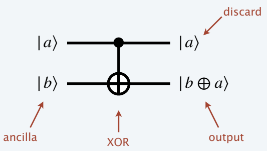
Pauli X gate (NOT). Flip a qubit from $|0\rangle$ to $|1\rangle$ or vice versa.

In	Out
$\lvert a\rangle$	$\lvert\neg a\rangle$
0	**1**
1	**0**



CNOT (controlled NOT). Flip the second qubit iff the first qubit is $|1\rangle$.

Input		Output	
$\lvert a\rangle$	$\lvert b\rangle$	$\lvert a\rangle$	$\lvert b\oplus a\rangle$
0	0	0	**0**
0	1	0	**1**
1	0	1	**1**
1	1	1	**0**



discard

ancilla

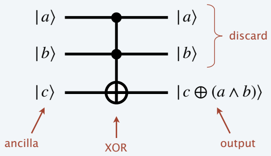
XOR

output

Quantum logic gates

Toffoli gate (quantum AND). Flip the third qubit iff both the first and second qubits are $|1\rangle$.

Input			Output		
$\lvert a\rangle$	$\lvert b\rangle$	$\lvert c\rangle$	$\lvert a\rangle$	$\lvert b\rangle$	$\lvert c\oplus(a\wedge b)\rangle$
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**



discard

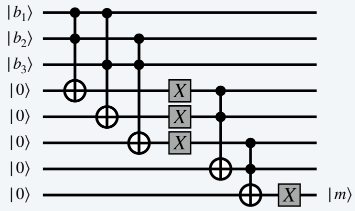
ancilla

XOR

output

Computation using quantum circuits

Majority. Suppose we have 3 qubits and we want to determine whether there are more $|0\rangle$s than $|1\rangle$s (or vice versa).



Theorem. Any classical circuit of size s can be simulated by a quantum circuit of size $O(s)$ using only Pauli X, CNOT, and Toffoli gates.

Theorem. Any quantum circuit of size s with n qubits can be simulated by a classical circuit of size $O(s \cdot n^2 \cdot 2^n)$.

CSCI 435: Algorithms and Complexity
10. Quantum Computing

- *qubits*
- *quantum circuits*
- *Deutsch's problem*
- *more quantum algorithms*
- *quantum complexity theory*

Black-box model

Queries. Imagine we have a black-box function f we want to learn about. We make queries to f and analyze its output.

Assumption. f is a function over a finite domain such as $\{0,1\}^n$.

Limitation. The only way we can learn about f is by querying it.

Deutsch’s problem

Constant or balanced? Consider $f\colon \{0,1\} \to \{0,1\}$. There are four such functions:

x	$f(x)$
0	**0**
1	**0**

x	$f(x)$
0	**1**
1	**1**

$f(0) = f(1)$

x	$f(x)$
0	**0**
1	**1**

x	$f(x)$
0	**1**
1	**0**

$f(0) \neq f(1)$

How many queries do we need to determine whether $f(0) = f(1)$?

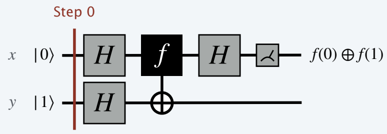
Classical computation. We require two queries.

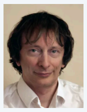
- Choose $a \in \{0,1\}$ to determine $f(a)$.
- We get no information about what $f(\neg a)$ may be.
- So we need another query.

Quantum computation. Just one query suffices!

Deutsch’s problem

Deutsch–Josza algorithm. Earliest example of quantum advantage: a quantum algorithm exponentially faster than any deterministic classical algorithm.



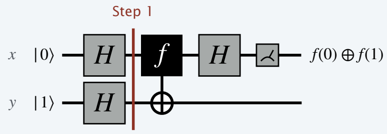

David Deutsch

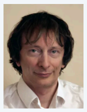
Richard Josza

Step 0. Initialize x and y to $|0\rangle$ and $|1\rangle$, respectively.

Deutsch’s problem

Deutsch–Josza algorithm. Earliest example of quantum advantage: a quantum algorithm exponentially faster than any deterministic classical algorithm.




David Deutsch

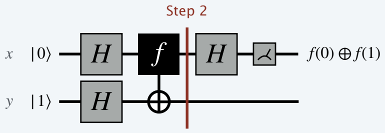
Richard Josza

Step 1. Apply H to each of x and y.

- This turns $|0\rangle$ into the $|+\rangle$ state and $|1\rangle$ into the $|-\rangle$ state.
- We can write these states together as $|+\rangle|-\rangle$.
- We can rewrite these states as a superposition $\frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$.

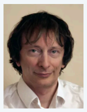
Deutsch's problem

Deutsch–Josza algorithm. Earliest example of quantum advantage:
a quantum algorithm exponentially faster than any deterministic classical algorithm.



Step 2. Apply f to the state.

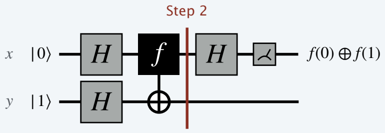
- On the second "wire", we now have $y \oplus f(x)$.
- There are four possibilities:
  - $f(x) = 0$: $\frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$
  - $f(x) = 1$: $\frac{1}{2}(-|00\rangle + |01\rangle - |10\rangle + |11\rangle)$
  - $f(x) = x$: $\frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)$
  - $f(x) = x \oplus 1$: $\frac{1}{2}(-|00\rangle + |01\rangle + |10\rangle - |11\rangle)$


David Deutsch

Richard Josza

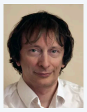
Deutsch's problem

Deutsch–Josza algorithm. Earliest example of quantum advantage:
a quantum algorithm exponentially faster than any deterministic classical algorithm.



Step 2. Apply f to the state.

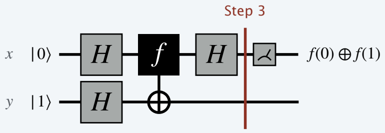
- The four possibilities factorize into two states:
  - $\pm|+\rangle|-\rangle$ if $f(0) = f(1)$;
  - $\pm|-\rangle|-\rangle$ if $f(0) \neq f(1)$.
- Note that the second qubit is now decoupled, so we can ignore it from this point on.


David Deutsch

Richard Josza

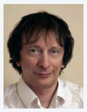
Deutsch's problem

Deutsch–Josza algorithm. Earliest example of quantum advantage:
a quantum algorithm exponentially faster than any deterministic classical algorithm.



Step 3. Apply H to x.

- This results in one of two states:
  - $\pm|0\rangle|-\rangle$ if $f(0) = f(1)$;
  - $\pm|1\rangle|-\rangle$ if $f(0) \neq f(1)$.
- All that remains is to measure the first qubit!
  - If it's 0, then f is constant.
  - If it's 1, then f is balanced.


David Deutsch

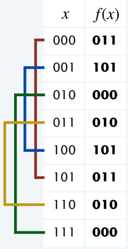
Richard Josza

CSCI 435: Algorithms and Complexity
10. Quantum Computing

- qubits
- quantum circuits
- Deutsch's problem
- more quantum algorithms
- quantum complexity theory

Simon's algorithm

Two-to-one functions. Consider a function $f\colon \{0,1\}^n \to \{0,1\}^n$ where every point in the range has exactly two preimages.



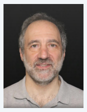
x	$f(x)$
000	**011**
001	**101**
010	**000**
011	**010**
100	**101**
101	**011**
110	**010**
111	**000**

Simon's property. Given such an f, there is an $r \in \{0,1\}^n$ such that for every colliding pair (a, b), $a \oplus b = r$.

Simon's problem. Given access to a black box for f, compute r.

Classical computation. Requires $\Omega(2^{n/2})$ queries.

Quantum computation. Just $O(n^2)$ queries!



Daniel Simon

Shor's algorithm

Factoring. Given an odd composite integer n, what are its integer factors?

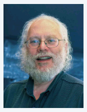
Shor's approach. Convert the factoring problem into a problem of finding the period of a function using quantum Fourier transforms.

Classical computation. No polynomial-time algorithm is known.

Quantum computation. Factoring an integer N takes $O(\log(N)^3)$ time!

In practice.

- In 2001, Shor's algorithm successfully factored $15 = 3 \times 5$.
- In 2012, it successfully factored $21 = 7 \times 3$.
- In 2019, it was used to try and factor 35, but the quantum computer accumulated too many errors.



Peter Shor

Grover's algorithm

Searching. Given a black-box function $f\colon \{0,1\}^n \rightarrow \{0,1\}$, find an input value $x \in \{0,1\}^n$ such that $f(x) = 1$.

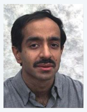
Grover's approach. Use amplitude amplification to "boost" the amplitude of desired states in a superposition, making them more likely to be measured.

Classical computation. In the worst case, requires 2^n queries.

Quantum computation. Can be done in $O(2^{n/2})$ queries.

still exponential?

Lower bound. Any quantum algorithm for searching requires $\Omega(2^{n/2})$ queries, so Grover's algorithm is optimal!



Lov Grover

CSCI 435: Algorithms and Complexity
10. Quantum Computing

- *qubits*
- *quantum circuits*
- *Deutsch's problem*
- *more quantum algorithms*
- *quantum complexity theory*

Classical circuit complexity

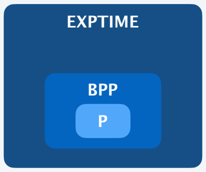
P. Class of decision problems solvable by a deterministic classical circuit of size $O(n^c)$ for some constant c.

BPP. Class of decision problems solvable by a probabilistic classical circuit of size $O(n^c)$ for some constant c with worst-case error probability $\leq \frac{1}{2} - \epsilon$.

EXPTIME. Class of decision problems solvable by a deterministic classical circuit of size $O(2^{n^c})$ for some constant c.

Theorem. **P** $\subseteq$ **BPP** $\subseteq$ **EXPTIME**.

Question. What is the quantum analogue of **P** and **BPP**?



Quantum circuit complexity

BQP. Class of decision problems solvable by a quantum circuit of size $O(n^c)$ for some constant c with worst-case error probability $\leq \frac{1}{2} - \epsilon$.

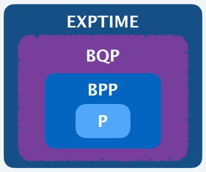
Theorem. **BPP** ⊆ **BQP**.

Pf sketch. Any circuit in **BPP** can be simulated by generating random qubits and simulating the resulting quantum circuit.

Theorem. **BQP** ⊆ **EXPTIME**.

Pf sketch. Any circuit in **BQP** can be simulated by an exponential-size classical circuit.

Question. Is **BPP** ≠ **BQP**?
This would imply quantum supremacy: quantum computers are stronger than classical computers.


EXPTIME
BQP
BPP
P