Deutsch-Josza and Simon's algorithms

February 10, 2026

Primitives for quantum computation: classical computations on quantum machines

- Let $f : \{0,1\}^n \to \{0,1\}$ and $C(f)$ be the smallest *classical circuit* that computes f.
- There exists a *quantum circuit* of size $O(C(f))$ which, for each input x to f, computes the following unitary transformation U_f (Bennet, 1973):

$$U_f : |x\rangle |y\rangle \to |x\rangle |y \oplus f(x)\rangle$$

- Only polynomial overheads. $\mathbf{P} \subseteq \mathbf{BQP}$ (will revisit later).
- If we feed U_f a superposition

$$\sum_{x \in \{0,1\}^n} \alpha_x |x\rangle |0\rangle$$

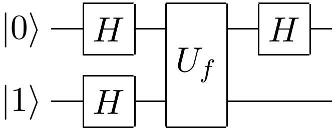
then, by linearity

$$U_f\Big(\sum_{x \in \{0,1\}^n} \alpha_x |x\rangle |0\rangle \Big) = \sum_{x \in \{0,1\}^n} \alpha_x U_f(|x\rangle |0\rangle) = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle |f(x)\rangle$$

Primitives for quantum computation: quantum parallelism

- $$\frac{|0\rangle+|1\rangle}{\sqrt{2}} = x \;\text{—}\; U_f \;\text{—}\; x = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$$
  $$|0\rangle = y \;\text{—}\; U_f \;\text{—}\; y \oplus f(x) = \frac{|0,f(0)\rangle+|1,f(1)\rangle}{\sqrt{2}}$$
- A single $f(x)$ circuit can evaluate the function at multiple values of x.
- Input and output *entangled*.
- What can we do with this?

Deutsch's algorithm



- $|\psi_0\rangle = |01\rangle$; $|\psi_1\rangle = \left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right]\left[\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right]$
- Applying U_f to the state $|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}$ we obtain

$$\begin{aligned} & |x\rangle(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle)/\sqrt{2} \\ = & |x\rangle(|f(x)\rangle - |1 \oplus f(x)\rangle)/\sqrt{2} \\ = & (-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle)/\sqrt{2} \end{aligned}$$

-

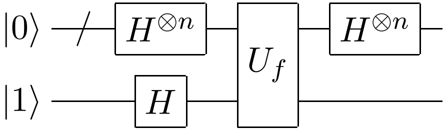
$$|\psi_2\rangle = \begin{cases} \pm\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & \text{if } f(0) = f(1) \\ \pm\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & \text{if } f(0) \neq f(1) \end{cases}$$

Deutsch-Josza algorithm

- After the final *Hadamard*

$$|\psi_3\rangle = \begin{cases} \pm|0\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{if } f(0) = f(1) \\ \pm|1\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{if } f(0) \neq f(1) \end{cases}$$

- $|\psi_3\rangle = \pm|f(0) \oplus f(1)\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$
- Measuring first *qubit* gives $f(0) \oplus f(1)$. Only one evaluation of $f(x)$.
- Faster than is possible with any classical apparatus.
- Can easily be extended to n bits



- For $x \in \{0, \ldots, 2^{n-1}\}$ and $f(x) \in \{0, 1\}$, determined whether $f(x)$ is *constant* or *balanced* with only one application of U_f.

Simon's problem

- Suppose we are given $f : \{0,1\}^n \to \{0,1\}^n$ to which we can make *oracle* calls.
- We are promised that there exists a *secret* string $a \in \{0,1\}^n$ such that:
    1. For all inputs $x \in \{0,1\}^n$, $f(x) = f(x \oplus a)$
    2. For all inputs $x, y \in \{0,1\}^n$, if $x \neq y \oplus a$, then $f(x) \neq f(y)$.
- Problem: find a.
- *Any classical algorithm, deterministic or randomized, needs* $\Omega(2^{n/2})$ *invocations of* f *to solve this problem.* (Show this. Hint: the *birthday paradox*)

Simon's algorithm

- Start with $|0\dots0\rangle|0\dots0\rangle$
- Apply Hadamard/Fourier transform H_{2^n} on the first register to obtain

$$\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}|x\rangle|0\dots0\rangle$$

- Compute $f(x)$ and store in the second register to obtain

$$\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}|x\rangle|f(x)\rangle$$

- Measure register 2 to obtain in the first register

$$\left(\frac{1}{\sqrt{2}}|x\rangle+\frac{1}{\sqrt{2}}|x\oplus a\rangle\right)$$

Clearly contains some information about a, how to extract it?

Simon's algorithm

- Apply Hadamard/Fourier transform H_{2^n} on the first register to obtain $\sum_{z \in \{0,1\}^n} \alpha_z \big| z \big\rangle$, where

$$\alpha_z = \frac{1}{\sqrt{2}} \frac{1}{2^{n/2}} (-1)^{z.x} + \frac{1}{\sqrt{2}} \frac{1}{2^{n/2}} (-1)^{z.(x \oplus a)} = \frac{1}{2^{(n+1)/2}} (-1)^{z.x} [1 + (-1)^{z.a}]$$

- $(-1)^{z.a} = -1 \implies \alpha_z = 0$, and, $(-1)^{z.a} = 1 \implies \alpha_z = \frac{\pm 1}{2^{(n-1)/2}}$.
- Hence if we measure the register, we will definitely see a z such that $z.a = 0$. Thus we get an equation

$$z_1 a_1 + \ldots + z_n a_n = 0 (mod 2)$$

  where $z = (z_1, \ldots, z_n)$ is chosen uniformly at random from $\{0,1\}^n$.
- A simple probablistic analysis shows that a can be obtained with high probablity with $O(n)$ trials.

Simon's algorithm: analysis

- We need $n-1$ linearly independent equations to solve using Gaussian elemination.
- Suppose we already have k linearly independent equations, with associated vectors $z^{(1)}, \ldots, z^{(k)}$. The vectors then span a subspace $S \subseteq \mathbb{Z}_2^n$ of size 2^k.
- Suppose we learn a new vector $z^{(k+1)}$. It lies *outside* S with probability at least $(2^n - 2^k)/2^n = 1 - 2^{k-n}$.
- So the probability that any n equations are independent is

$$\left(1-\frac{1}{2^n}\right)\times\left(1-\frac{1}{2^{n-1}}\right)\times\cdots\times\left(1-\frac{1}{4}\right)\times\left(1-\frac{1}{2}\right) \geq \prod_{k=1}^{\infty}\left(1-\frac{1}{2^k}\right) \approx 0.28879$$

Quantum computing faster?

Do these prove that quantum computing is decidedly faster than classical computing?