On the Power of Quantum Computation

Daniel R. Simon (1995)

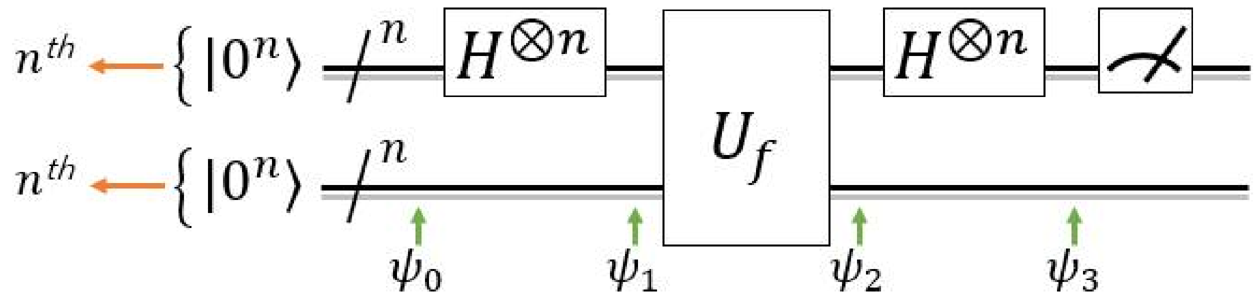
School of Air Transport, Transportation & Logistics **2018310015 Dongsin Kim**

Simon's Problem

In *Simon's problem* we are given a function from n-bit strings to n-bit strings, $f: \{0,1\}^n \to \{0,1\}^n$ which is guaranteed to satisfy $[f(x) = f(y)] \Leftrightarrow [y = x \oplus s]$ for a string $x, y, s \in \{0,1\}^n$. Simon's problem is then, by querying $f(x)$ to determine whether the function belongs to (i) $\mathrm{s} = 0^n$ (i.e. f is *one-to-one function*) or to (ii) $\mathrm{s} \neq 0^n$ (i.e. f is *two-to-one function*).

Simon's Algorithm

Simon's algorithm can be implemented with the following quantum circuit with $2n$-qubits initialized with, respectively, $|0\rangle$:



First, the algorithm starts with $|\psi_0\rangle = |0^n\rangle \otimes |0^n\rangle = |00 \cdots 0\rangle \otimes |00 \cdots 0\rangle = |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes n}$. Analogously to the previous reports, it is clear that

$$|\psi_1\rangle = \left(H^{\otimes n} \otimes I^{\otimes n}\right)|\psi_0\rangle = H^{\otimes n}|00 \cdots 0\rangle \otimes I^{\otimes n}|00 \cdots 0\rangle,$$

$$= \left(\frac{1}{(\sqrt{2})^n} \sum_{x \in \{0,1\}^n} (-1)^{0 \times x_1 + \cdots + 0 \times x_n} |x\rangle\right) \otimes |0\rangle^{\otimes n} = \left(\frac{1}{(\sqrt{2})^n} \sum_{x \in \{0,1\}^n} |x\rangle\right) \otimes |0\rangle^{\otimes n}.$$

$$\because\ H^{\otimes n}|a\rangle = \frac{1}{(\sqrt{2})^n} \sum_{b \in \{0,1\}^n} (-1)^{a_1 b_1 + \cdots + a_n b_n} |b\rangle$$

Second, remembering that $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$:

$$|\psi_2\rangle = U_f|\psi_1\rangle = \left(\frac{1}{(\sqrt{2})^n} \sum_{x \in \{0,1\}^n} |x\rangle\right) \otimes |0 \oplus f(x)\rangle^{\otimes n} = \frac{1}{(\sqrt{2})^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle\, |f(x)\rangle.$$

Finally, the last set of Hadamard gates are applied, which results in state

$$|\psi_3\rangle = \left(H^{\otimes n} \otimes I^{\otimes n}\right)|\psi_2\rangle = \left(\frac{1}{(\sqrt{2})^n} \sum_{x \in \{0,1\}^n} \left(\frac{1}{(\sqrt{2})^n} \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z} |z\rangle |f(x)\rangle\right)\right).$$

Now, we are interested in the probability with which each string results from the measurement. Let us first consider the special case where $\mathrm{s} = 0^n$, which means that f is a one-to-one function.

Case 1: $\mathbf{s}=\mathbf{0}^{\boldsymbol{n}}$. Analogously to Deutsch-Jozsa algorithm, we can write the state from $|\psi_3\rangle$ as

$$\sum_{z\in\{0,1\}^n}|z\rangle\left(\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{x\cdot z}|f(x)\rangle\right).$$

So the probability that the measurement results in each string z is

$$\left\|\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{x\cdot z}|f(x)\rangle\right\|^2=\left\|\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{x\cdot z}|x\rangle\right\|^2=\frac{1}{2^n}.$$

As f is a one-to-one function, the two vectors only differ in the ordering of their entries. The value of the middle-term is more easily seen to be 2^{-n}. Thus, *the outcome is simply a uniformly distributed* n*-bit string when the case 1.*

Case 2: $\mathbf{s}\neq\mathbf{0}^{\boldsymbol{n}}$. The analysis from before still works to imply that the probability to measure any given string z is

$$\left\|\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{x\cdot z}|f(x)\rangle\right\|^2.$$

Let $\mathrm{A}=\mathrm{range}(f)$. If $k\in\mathrm{A}$, there must exist 2 distinct strings $x_k,x_k'\in\{0,1\}^n$ such that $f(x_k)=f(x_k')=k$, and moreover it is necessary that $x_k\oplus x_k'=s$ (which is equivalent to $x_k'=x_k\oplus s$).

$$\left\|\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{x\cdot z}|f(x)\rangle\right\|^2=\left\|\frac{1}{2^n}\sum_{k\in A}\left((-1)^{x_k\cdot z}+(-1)^{x_k'\cdot z}\right)|k\rangle\right\|^2,$$

$$=\left\|\frac{1}{2^n}\sum_{k\in A}\left((-1)^{x_k\cdot z}+(-1)^{(x_k\oplus s)\cdot z}\right)|k\rangle\right\|^2=\left\|\frac{1}{2^n}\sum_{k\in A}(-1)^{x_k\cdot z}(1+(-1)^{s\cdot z})|k\rangle\right\|^2,$$

$$\because\ (x_z\oplus s)\cdot y=(x_z\cdot y)\oplus(s\cdot y)$$

$$=\begin{cases}2^{-(n-1)} & \text{if } s\cdot z=0\\ 0 & \text{if } s\cdot z=1\end{cases}.$$

So, *the measurement always results in some string* z *that satisfies* $s\cdot z=0$, and the distribution is uniform over all of the strings that satisfy this constraint.

Classical post-processing & Conclusions

When we run the circuit above, there are two cases: (i) $\mathrm{s}=0^n$, the measurement results in each string $z\in\{0,1\}^n$ with probability $p_z=2^{-n}$; (ii) $\mathrm{s}\neq0^n$, the probability to obtain each string z is

$$p_z=\begin{cases}2^{-(n-1)} & \text{if } s\cdot z=0\\ 0 & \text{if } s\cdot z=1\end{cases}.$$

Thus, in both cases the measurement results in some string z that satisfies $s\cdot z=0$, and the distribution is uniform over all of the strings that satisfy this constraint.

Specifically, if the above process is run $(n-1)$-times, you will get $(n-1)$-strings $z_1,\dots,z_{n-1}$ such that $s\cdot z_1(=s_1z_{11}+s_2z_{12}+\cdots+s_n\cdot z_{1n})=0$, $\cdots$, $s\cdot z_{n-1}\left(=s_1z_{(n-1)1}+s_2z_{(n-1)2}+\cdots+s_n\cdot z_{1(n-1)n}\right)=0$. *This is a system if* $(n-1)$*-linear equations in* n*-unknowns (the bits of* s*), and the goal is to solve to obtain* s*.*