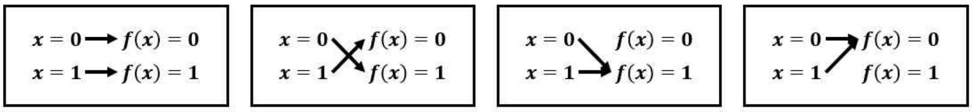
Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer

David Deutsch (1985)

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

Deutsch's Problem

Given a uni-variate function $f:\{0,1\} \rightarrow \{0,1\}$, determine whether f is balanced or constant. A function (f) is balanced if $f(0) \neq f(1)$ and constant if $f(0) = f(1)$. In fact, there only 4 possible functions that meet the condition:



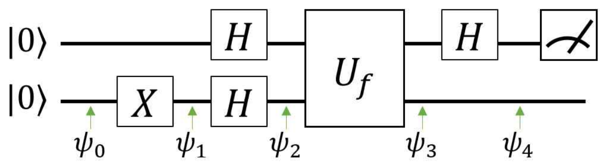
Balanced *Balanced* *Constant* *Constant*

The naive (classical) solution simply requires two-*queries* to determine whether f is balanced or constant (i.e. one to compute $f(0)$ and one to compute $f(1)$). *Can we do better with a quantum computer?* Deutsch showed (in 1985) how to achieve this with just a single-query.

Please note that the *Deutsch's problem* is a special (1-bit) case of the general (n-bit) *Deutch-Jozsa problem*.

Deutsch's Algorithm

Deutsch's algorithm can be implemented with the following quantum circuit with 2-*qubits* (represented by 2-lines) initialized with, respectively, $|0\rangle$:



In quantum mechanics, the general quantum state of a qubit can be represented by a linear *superposition* of its two orthonormal basis states (or basis vectors). These vectors are usually denoted as $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$. These two orthonormal basis states, $\{|0\rangle, |1\rangle\}$, together called the *computational basis*, are said to span the two-dimensional linear vector (*Hilbert*) space of the qubit.

Qubit basis states can also be combined to form product basis states. For example, 2-qubits could be represented in a four-dimensional linear vector space spanned by the following product basis states $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1 \otimes \begin{bmatrix}1\\0\end{bmatrix}\\ 0 \otimes \begin{bmatrix}1\\0\end{bmatrix}\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix},\ |01\rangle = \begin{bmatrix}0\\1\\0\\0\end{bmatrix},\ |10\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix},\ |11\rangle = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

-[Wikipedia "Qubit"](#)에서 참고인용 후 일부 수정-

First, the algorithm starts with $|\psi_0\rangle = |00\rangle$. We apply X-gate to set the second qubit as $|1\rangle$.

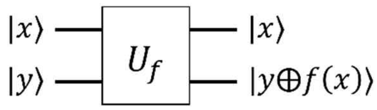
$$|\psi_1\rangle = (I \otimes X)|00\rangle = |01\rangle \;\; \textit{where } X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Second, we apply H-gate to put both qubits, respectively, into superposition states.

$$|\psi_2\rangle = (H \otimes H)|\psi_1\rangle = \frac{1}{2}\left(\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\right)\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)$$

$$= \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = \frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2} \;\; \textit{where } H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Let's assume that for the function f, we have a black-box *unitary* (preserving norm and reversible) gate U_f (also called a *quantum oracle*) that takes an input $|x, y\rangle$ to the state $|x, y \oplus f(x)\rangle$ where $\oplus$ is *XOR* operation. Inner workings of the quantum oracle are unknown to the algorithm:



Case 1: balanced f. By definition, if f is constant, then $f(0) \neq f(1)$. Here, $0 \oplus f(0) = 1 \oplus f(1) = f(0)$ and $0 \oplus f(1) = 1 \oplus f(0) = f(1)$. After U_f is applied:

$$|\psi_3\rangle = U_f\left(\frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2}\right) = \frac{|0\rangle|0 \oplus f(0)\rangle - |0\rangle|1 \oplus f(0)\rangle + |1\rangle|0 \oplus f(1)\rangle - |1\rangle|1 \oplus f(1)\rangle}{2}$$

$$= \frac{(|0\rangle - |1\rangle) \otimes |f(0)\rangle - (|0\rangle - |1\rangle) \otimes |f(1)\rangle}{2} = \frac{1}{2}(|0\rangle - |1\rangle) \otimes (|f(0)\rangle - |f(1)\rangle)$$

Finally, we apply last the H-gate on the first qubit.

$$|\psi_4\rangle = (H \otimes I)|\psi_3\rangle = \left(\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes I\right)\left(\frac{1}{2}\begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes (|f(0)\rangle - |f(1)\rangle)\right) = \frac{1}{\sqrt{2}}|1\rangle \otimes (|f(0)\rangle - |f(1)\rangle)$$

Case 2: Constant f. By definition, if f is balanced, then $f(0) = f(1)$. Here, $0 \oplus f(0) = f(0)$ and $0 \oplus f(1) = f(1)$. Now, we apply U_f and last the H-gate on the first qubit.

$$|\psi_3\rangle = U_f\left(\frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2}\right) = \frac{|0\rangle|0 \oplus f(0)\rangle - |0\rangle|1 \oplus f(0)\rangle + |1\rangle|0 \oplus f(1)\rangle - |1\rangle|1 \oplus f(1)\rangle}{2}$$

$$= \frac{(|0\rangle + |1\rangle) \otimes |f(0)\rangle - (|0\rangle + |1\rangle) \otimes |1 \oplus f(1)\rangle}{2} = \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|f(0)\rangle - |1 \oplus f(1)\rangle)$$

$$|\psi_4\rangle = (H \otimes I)|\psi_3\rangle = \left(\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes I\right)\left(\frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes (|f(0)\rangle - |1 \oplus f(1)\rangle)\right) = \frac{1}{\sqrt{2}}|0\rangle \otimes (|f(0)\rangle - |1 \oplus f(1)\rangle)$$

Conclusions

Therefore, *by measuring the first qubit,* we can answer to the question. If f is balanced, the algorithm outputs 1 and if f is constant, the algorithm outputs 0. Thus, the algorithm decides whether f is constant or balanced, using just a single-query.