C/CS/Phys 191 Deutsch and Deutsch-Josza algorithms 10/25/05
Fall 2005 Lecture 17

# 1 Readings

Benenti et al., Ch. 3.9 - 3.9.2

Stolze and Suter, Quantum Computing, Ch. 8.2 - 8.2.5)

Nielsen and Chuang, Quantum Computation and Quantum Information, Ch. 1.4.3, 1.4.4

# 2 Deutsch's algorithm

Deutsch's algorithm is a perfect illustration of all that is miraculous, subtle, and disappointing about quantum computers. It calculates a solution to a problem faster than any classical computer *ever* can. It illustrates the subtle interaction of superposition, phase-kick back, and interference. Finally, unfortunately, is solves a completely pointless problem.

Deutsch's algorithm answers the following question: suppose $f(x)$ is either constant or balanced, which one is it? If $f(x)$ were constant then for all $x$ the result is either 0 or 1. However, if $f(x)$ were balanced then for one half of the inputs $f(x)$ is 0, and for the other half it is 1 (which x's correspond to 0 or 1 is completely arbitrary). To answer this question classically, we clearly need to query the function for both $x = 0$ and $x = 1$, hence two queries are required. Using a quantum algorithm it turns out that we can solve the problem with just one query of the function.

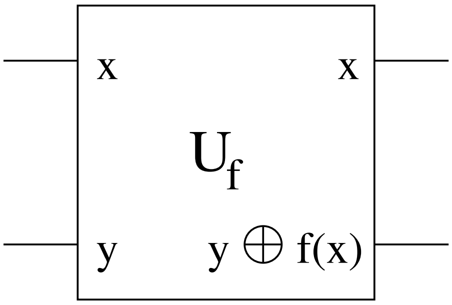


Figure 1: If $U_f$ implements $f$, $x$ is input as $\left(|0\rangle + |1\rangle\right)/\sqrt{2}$ and $y$ as $|0\rangle$, then the output is equal to $\left(|0, f(0)\rangle + |1, f(1)\rangle\right)/\sqrt{2}$. This illustrates the basic feature of parallelism in quantum algorithms.

We begin by illustrating how superposition of quantum state creates *quantum parallelism* or the ability to compute on many states simultaneously.

Given a function $f(x) : \{0,1\} \rightarrow \{0,1\}$ using a quantum computer, use two qubits $|x, y\rangle$ and transform them into $|x, y \oplus f(x)\rangle$ (where $\oplus$ represents addition modular two). We use two qubits since we wish to leave the input $x$ or the query register, "un-changed". The second qubit, $y$, acts as a result register. Let $U_f$ be the unitary transform that implements this. This is illustrated in Figure 1.

Suppose we wish to calculate $f(0)$, then we could input $x$ as $|0\rangle$, and $y$, our output register, as $|0\rangle$ and apply the $U_f$ transform.

The input is written as $\left|0\right\rangle \otimes \left|0\right\rangle = \left|0,0\right\rangle$.

The output is transformed by $U_f$ to be $\left|0, 0 \oplus f(0)\right\rangle$.

Suppose we wish to calculate $f(1)$, then we could input $x$ as $\left|1\right\rangle$, and $y$, our output register, as $\left|0\right\rangle$ and apply the $U_f$ transform.

The input is written as $\left|1\right\rangle \otimes \left|0\right\rangle = \left|1,0\right\rangle$.

The output is transformed by $U_f$ to be $\left|1, 0 \oplus f(1)\right\rangle$.

But this is not a classical computer – we can actually query the results of 0 and 1 simultaneously using quantum parallelism. For this, let $x$ equal $\left(\left|0\right\rangle + \left|1\right\rangle\right)/\sqrt{2}$ and $y$ equal 0.

The input $\left|\psi_1\right\rangle = \frac{\left|0,0\right\rangle + \left|1,0\right\rangle}{\sqrt{2}}$

The output $\left|\psi_2\right\rangle = \frac{\left|0,f(0)\right\rangle + \left|1,f(1)\right\rangle}{\sqrt{2}}$

$\longrightarrow$ Remarkable: $U_f$ is applied to $\left|0\right\rangle$ and $\left|1\right\rangle$ simultaneously! This is known as quantum parallelism.

$\longrightarrow$ Problem: sounds good, but measurement produces either $\left|0, f(0)\right\rangle$ or $\left|1, f(1)\right\rangle$. Hence we need to be clever about what type of question we ask, and how we go about extracting the answer.

The solution is to use another quantum mechanical property: *interference*.

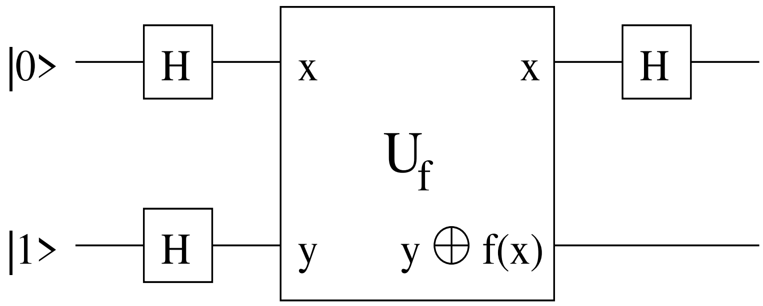


Figure 2: Quantum circuit for Deutsch's algorithm, testing whether a Boolean function on one qubit is constant or balanced.

> **Aside:**
> Deutsch's algorithm, as all known quantum algorithms that provide exponential speedup over classical systems do, answers a question about a global property of a solution space. These are often called *promise problems*, whereby the structure of the solution space is promised to be of some form and by carefully using superposition, entanglement and interference we can extract information about that structure. The reason these problems obtain exponential improvement over all known classical algorithms is that classically one has to calculate every point in the solution space in order to obtain full knowledge about this structure. Quantum mechanically we calculate every point using quantum parallelism. Unfortunately this is often **not** how most algorithms are phrased. Usually we work with problems that are phrased of the form "what $x$ gives a value of $f(x)$ with the desired property?" Thus far, quantum computers can only provide square-root improvement to such query-based problems.

Let $\left|\psi_0\right\rangle$ be the initial state vector and $\left|\psi_1\right\rangle$ be the state of the system prior to applying $U_f$. Let $\left|\psi_2\right\rangle$ be the state of the system after applying $U_f$ and $\left|\psi_3\right\rangle$ be the state of the system prior to measurement.

Input: $|\psi_0\rangle = |0,1\rangle$

It may seem strange to start out with a result register of 1 instead of 0, but ignore this for now, we will return to it shortly. Apply the $H$ gate to the query and result registers to obtain:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

The first qubit is referred to as the 'query' qubit, the second as the 'result' qubit. (Note that the term 'query' is used in two different respects here - querying the function $f(x)$, i.e., evaluating it, and querying the qubit 1, i.e., measuring it.

Now, lets examine $y \oplus f(x)$:

Suppose $f(x) = 0$ then $y \oplus f(x) = y \oplus 0 = \frac{1}{\sqrt{2}}\left(|0\oplus 0\rangle - |1\oplus 0\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$

Suppose $f(x) = 1$ then $y \oplus f(x) = y \oplus 1 = \frac{1}{\sqrt{2}}\left(|0\oplus 1\rangle - |1\oplus 1\rangle\right) = \frac{1}{\sqrt{2}}\left(-|0\rangle + |1\rangle\right)$

Since $\pm 1 = (-1)^{f(x)}$, we can compactly describe this behavior for both instances with the following single formula:

$$y \oplus f(x) = (-1)^{f(x)}\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

Thus, $U_f$ transforms $|x\rangle\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$ into:

$$(-1)^{f(x)}|x\rangle\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

Or we can write it all out in detail:

$$U_f\left[\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)\right] = \frac{1}{2}\left[(-1)^{f(0)}|0\rangle\left(|0\rangle - |1\rangle\right) + (-1)^{f(1)}|1\rangle\left(|0\rangle - |1\rangle\right)\right]$$

Thus is an example of backward sign propagation or kick-back of phase from qubit 2 to qubit 1 - actually a very simple one since the phase is only a global phase here at this point. But below we shall see how it gets moved *into* the state of qubit 1, resulting in a real overall kick-back from qubit 2 to qubit 1.

Now we look at what this state is for the two instances of $f$ being constant or balanced. First, suppose $f$ is constant, that is $f(0) = f(1)$. Then:

$$\begin{aligned}
&\frac{1}{2}\left[(-1)^{f(0)}|0\rangle\left(|0\rangle - |1\rangle\right) + (-1)^{f(1)}|1\rangle\left(|0\rangle - |1\rangle\right)\right] \\
&= \frac{1}{2}(-1)^{f(0)}\left[|0\rangle\left(|0\rangle - |1\rangle\right) + |1\rangle\left(|0\rangle - |1\rangle\right)\right] \\
&= \pm\frac{1}{2}\left[|0\rangle\left(|0\rangle - |1\rangle\right) + |1\rangle\left(|0\rangle - |1\rangle\right)\right] \\
&= \pm\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)
\end{aligned}$$

Since the first qubit is in $|+\rangle$ we can anticipate that performing a Hadamard gate on qubit 1 will then transform this to $|0\rangle$.

Suppose now instead that $f$ is balanced, that is $f(0) \neq f(1)$, then:

$$\begin{aligned}
&\frac{1}{2}\left[(-1)^{f(0)}|0\rangle\left(|0\rangle - |1\rangle\right) + (-1)^{f(1)}|1\rangle\left(|0\rangle - |1\rangle\right)\right] \\
&= \frac{1}{2}\left[(-1)^{f(0)}|0\rangle\left(|0\rangle - |1\rangle\right) + (-1)\times(-1)^{f(0)}|1\rangle\left(|0\rangle - |1\rangle\right)\right] \\
&= \frac{1}{2}(-1)^{f(0)}\left[|0\rangle\left(|0\rangle - |1\rangle\right) - |1\rangle\left(|0\rangle - |1\rangle\right)\right] \\
&= \pm\frac{1}{2}\left[|0\rangle\left(|0\rangle - |1\rangle\right) - |1\rangle\left(|0\rangle - |1\rangle\right)\right] \\
&= \pm\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)
\end{aligned}$$

Since the first qubit is in $|-\rangle$ here, we can anticipate that performing a Hadamard gate on qubit 1 in this case will transform this to $|1\rangle$.

So it seems we can get orthogonal states for qubit 1 for the two different instances. So lets now run the $|x\rangle$ qubit through an $H$ gate to get $|\psi_3\rangle$:

$$|\psi_3\rangle = \begin{array}{ll} \pm\frac{1}{\sqrt{2}}|0\rangle\left(|0\rangle - |1\rangle\right) & if \quad f(0) = f(1) \\ \pm\frac{1}{\sqrt{2}}|1\rangle\left(|0\rangle - |1\rangle\right) & if \quad f(0) \neq f(1) \end{array}$$

Since in our case $f(0) \oplus f(1) = 0 \Leftrightarrow f(0) = f(1)$ we can write this as

$$|\psi_3\rangle = \pm|f(0) \oplus f(1)\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

Hence it is possible to measure $x$ (the first, 'query', qubit) to find $f(0) \oplus f(1)$.

**Aside:**
Note that $f(0) \oplus f(1)$ is a global property of $f(x)$. Classically it would require two evaluations of $f(x)$ to find this answer. Using a quantum computer we are able to evaluate both answers simultaneously and then interfere these answers to combine them together. Another more subtle point is that the phase of the result qubit transfers to the query qubit. This is a special case of phase kick back. In effect, the query qubit acts as a control of whether or not to flip the result qubit. While the result qubit is potentially flipped by the state of the query qubit, the phase of the query qubit is altered by the phase of the result (or target) qubit! This property is also critical to Shor's algorithm.

# 3 Deutsch-Josza algorithm

The Deutsch-Jozsa algorithm is a generalization of Deutsch's algorithm to Boolean functions on $n$ qubits.

Suppose $f(x) : \{2^n\} \rightarrow \{0,1\}$ and that f is either constant or balanced. The goal is determine which one it is. Classically it is easy to see that this would require (in worst case) querying just over half the solution space, or $2^n/2+1$ queries. The Deutsch-Jozsa algorithm answers this question with just *one* query!

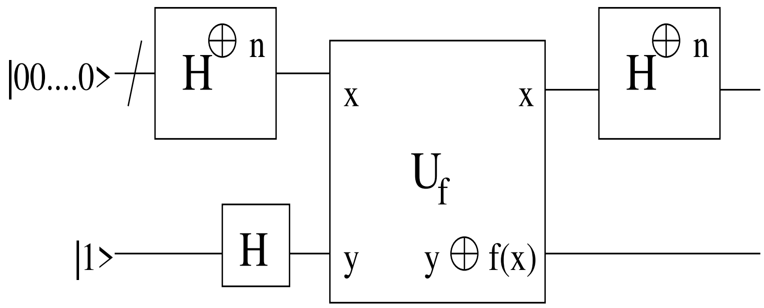


Figure 3: Quantum circuit for the Deutsch-Jozsa algorithm, an $n$-qubit generalization of Deutsch's algorithm

We need $n$ qubits and one additional qubit: the former are the analog of $x$ and constitute a query register, while the latter corresponds to the result qubit $y$ in the Deutsch algorithm. The starting state of the system $|\psi_0\rangle$ is fairly straightforward

$$|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle$$

The symbolic notation $|0\rangle^{\otimes n}$ simply means $n$ consecutive $|0\rangle$ qubits.

We then apply the $H^{\otimes n}$ transform. This symbol means to apply the $H$ gate to each of the $n$ qubits (in parallel, although this does not matter. The key is only that the $H$ gate is applied once to each qubit). In Homework 3 you showed that this transform is:

$H^{\otimes n}|i\rangle = \sum_j \frac{(-1)^{i\cdot j}}{\sqrt{2^n}}|j\rangle$

**Here is another proof of this important relation**: Consider first an $H$ gate applied to a single qubit $|x\rangle$. We need to multiply the component $|1\rangle$ by 1 if $x=0$ and by $-1$ if $x=1$: these two procedures can be combined as in the Deutsch algorithm above by simply multiplying by the factor $(-1)^{f(x)}$. Then

$$
\begin{aligned}
H|x\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x|1\rangle\right) \\
&= \frac{1}{\sqrt{2}}\sum_z (-1)^{xz}|z\rangle
\end{aligned}
$$

where here $z$ spans only two values, 0 and 1. This seems like notational overkill to represent a simple Hadamard gate, $H$. However, when we generalize the latter to $H^{\otimes n}$ the notation pays off since the above form can immediately be generalized by summing over all possible combinations of qubit basis states, i.e., over all $n$-qubit states $\vec{z}$:

$$
H^{\otimes n}|\vec{x}\rangle = \frac{1}{\sqrt{2^n}}\sum_{\vec{z}}(-1)^{\vec{x}\cdot\vec{z}}|\vec{z}\rangle .
$$

Each qubit contributes an independent phase term, leading to the dot product $\vec{x}\cdot\vec{z}$.

Coming back to Deutsch-Jozsa, we now transform $|\psi_0\rangle$ with the $n$-qubit and 1-qubit Hadamard gates as:

$$
|\psi_1\rangle = H^{\otimes n}|0\rangle^{\otimes n} H|1\rangle
$$

$$
= \sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^n}}|x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]
$$

Note that the notation $\{0,1\}^n$ means all possible bit strings of size $n$. For example, for $n=2$, we have “00”, “01”, “10”, and “11”.

We then apply the transform $U_f$ that implements $f(x)$ to obtain the state $|\psi_2\rangle$:

$$
|\psi_2\rangle = \sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)}}{\sqrt{2^n}}|x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]
$$

Finally we apply another $H^{\otimes n}$ transform to obtain $|\psi_3\rangle$:

$$
|\psi_3\rangle = \sum_{z\in\{0,1\}^n}\sum_{x\in\{0,1\}^n} \frac{(-1)^{x\cdot z + f(x)}}{2^n}|z\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]
$$

The key to the Deutsch-Jozsa algorithm is the following rather subtle point. We measure the probability amplitude of $z = |0\rangle^{\otimes n}$, the all zero state. (If the qubits are spins in a magnetic field, this would be the ground state - correct?) Consider the instance when $f(x)$ is constant. Since $z = |0\rangle^{\otimes n}$, $x\cdot z$ must also be equal to zero and hence $(-1)^{x\cdot z + f(x)}$ is either $-1$ or $+1$ for all values of x, where $-1$ holds for $f(x)=1$ and 1 holds for $f(x)=0$. In this case the amplitude for $z = |0\rangle^{\otimes n}$ is

$$
\pm \sum_{x\in\{0,1\}^n} \frac{1}{2^n} = \pm 1
$$

which exhausts the probability amplitude for $\psi_3$. In other words, since $\psi_3$ is normalized to 1 and the amplitude of $z = |0\rangle^{\otimes n}$ already gives probability 1, there can be no other component in $\psi_3$ - all other amplitudes must be zero. Hence when you measure the first $n$ qubits in the query register, you will obtain a zero (or more correctly, $0^{\otimes n}$).

Conversely, if $f(x)$ is balanced then $(-1)^{x\cdot z+f(x)}$ will be $+1$ for some values of $x$ and $-1$ for other values of x. This is where the balanced requirement comes into play. Since all possible x' values are considered and the function is perfectly balanced, one must have equal numbers of $+1$ and $-1$. The amplitude of the all zero state $z = |0\rangle^{\otimes n}$ is then:

$$\sum_{x_1} \frac{+1}{2^n} + \sum_{x_2} \frac{-1}{2^n} = 0$$

where $x_1$ is the set of x's such that $f(x)$ is equal to 0 and $x_2$ is the set of x's where $f(x)$ is equal to 1. Hence you will not measure the all zero eigenvalue $0^{\otimes n}$ when $f(x)$ is balanced since the probability amplitudes interfere destructively to produce a net probability amplitude of zero for the all zero state.

What will be measured if the function is balanced? Anything except the all zero eigenvalue. At least one qubit will result in a measurement value of 1.

Note that this algorithm does require only one query of $f(x)$, i.e., of $U_f$, but it requires the ability to make an $n$-qubit measurement.