balanced. So determining \( f(0) \oplus f(1) \) is equivalent to determining whether the function \( f \) is constant or balanced.

### The Deutsch Problem

**Input:** A black box for computing an unknown function \( f : \{0,1\} \to \{0,1\} \).

**Problem:** Determine the value of \( f(0) \oplus f(1) \) by making queries to \( f \).

How many queries to the oracle for \( f \) must be made classically to determine \( f(0) \oplus f(1) \)? Clearly the answer is 2. Suppose we compute \( f(0) \) using one (classical) query. Then the value of \( f(1) \) could be 0, making \( f(0) \oplus f(1) = 0 \), or the value of \( f(1) \) could be 1, making \( f(0) \oplus f(1) = 1 \). Without making a second query to the oracle to determine the value of \( f(1) \), we can make no conclusion about the value of \( f(0) \oplus f(1) \). The Deutsch algorithm is a quantum algorithm capable of determining the value of \( f(0) \oplus f(1) \) by making only a single query to a quantum oracle for \( f \).

The given reversible circuit for \( f \) can be made into a quantum circuit, by replacing every reversible classical gate in the given circuit with the analogous unitary quantum gate. This quantum circuit can be expressed as a unitary operator

\[
U_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle.
\]

(6.3.1)

Having created a quantum version of the circuit for \( f \), we can supply quantum bits as inputs. We define \( U_f \) so that if we set the second input qubit to be in the state \( |y\rangle = |0\rangle \), then \( |x\rangle = |0\rangle \) in the first input qubit will give \( |0 \oplus f(0)\rangle = |f(0)\rangle \) in the second output bit, and \( |x\rangle = |1\rangle \) in the first input qubit will give \( |f(1)\rangle \). So we can think of \( |x\rangle = |0\rangle \) as a quantum version of the (classical) input bit 0, and \( |x\rangle = |1\rangle \) as a quantum version of the input bit 1. Of course, the state of the input qubit can be some superposition of \( |0\rangle \) and \( |1\rangle \). Suppose, still keeping the second input qubit \( |y\rangle = |0\rangle \), we set the first input qubit to be in the superposition state

\[
\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle.
\]

(6.3.2)

Then the two qubit input to \( U_f \) is

\[
\left( \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \right)|0\rangle \quad \text{(6.3.3)}
\]

and

\[
\frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|0\rangle.
\]

(6.3.4)
The output of $U_f$ will be the state

$$U_f \left( \frac{1}{\sqrt{2}} |0\rangle |0\rangle + \frac{1}{\sqrt{2}} |1\rangle |0\rangle \right) \tag{6.3.5}$$

$$= \frac{1}{\sqrt{2}} U_f |0\rangle |0\rangle + \frac{1}{\sqrt{2}} U_f |1\rangle |0\rangle \tag{6.3.6}$$

$$= \frac{1}{\sqrt{2}} |0\rangle |0\rangle + f(0) + \frac{1}{\sqrt{2}} |1\rangle |0\rangle + f(1) \tag{6.3.7}$$

$$= \frac{1}{\sqrt{2}} |0\rangle |f(0)\rangle + \frac{1}{\sqrt{2}} |1\rangle |f(1)\rangle. \tag{6.3.8}$$

In some sense, $U_f$ has simultaneously computed the value of $f$ on both possible inputs 0 and 1 in superposition. However, recalling how quantum measurement works from Section 3.4, if we now measure the output state in the computational basis, we will observe either $|0\rangle |f(0)\rangle$ (with probability $\frac{1}{2}$), or $|1\rangle |1 \oplus f(1)\rangle$ (with probability $\frac{1}{2}$). After the measurement, the output state will be either $|f(0)\rangle$ or $|f(1)\rangle$, respectively, and so any subsequent measurements of the output state will yield the same result. So this means that although we have successfully computed two values in superposition, only one of those values is accessible through a quantum measurement in the computational basis. Fortunately, this is not the end of the story.

Recall that for the Deutsch problem we are ultimately not interested in individual values of $f(x)$, but wish to determine the value of $f(0) \oplus f(1)$. The Deutsch algorithm illustrates how we can use quantum interference to obtain such global information about the function $f$, and how this can be done more efficiently than is possible classically. The Deutsch algorithm is implemented by the quantum circuit shown in Figure 6.8.

Note that the second input bit has been initialized to the state $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$. This state can easily be created from the state $|1\rangle$ by applying a single Hadamard gate. We do not show this gate, however, to emphasize a certain symmetry that is characteristic of these algorithms. A convenient way to analyse the behaviour of a quantum algorithm is to work through the state at each stage of the circuit. First, the input state is

$$|0\rangle \quad \xrightarrow{H} \quad \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad \xrightarrow{U_f} \quad \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad \xrightarrow{H} \quad |\psi_3\rangle$$

![Fig. 6.8 A circuit implementing the Deutsch algorithm. The measured value equals $f(0) \oplus f(1)$.](image)

After the first Hadamard gate is applied to the first qubit, the state becomes

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \tag{6.3.10}$$

$$= \frac{1}{\sqrt{2}} |0\rangle \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) + \frac{1}{\sqrt{2}} |1\rangle \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right). \tag{6.3.11}$$

Recalling Equation (6.2.15), after applying the $U_f$ gate we have the state

$$|\psi_2\rangle = \frac{(-1)^{f(0)}}{\sqrt{2}} |0\rangle \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) + \frac{(-1)^{f(1)}}{\sqrt{2}} |1\rangle \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \tag{6.3.12}$$

$$= \frac{(-1)^{f(0)}}{\sqrt{2}} |0\rangle \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) + \frac{(-1)^{f(1)}}{\sqrt{2}} |1\rangle \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \tag{6.3.13}$$

$$= (-1)^{f(0)} \left( \frac{|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \right) \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \tag{6.3.14}$$

where the last equality uses the fact that $(-1)^{f(0)}(-1)^{f(1)} = (-1)^{f(0) \oplus f(1)}$.

If $f$ is a constant function (i.e. $f(0) \oplus f(1) = 0$), then we have

$$|\psi_2\rangle = (-1)^{f(0)} \left( \frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \tag{6.3.15}$$

and in the final Hadamard gate on the first qubit transforms the state to

$$|\psi_3\rangle = (-1)^{f(0)} |0\rangle \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right). \tag{6.3.16}$$

The squared norm of the basis state $|0\rangle$ in the first qubit is 1. This means that for a constant function a measurement of the first qubit is certain to return the value $0 = f(0) \oplus f(1)$.

If $f$ is a balanced function (i.e. $f(0) \oplus f(1) = 1$), then we have

$$|\psi_2\rangle = (-1)^{f(0)} \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \tag{6.3.17}$$

and in the final Hadamard gate on the first qubit transforms the state to

$$|\psi_3\rangle = (-1)^{f(0)} |1\rangle \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right). \tag{6.3.18}$$

In this case the squared norm of the basis state $|1\rangle$ in the first qubit is 1. This means that for a balanced function a measurement of the first qubit is certain to return the value $1 = f(0) \oplus f(1)$. So a measurement of the first qubit at the
Fig. 6.9 The circuit for Deutsch's algorithm with the c-\(\hat{U}f(x)\) drawn instead of \(\hat{U}f(x)\). When c-\(\hat{U}f(x)\) is applied, the control qubit is in a superposition of \(|0\rangle\) and \(|1\rangle\), which pick up phase factors of \((-1)^{f(0)}\) and \((-1)^{f(1)}\), corresponding to the eigenvalues of \(\hat{U}f(x)\) for \(x = 0\) and \(1\), respectively. The Hadamard gate followed by a measurement in the computational basis determines the relative phase factor between \(|0\rangle\) and \(|1\rangle\), and thus whether the function is constant or balanced.

To gain some insight into how the Deutsch algorithm can generalize, it is helpful to remember that the operator \(U_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\) in the Deutsch algorithm can be viewed as a single-qubit operator \(\hat{U}f(x)\), whose action on the second qubit is controlled by the state of the first qubit (see Figure 6.9). The state \(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\) is an eigenstate of \(\hat{U}f(x)\) with eigenvalue \((-1)^{f(x)}\). By encoding these eigenvalues in the phase factors of the control qubit, we are able to determine \(f(0) \oplus f(1)\) by determining the relative phase factor between \(|0\rangle\) and \(|1\rangle\).

Distinguishing \(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\) and \(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\) is done using the Hadamard gate.

We will see this technique of associating phase factors (corresponding to eigenvalues) with the control register, and then using quantum interference to determine the relative phase, applied throughout this chapter and the next chapter.

### Exercise 6.3.1

In the Deutsch algorithm, when we consider \(U_f\) as a single-qubit operator \(\hat{U}f(x)\), \(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\) is an eigenstate of \(\hat{U}f(x)\), whose associated eigenvalue gives us the answer to the Deutsch problem. Suppose we were not able to prepare this eigenstate directly. Show that if we instead input \(|0\rangle\) to the target qubit, and otherwise run the same algorithm, we get an algorithm that gives the correct answer with probability \(\frac{1}{2}\) (note this also works if we input \(|1\rangle\) to the second qubit). Furthermore, show that with probability \(\frac{1}{2}\) we know for certainty that the algorithm has produced the correct answer.

**Hint:** Write \(|0\rangle\) in the basis of eigenvectors of \(U_f\).

**Note:** Deutsch originally presented his algorithm in terms of the \(U_f\) operator with \(|0\rangle\) input to the second qubit. Shor analyzed his algorithm for finding orders (factoring) in an analogous manner. Later, it was found that analyzing these algorithms in the eigenbasis of a suitable controlled operator is often convenient (Appendix A.6 discusses this issue; the operators are usually different from the \(U_f(x)\) operators we describe in this exercise). Note that for many algorithms (including the algorithm for finding,

---

### 6.4 The Deutsch–Jozsa Algorithm

The Deutsch–Jozsa algorithm solves a problem that is a straightforward generalization of the problem solved by the Deutsch algorithm. The algorithm has exactly the same structure. As with the Deutsch algorithm, we are given a reversible circuit implementing an unknown function \(f\), but this time \(f\) is a function from \(n\)-bit strings to a single bit. That is,

\[
f : \{0, 1\}^n \rightarrow \{0, 1\}.
\]  

(6.4.1)

We are also given the promise that \(f\) is either constant (meaning \(f(x)\) is the same for all \(x\)), or \(f\) is balanced (meaning \(f(x) = 0\) for exactly half of the input strings \(x\), and \(f(x) = 1\) for the other half of the inputs). The problem here is to determine whether \(f\) is constant, or balanced, by making queries to the circuit for \(f\).

#### The Deutsch–Jozsa Problem

**Input:** A black-box for computing an unknown function \(f : \{0, 1\}^n \rightarrow \{0, 1\}\).

**Promise:** \(f\) is either constant or a balanced function.

**Problem:** Determine whether \(f\) is constant or balanced by making queries to \(f\).

Consider solving this problem by a classical algorithm. Suppose we have used the oracle to determine \(f(x)\) for exactly half of the possible inputs \(x\) (i.e. you have made \(2^{n-1}\) queries to \(f\)), and that all queries have returned \(f(x) = 0\). At this point, we would strongly suspect that \(f\) is constant. However, it is possible that if we queried \(f\) on the remaining \(2^{n-1}\) inputs, we might get \(f(x) = 1\) each time. So it is still possible that \(f\) is balanced. So in the worst case, using a classical algorithm we cannot decide with certainty whether \(f\) is constant or balanced using any less than \(2^{n-1} + 1\) queries. The property of being constant or balanced is a global property of \(f\). As for the Deutsch problem, a quantum algorithm can take advantage of quantum superposition and interference to determine this global property of \(f\). The Deutsch–Jozsa algorithm will determine whether \(f\) is constant, or balanced, making only one query to a quantum version of the reversible circuit for \(f\).

Analogous to what we did for the Deutsch algorithm, we will define the quantum operation

\[
U_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle.
\]  

(6.4.2)

This time we write \(x\) in boldface, because it refers to an \(n\)-bit string. As before, we think of \(U_f\) as a 1-qubit operator \(U_f(x)\), this time controlled by the register
of qubits in the state $|x\rangle$. We can see that $|0\rangle - |1\rangle / \sqrt{2}$ is an eigenstate of $\hat{U}_{f(x)}$ with eigenvalue $(-1)^{f(x)}$.

The circuit for the Deutsch–Jozsa algorithm is shown in Figure 6.10.

Notice the similarity between the circuit for the Deutsch algorithm, and the circuit for the Deutsch–Jozsa algorithm. In place of a simple 1-qubit Hadamard gate, we now have tensor products of $n$ 1-qubit Hadamard gates (acting in parallel). This is denoted $H^\otimes n$. We use $|0\rangle^\otimes n$, or $|0\rangle$ to denote the state that is the tensor product of $n$ qubits, each in the state $|0\rangle$.

As did for the Deutsch algorithm, we follow the state through the circuit. Initially the state is

$$|\psi_0\rangle = |0\rangle^\otimes n \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right). \quad (6.4.3)$$

Consider the action of an $n$-qubit Hadamard transformation on the state $|0\rangle^\otimes n$:

$$H^\otimes n |0\rangle^\otimes n = \left(\frac{1}{\sqrt{2}}\right)^n \left(|0\rangle + |1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + |1\rangle\right). \quad (6.4.4)$$

By expanding out the tensor product, this can be rewritten as

$$H^\otimes n |0\rangle^\otimes n = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle. \quad (6.4.5)$$

This is a very common and useful way of writing this state; the $n$-qubit Hadamard gate acting on the $n$-qubit state of all zeros gives a superposition of all $n$-qubit basis states, all with the same amplitude $\frac{1}{\sqrt{2^n}}$ (called an `equally weighted superposition`). So the state immediately after the first $H^\otimes n$ in the Deutsch–Jozsa algorithm is

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right). \quad (6.4.6)$$

Notice that the query register is now in an equally weighted superposition of all the possible $n$-bit input strings. Now consider the state immediately after the $\hat{U}_f$ (equivalently the c-$\hat{U}_{f(x)}$) gate. The state is

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

$$= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \quad (6.4.7)$$

where we have associated the phase shift of $(-1)^{f(x)}$ with the first qubit (recall Section 6.2).

In the analysis of the state after the interference is completed by the second Hadamard gate, consider the action of the $n$-qubit Hadamard gate on an $n$-qubit basis state $|x\rangle$.

It is easy to verify that the effect of the 1-qubit Hadamard gate on a 1-qubit basis state $|x\rangle$ can be written as

$$H|x\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^x |1\rangle\right) \quad (6.4.8)$$

$$= \frac{1}{\sqrt{2}} \sum_{z \in \{0,1\}} (-1)^{x z} |z\rangle. \quad (6.4.9)$$

Then we can see that the action of the Hadamard transformation on an $n$-qubit basis state $|x\rangle = |x_1\rangle|x_2\rangle \cdots |x_n\rangle$ is given by

$$H^\otimes n |x\rangle = H^\otimes n (|x_1\rangle|x_2\rangle \cdots |x_n\rangle)$$

$$= H|x_1\rangle H|x_2\rangle \cdots H|x_n\rangle$$

$$= \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^{x_1} |1\rangle\right) \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^{x_2} |1\rangle\right) \cdots \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^{x_n} |1\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{z_1, z_2, \ldots, z_n \in \{0,1\}^n} (-1)^{z_1 x_1 + z_2 x_2 + \cdots + z_n x_n} |z_1\rangle|z_2\rangle \cdots |z_n\rangle. \quad (6.4.13)$$
Exercise 6.4.1 Prove that
\[
\frac{1}{\sqrt{2}} \sum_{z_1,z_2, \ldots, z_n \in \{0,1\}^n} (-1)^{z_1 x_1 + z_2 x_2 + \cdots + z_n x_n} |z_1>|z_2> \cdots |z_n>. \tag{6.4.14}
\]

The above equation above can be written more succinctly as
\[
H^{\otimes n}|x> = \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z}|z> \tag{6.4.16}
\]
where \(x \cdot z\) denotes the bitwise inner product of \(x\) and \(z\), modulo 2 (we are able to reduce modulo 2 since \((-1)^2 = 1\)). Note that addition modulo 2 is the same as the XOR operation. The state after the final \(n\)-qubit Hadamard gate in the Deutsch–Jozsa algorithm is
\[
|\psi_3> = \left( \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} \frac{1}{\sqrt{2}} \sum_{z \in \{0,1\}^n} (-1)^{x \cdot z}|z> \right) \left( \frac{|0> - |1>}{\sqrt{2}} \right)
\]
\[
= \frac{1}{2^n} \sum_{z \in \{0,1\}^n} \left( \sum_{x \in \{0,1\}^n} (-1)^{f(x) + x \cdot z} \right)|z> \left( \frac{|0> - |1>}{\sqrt{2}} \right). \tag{6.4.17}
\]

At the end of the algorithm a measurement of the first register is made in the computational basis (just as was done for the Deutsch algorithm). To see what happens, consider the total amplitude (coefficient) of \(|z> = \vert 0 \rangle \otimes z\) in the first register of state \(|\psi_3>\). This amplitude is
\[
\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)}. \tag{6.4.18}
\]

Consider this amplitude in the two cases: \(f\) constant and \(f\) balanced. If \(f\) is constant, the amplitude of \(|0 \rangle \otimes z\) is either +1 or -1 (depending on what value \(f(0)\) takes). So if \(f\) is constant, a measurement of the first register is certain to return all \(0\)'s (by 'all 0's' we mean the binary string \(00 \cdots 0\)). On the other hand, if \(f\) is balanced, then it is easy to see that the positive and negative contributions of the amplitudes cancel, and the overall amplitude of \(|0 \rangle \otimes z\) is 0. So if \(f\) is balanced, a measurement of the first register is certain not to return all \(0\)'s. So to determine whether \(f\) is constant or balanced, the first register is measured. If the result of the measurement is all \(0\)'s, then the algorithm outputs 'constant', and otherwise it outputs 'balanced'.

Exercise 6.4.2

(a) Show that a probabilistic classical algorithm making 2 evaluations of \(f\) can with probability at least \(\frac{2}{3}\) correctly determine whether \(f\) is constant or balanced.

Hint: Your guess does not need to be a deterministic function of the results of the two queries. Your result should not assume any particular a priori probabilities of having a constant or balanced function.

(b) Show that a probabilistic classical algorithm that makes \(O(n)\) queries can with probability at least \(1 - \frac{1}{n}\) correctly determine whether \(f\) is constant or balanced.

Hint: Use the Chernoff bound (Appendix A.1).

It is worth noting that although deterministic classical algorithms would require \(2^n + 1\) queries in the worst case (compared to only 1 query for this quantum algorithm), as shown in Exercise 6.4.2, a probabilistic classical algorithm could solve the Deutsch–Jozsa problem with probability of error at most \(\frac{1}{4}\) using 2 queries. The probability of error can be reduced to less than \(\frac{1}{2}\) with only \(n + 1\) queries. So although there is an exponential gap between deterministic classical and 'exact' quantum query complexity (see Definitions 9.4.1. and 9.4.2.), the gap between classical probabilistic query complexity and the quantum computational query complexity is constant in the case of constant error, and can be amplified to a linear gap in the case of exponentially small error. The next section gives one of the first examples where a quantum algorithm can solve a problem with a polynomial number of queries, where any classical algorithm would require an exponential number of queries even to succeed with bounded error.

6.5 Simon's Algorithm

Consider a function \(f: \{0,1\}^n \rightarrow X\), for some finite set \(X\), where we have the promise that there is some 'hidden' string \(s = s_1s_2 \cdots s_n\) so that \(f(x) = f(y)\) if and only if \(x = y \oplus s\). In this section we will treat the domain \(\{0,1\}^n\) of \(f\) as the vector space\(^1\) \(Z_2^n\) over \(Z_2\) (in general, one can treat it as additive group). For convenience, we will assume that \(X \subseteq \{0,1\}^n\).

\(^1\)To avoid potential confusion, it is worth pointing out that we are talking about two different types of vector spaces. On the one hand, we are referring to the vector space \(\mathbb{R}^n\) over \(\mathbb{R}\), which consists of \(n\)-tuples of \(\mathbb{R}\) and is. This vector space has dimension \(n\) since it can be generated by the \(n\) linearly independent vectors consisting of \(n\)-tuples with exactly one 1 in the \(i\)th position, for \(i = 1, 2, \ldots, n\). The quantum algorithm is executed in a complex vector (i.e. Hilbert) space whose basis elements are labelled by the elements of the vector space \(\mathbb{R}^n\). This Hilbert space has dimension \(2^n\).