Quantum Computing (CST Part II)

Lecture 7: Deutsch-Jozsa Algorithm

Computer programming is an art form,
like the creation of poetry or music.
Donald Knuth

Quantum algorithms

We have seen that using quantum states can lead to tangible advantages in certain information processing tasks.

We now turn our attention to the potential benefits of using quantum information in algorithms: that is, quantum computing proper. We have seen that quantum computing does not violate the Church-Turing thesis, and so our attention will be on the extent to which quantum computing can speed-up computational tasks.

The story starts with an algorithm discovered in 1985 by David Deutsch, which was the first to display a computational advantage compared to the best possible corresponding classical algorithm.

Binary numbers and quantum states

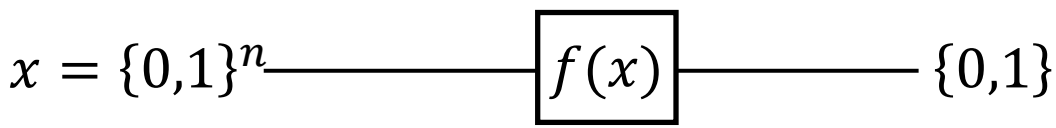
It is usually convenient to express the action of a quantum circuit for any computational basis state and then use linearity to express a sum of superposed terms if necessary. To do so, we analyse the circuit for a general n-qubit computational basis state, $|x\rangle$:

$$|x\rangle = |x_1 x_2 \cdots x_n\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle$$

where $x_i \in \{0, 1\}$ for $i = 1 \cdots n$ and associated with $|x\rangle$ is the n-bit binary number x.

Computing mathematical functions on quantum computers

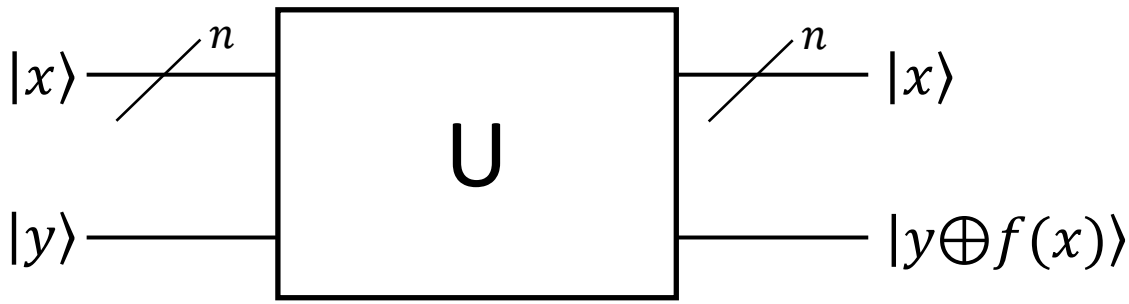
In order to design quantum algorithms we need to know how to execute functions on a quantum computer. In particular, we are interested in functions that take a (binary) number, and output a truth value (i.e., $\{0,1\}$). That is, functions of the form $f : \{0,1\}^n \to \{0,1\}$.



However, we know that a quantum circuit is composed of quantum gates, which are unitary operations, and apart from for trivial functions, this is not a unitary.

Arbitrary mathematical functions as unitaries

Instead, we must use the same trick that allowed us to write down the Toffoli gates as a quantum (unitary) version of the classical AND gate: as well as evaluating the function we output the input data:

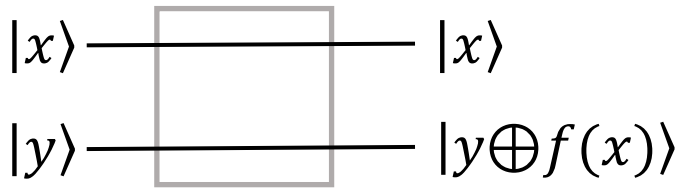
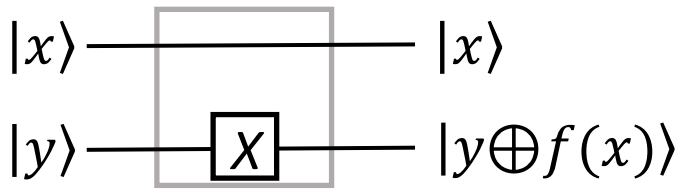
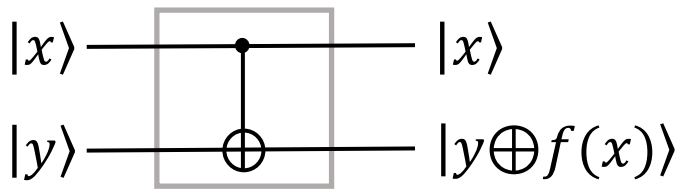
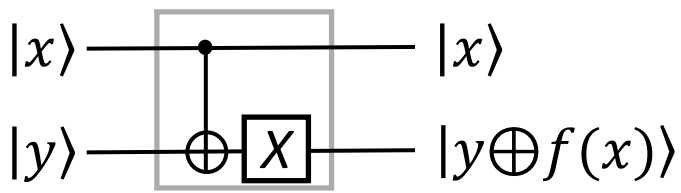


U can easily be seen to be self-inverse, and the universality of quantum computing implies that any function $f(x)$ can be efficiently encoded in this way (to desired accuracy) as a quantum circuit consisting of gates from a finite universal set.

The crossed through wire labelled "n" in the circuit denotes a bundle of n qubits, sometimes termed a n-qubit register. It is also worth pointing out that the same principle holds when the output of $f(x)$ is not restricted to be a single bit, but rather can be n_f bits, and thus the second wire is itself a n_f-qubit register.

Constant and balanced function on a single bit

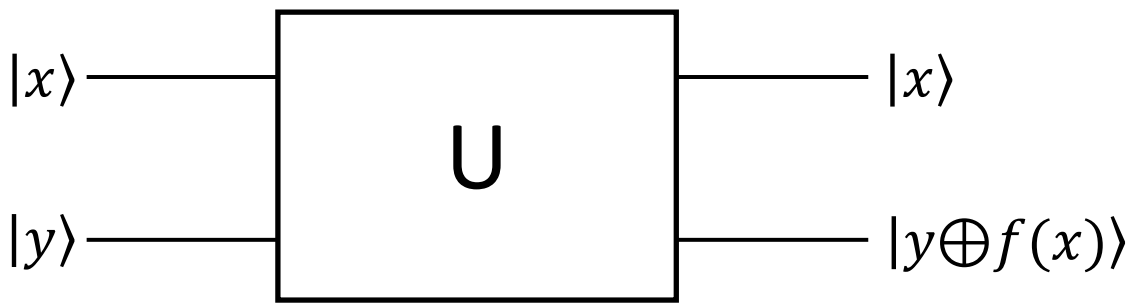
If the input, x, is a single bit (as is the output), then we have four possible functions:

Function	x	$f(x)$	**Type**	**Unitary**
$f(x) = 0$	0 1	0 0	Constant	
$f(x) = 1$	0 1	1 1	Constant	
$f(x) = x$	0 1	0 1	Balanced	
$f(x) = x \oplus 1$	0 1	1 0	Balanced	

Deutsch's algorithm set-up

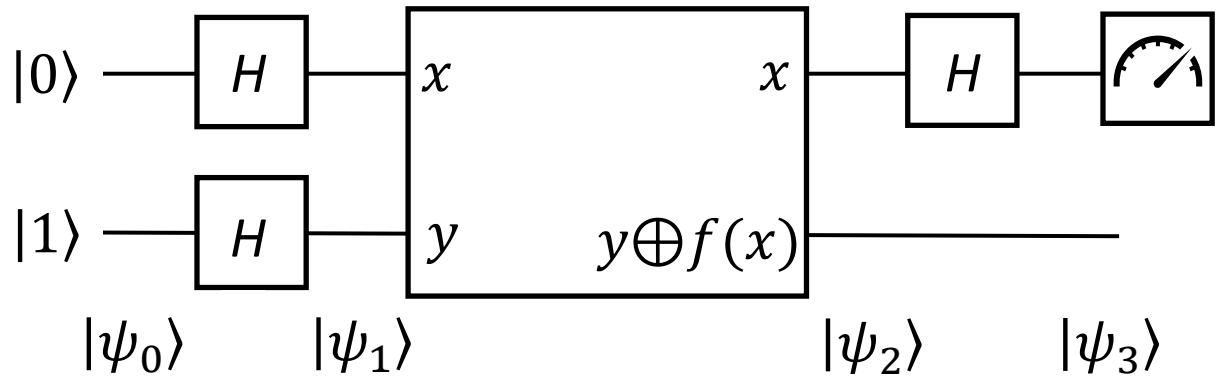
We want to find out whether a particular function, with one input bit and one output bit is constant or balanced. Classically, we need to evaluate the function *twice* (i.e., for input $= 0$ and input $= 1$), but remarkably, we only need to evaluate the function *once* quantumly, by using Deutsch's algorithm.

We have a two qubit unitary, which is one of the four on the previous slide (we don't know which):



Which we are going to incorporate into a quantum circuit.

Deutsch's algorithm (1)



Initially we prepare the state:

$$|\psi_0\rangle = |01\rangle$$

Which the initial Hadamard gates put in the superposition state:

$$\begin{aligned} |\psi_1\rangle &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right) \end{aligned}$$

Deutsch's algorithm (2)

$$|\psi_1\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right)$$

Next the unitary is implemented, which sets the second qubit to $y \oplus f(x)$, so we have four options for $|\psi_2\rangle$:

$$\begin{array}{ll} f(x) = 0 & |\psi_2\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |10\rangle - |11\rangle\right) \\ f(x) = 1 & |\psi_2\rangle = \frac{1}{2}\left(|01\rangle - |00\rangle + |11\rangle - |10\rangle\right) \\ f(x) = x & |\psi_2\rangle = \frac{1}{2}\left(|00\rangle - |01\rangle + |11\rangle - |10\rangle\right) \\ f(x) = x \oplus 1 & |\psi_2\rangle = \frac{1}{2}\left(|01\rangle - |00\rangle + |10\rangle - |11\rangle\right) \end{array}$$

which factorises as:

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

That is the two balanced cases differ only by an unobservable global phase (and likewise for the two constant cases).

Deutsch's algorithm (3)

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

The next step is to use the Hadamard gate to interfere the superposition on the first qubit, which yields:

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

The final step is to measure the first qubit, and we can see that the outcome will always be 0 if the function is constant, and 1 if balanced.

We can see that superposition and interference, in some sense, play complementary roles: we prepare a state in superposition, perform some operations, and then use interference to discern some global property of the state.

Quantum computing jargon

Query complexity:

- In Deutsch's algorithm we are not using a quantum computer to evaluate a "classically difficult" function *per se*, but rather using quantum phenomena to reduce the number of queries we need to make to an unknown function, to ascertain some information thereabout.

Oracles and black boxes:

- In Deutsch's algorithm, and other query complexity algorithms, we query U, which is known as a "black box", or often in quantum computing an "*oracle*". The oracle in Deutsch's algorithm is sufficiently simple that we can explicitly express each possible option, but frequently in quantum computing problems are framed in terms of oracles, even when this is not the case.

Deutsch-Jozsa algorithm

Academia Europaea

Richard Jozsa

Together with Richard Jozsa, who is now a Professor here in DAMTP, in 1992 David Deutsch generalised the algorithm to apply to constant / balanced functions of any input size.

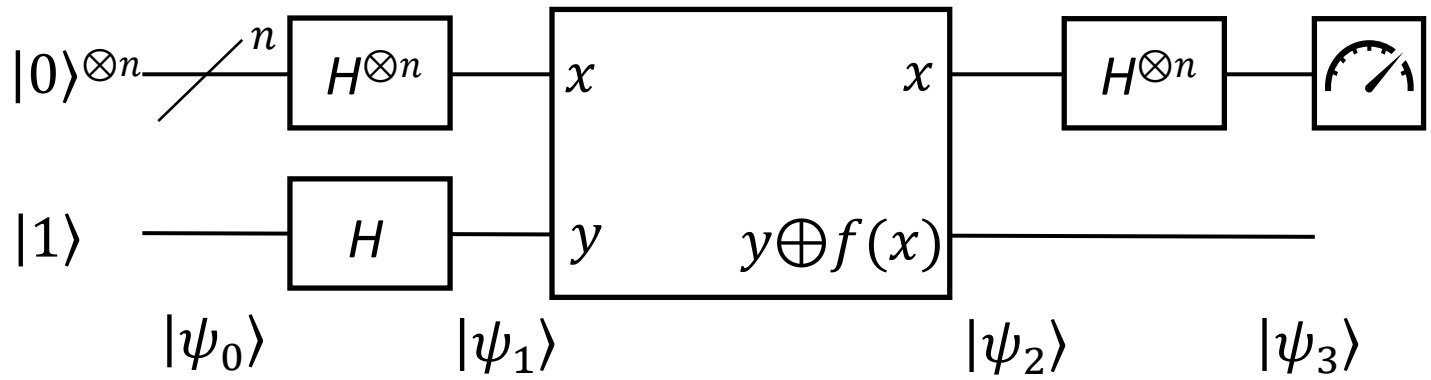
Deutsch's problem

The Deutsch-Jozsa algorithm is usually motivated in terms of Deutsch's problem:

- We have a function $f : \{0,1\}^n \to \{0,1\}$.
- We are promised this function is either constant (same output for each x) or balanced ($f(x)$ is equal to each of 0 and 1 for exactly half of the possible values of x).
- Classically, we can see that we may need to query the function $\frac{2^n}{2} + 1$ times to be sure whether the function is constant or balanced. That is, because there are 2^n possible bitstrings, x, so in the worst case even if we get the same outcome the first $\frac{2^n}{2}$ times we classically query the oracle, we cannot be sure whether the function is constant and balanced – only the $\left(\frac{2^n}{2} + 1\right)$th query will tell us this.
- But quantumly, if the function is encoded as a quantum oracle, then the Deutsch-Jozsa algorithm allows us to determine whether the function is constant or balanced with only a single oracle call.

Deutsch-Jozsa algorithm (1)

The circuit of the Deutsch-Jozsa algorithm closely resembles that of Deutsch's algorithm:



The initial state of which can be expressed:

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

which is then put into superposition, which can conveniently be expressed:

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

Deutsch-Jozsa algorithm (2)

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

The unitary transforms $|\psi_1\rangle$ to:

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

We now address the interference $H^{\otimes n}$ on the first n wires, for which we use the expression:

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{x\cdot z} |z\rangle$$

which allows us to express:

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

Deutsch-Jozsa algorithm (3)

We can now determine whether the function is constant or balanced by measuring the first n qubits of the final state (i.e., we neglect the final qubit which is in the $|-\rangle$ state):

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

Specifically, we consider the probability of measuring zero on every qubit, which corresponds to the term in the superposition where $|z\rangle$ is $|0\rangle^{\otimes n}$:

- In the case where the function is constant, then the co-efficient of $|0\rangle^{\otimes n}$, $\sum_x (-1)^{f(x)}/2^n$ is equal to ± 1... as this has amplitude 1, then we measure $|0\rangle^{\otimes n}$ with probability one.
- In the case where the function is balanced then $\sum_x (-1)^{f(x)}/2^n = 0$, and so we will never measure $|0\rangle^{\otimes n}$.

So it follows that measuring the first n qubits allows us to determine with certainty whether the function is constant (measure all zeros) or balanced (measure at least one 1).

Potential for exponential speed-up using a quantum computer

Imagine now that the oracle is held by a person, "Bob", who is spatially separated from the person, "Alice", who is trying to determine whether the function is constant or balanced.

- To resolve an instance of Deutsch's problem, classically Alice transmits $\frac{2^n}{2} + 1$ messages, each of size n bits, and each of which Bob replies to with a one bit message.
- Whereas quantumly the Deutsch-Jozsa algorithm requires only the transmission of a single $n + 1$ *qubit* message by Alice, to which Bob replies with a n qubit message.

So there is an exponential reduction in the amount of information transfer required to solve an instance of Deutsch's problem.

Where is this potential exponential advantage coming from?

Consider a function, f, which takes a n-bit binary number as an input. Note there are 2^n different n-bit binary numbers.

- Classically, we can only evaluate f for one of these binary numbers at a time.
- But quantumly, noting the direct correspondence between quantum states and binary numbers highlighted at the start of this lecture, we can evaluate the function for a superposition of all 2^n binary numbers in one go.

This fundamental property of quantum computing got everybody *very* excited but, as we shall see, finding useful quantum algorithms with an exponential advantage is actually rather more nuanced.

Summary

- We can encode any mathematical function as a unitary matrix.
- Deutsch's algorithm was the first algorithm that demonstrated a quantum advantage: specifically a reduction in query complexity compared to the classical case.
- The Deutsch-Jozsa algorithm generalises Deutsch's algorithm, and reveals the possibility of exponential speed-ups using quantum computers.