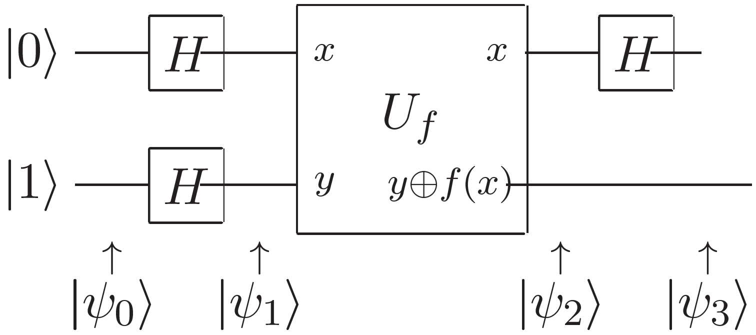

Figure III.22: Quantum circuit for Deutsch algorithm. [fig. from NC]

D Quantum algorithms

D.1 Deutsch-Jozsa

D.1.a Deutsch algorithm

¶1. This is a simplified version of Deutsch's original algorithm, which shows how it is possible to extract global information about a function by using quantum parallelism and interference (Fig. III.22).[5]

¶2. Suppose we have a function $f : \mathbf{2} \rightarrow \mathbf{2}$, as in Sec. C.5.
The goal is to determine whether $f(0) = f(1)$ with a *single* function evaluation. This is not a very interesting problem (since there are only four such functions), but it is a warmup for the Deutsch-Jozsa algorithm.

¶3. It could be expensive to decide on a classical computer. For example, suppose $f(0) =$ the millionth bit of π and $f(1) =$ the millionth bit of e. Then the problem is to decide if the millionth bits of π and e are the same.
It is mathematically simple, but computationally complex.

¶4. **Initial state:** Begin with the qubits $|\psi_0\rangle = |01\rangle$.

[5] This is the 1998 improvement by Cleve et al. to Deutsch's 1985 algorithm (Nielsen & Chuang, 2010, p. 59).

¶5. **Superposition:** Transform it to a pair of superpositions

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |+-\rangle. \tag{III.21}$$

by two tensored Hadamard gates.
Recall $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$ and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$.

¶6. **Function application:** Next apply U_f to $|\psi_1\rangle = |+-\rangle$.

¶7. Note $U_f|x\rangle|0\rangle = |x\rangle|0 \oplus f(x)\rangle = |x\rangle|f(x)\rangle$.

¶8. Also note $U_f|x\rangle|1\rangle = |x\rangle|1 \oplus f(x)\rangle = |x\rangle|\neg f(x)\rangle$.

¶9. Therefore, expand Eq. III.21 and apply U_f:

$$\begin{aligned}
|\psi_2\rangle &= U_f|\psi_1\rangle \\
&= U_f\left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right] \\
&= \frac{1}{2}[U_f|00\rangle - U_f|01\rangle + U_f|10\rangle - U_f|11\rangle] \\
&= \frac{1}{2}[|0, f(0)\rangle - |0, \neg f(0)\rangle + |1, f(1)\rangle - |1, \neg f(1)\rangle]
\end{aligned}$$

There are two cases: $f(0) = f(1)$ and $f(0) \neq f(1)$.

¶10. **Equal (constant function):** If $f(0) = f(1)$, then

$$\begin{aligned}
|\psi_2\rangle &= \frac{1}{2}[|0, f(0)\rangle - |0, \neg f(0)\rangle + |1, f(0)\rangle - |1, \neg f(0)\rangle] \\
&= \frac{1}{2}[|0\rangle(|f(0)\rangle - |\neg f(0)\rangle) + |1\rangle(|f(0)\rangle - |\neg f(0)\rangle)] \\
&= \frac{1}{2}(|0\rangle + |1\rangle)(|f(0)\rangle - |\neg f(0)\rangle) \\
&= \pm\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) \\
&= |+-\rangle.
\end{aligned}$$

The last line applies because global phase (including ±) doesn't matter.

¶11. **Unequal (balanced function):** If $f(0) \neq f(1)$, then

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{2}[|0,f(0)\rangle - |0,\neg f(0)\rangle + |1,\neg f(0)\rangle - |1,f(0)\rangle] \\ &= \frac{1}{2}[|0\rangle(|f(0)\rangle - |\neg f(0)\rangle) + |1\rangle(|\neg f(0)\rangle - |f(0)\rangle)] \\ &= \frac{1}{2}[|0\rangle(|f(0)\rangle - |\neg f(0)\rangle) - |1\rangle(|f(0)\rangle - |\neg f(0)\rangle)] \\ &= \frac{1}{2}(|0\rangle - |1\rangle)(|f(0)\rangle - |\neg f(0)\rangle) \\ &= \pm\frac{1}{2}(|0\rangle - |1\rangle)(|0\rangle - |1\rangle) \\ &= |--\rangle \end{aligned}$$

Clearly we can discriminate between the two cases by measuring the first qubit in the sign basis.

¶12. **Measurement:** Therefore we can determine whether $f(0) = f(1)$ or not by measuring the first bit of $|\psi_2\rangle$ in the sign basis, which we can do with the Hadamard gate (recall $H|+\rangle = |0\rangle$ and $H|-\rangle = |1\rangle$):

$$\begin{aligned} |\psi_3\rangle &= (H \otimes I)|\psi_2\rangle \\ &= \begin{cases} \pm|0\rangle|-\rangle, & \text{if } f(0) = f(1) \\ \pm|1\rangle|-\rangle, & \text{if } f(0) \neq f(1) \end{cases} \\ &= \pm|f(0) \oplus f(1)\rangle|-\rangle. \end{aligned}$$

¶13. Notice that the information is in the *data* register, not the result register. This technique is called *phase kick-back* (i.e., kicked back into the phase of the data register).

¶14. Therefore we can determine whether or not $f(0) = f(1)$ with a *single evaluation* of f.
(This is very strange!)

¶15. In effect, we are evaluating f on a superposition of $|0\rangle$ and $|1\rangle$ and determining how the results interfere with each other. As a result we get a definite (not probabilistic) determination of a global property with a single evaluation.

¶16. This is a clear example where a quantum computer can do something faster than a classical computer.

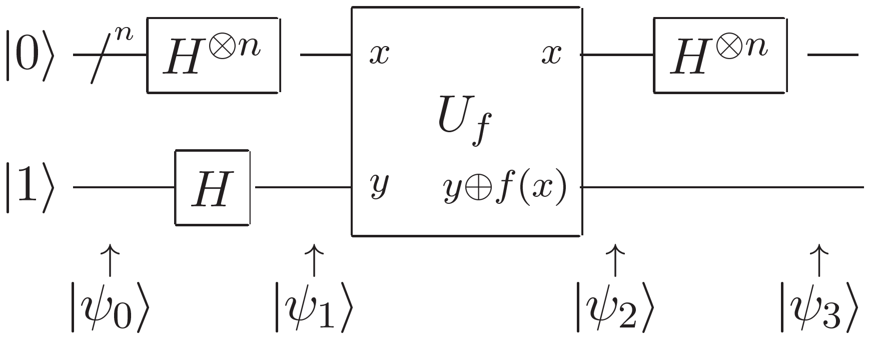

Figure III.23: Quantum circuit for Deutsch-Jozsa algorithm. [fig. from NC]

¶17. However, note that U_f has to uncompute f, which takes as much time as computing it, but we will see other cases (Deutsch-Jozsa) where the speedup is much more than 2×.

D.1.b Deutsch-Jozsa algorithm

¶1. The Deutsch-Jozsa algorithm is a generalization of the Deutsch algorithm to n bits; they published it in 1992; this is an improved version (Nielsen & Chuang, 2010, p. 59).

¶2. **The problem:** Suppose we are given an unknown function $f : \mathbf{2}^n \rightarrow \mathbf{2}$ in the form of a unitary transform $U_f \in \mathcal{L}(\mathcal{H}^{n+1}, \mathcal{H})$ (Fig. III.23).

¶3. We are told only that f is either constant or *balanced*, which means that it is 0 on half its domain and 1 on the other half. Our task is to determine into which class a given f falls.

¶4. **Classical:** Consider first the classical situation. We can try different input bit strings $\mathbf{x}$.
We might (if we're lucky) discover after the second query of f that it is not constant.
But we might require as many as $2^n/2+1$ queries to answer the question.
So we're facing $\mathcal{O}(2^{n-1})$ function evaluations.

¶5. **Initial state:** As in the Deutsch algorithm, prepare the initial state $|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle$.

¶6. **Superposition:** Use the Walsh-Hadamard transformation to create a superposition of all possible inputs:

$$|\psi_1\rangle = (H^{\otimes n} \otimes H)|\psi_0\rangle = \sum_{\mathbf{x}\in\mathbf{2}^n} \frac{1}{\sqrt{2^n}}|\mathbf{x}, -\rangle.$$

¶7. **Claim:** We will show that $U_f|\mathbf{x}, -\rangle = (-)^{f(\mathbf{x})}|\mathbf{x}\rangle|-\rangle$, where $(-)^n$ is an abbreviation for $(-1)^n$.

¶8. From the definition of $|-\rangle$ and U_f, $U_f|\mathbf{x}, -\rangle = |\mathbf{x}\rangle\frac{1}{\sqrt{2}}(|f(\mathbf{x})\rangle - |\neg f(\mathbf{x})\rangle)$.

¶9. Since $f(\mathbf{x}) \in \mathbf{2}$, $\frac{1}{\sqrt{2}}(|f(\mathbf{x})\rangle - |\neg f(\mathbf{x})\rangle) = |-\rangle$ if $f(\mathbf{x}) = 0$, and it $= -|-\rangle$ if $f(\mathbf{x}) = 1$.
This establishes the claim.

¶10. **Function application:** Since $U_f|\mathbf{x}, y\rangle = |\mathbf{x}, y \oplus f(x)\rangle$, you can see that:

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

¶11. The top n lines contain a superposition of the 2^n simultaneous evaluations of f. To see how we can make use of this information, let's consider their state in more detail.

¶12. For a *single* bit you can show (exercise!):

$$H|x\rangle = \sum_{z\in\mathbf{2}} \frac{1}{\sqrt{2}}(-)^{xz}|z\rangle.$$

(This is just another way of writing $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.)

¶13. Therefore, for the n bits:

$$\begin{aligned} H^{\otimes n}|x_1, x_2, \ldots, x_n\rangle &= \frac{1}{\sqrt{2^n}} \sum_{z_1,\ldots,z_n\in\mathbf{2}} (-)^{x_1z_1+\cdots+x_nz_n}|z_1, z_2, \ldots, z_n\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{\mathbf{z}\in\mathbf{2}^n} (-)^{\mathbf{x}\cdot\mathbf{z}}|\mathbf{z}\rangle, \qquad \text{(III.22)} \end{aligned}$$

where $\mathbf{x} \cdot \mathbf{z}$ is the bitwise inner product. (It doesn't matter if you do addition or $\oplus$ since only the parity of the result is significant.)
Remember this formula!

¶14. Combining this and the result in ¶10,

$$|\psi_3\rangle = (H^{\otimes n} \otimes I)|\psi_2\rangle = \sum_{\mathbf{z}\in\mathbf{2}^n} \sum_{\mathbf{x}\in\mathbf{2}^n} \frac{1}{2^n}(-)^{\mathbf{x}\cdot\mathbf{z}+f(\mathbf{x})}|\mathbf{z}\rangle|-\rangle.$$

¶15. **Measurement:** Consider the first n qubits and the amplitude of one particular basis state, $\mathbf{z} = |0\rangle^{\otimes n}$.
Its amplitude is $\sum_{\mathbf{x}\in\mathbf{2}^n} \frac{1}{2^n}(-)^{f(\mathbf{x})}$.

¶16. **Constant function:** If the function is constant, then all the exponents of -1 will be the same (either all 0 or all 1), and so the amplitude will be ± 1.
Therefore all the other amplitudes are 0 and any measurement must yield 0 for all the bits (since only $|0\rangle^{\otimes n}$ has nonzero amplitude).

¶17. **Balanced function:** If the function is not constant then (*ex hypothesi*) it is balanced.
But more specifically, if it is balanced, then there must be an equal number of $+1$ and -1 contributions to the amplitude of $|0\rangle^{\otimes n}$, so its amplitude is 0.
Therefore, when we measure the state, at least one qubit must be nonzero (since the all-0s state has amplitude 0).

¶18. **Good and bad news:** The *good news* is that with one quantum function evaluation we have got a result that would require between 2 and $\mathcal{O}(2^{n-1})$ classical function evaluations (exponential speedup).
The *bad news* is that the algorithm has no known applications!

¶19. Even if it were useful, the problem could be solved probabilistically on a classical computer with only a few evaluations of f.

¶20. However, it illustrates principles of quantum computing that can be used in more useful algorithms.