Lecture 13

Comment: HW1 Q4: good to redo in Dirac notation if you used matrices because when there are more than two qubits (say, on exam), direct matrix manipulation is really inefficient. HW1 Q2 interpretation: *non-adaptive* deterministic query complexity of NAND tree on n variables is n. If adaptive, answer is still $\Omega(n)$, but need an enhanced argument. Consider $(x_1 \wedge x_2) \vee (\neg x_1 \wedge x_3)$, which needs 3 non-adaptive queries but only 2 adaptive ones.

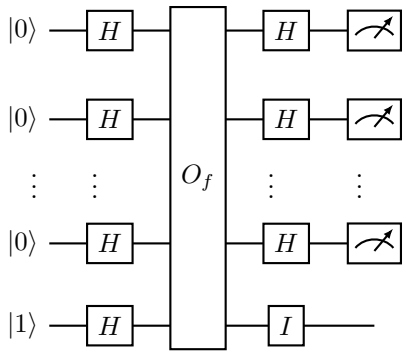
Recall the Deutsch-Jozsa problem: given $f\colon \{0,1\}^n \to \{0,1\}$, ask balanced or constant. Classically, need $2^{n-1}+1$ queries f to solve with certainty. Quantumly, it turns out that only 1 (quantum) query to f suffices.

Recall the quantum oracle for f is given by

$$O_f |x\rangle |b\rangle \to |x\rangle |b \oplus f(x)\rangle , \tag{74}$$

where $x \in \{0,1\}^n, b \in \{0,1\}$.

The quantum circuit for the Deutsch-Jozsa algorithm is shown below:



Output "constant" if the measurement outcome is 0^n and "balanced" if the measurement outcome is not 0^n.

To make the analysis easier, we will change the I to an H (which does not affect the measurement distribution as per a lemma from 2 lectures ago), introduce the quantum phase oracle and give a general expression for how multiple Hadmard gates act on a computational basis state.

Definition 13. The quantum phase oracle for f is given by

$$U_f |x\rangle |b\rangle = (-1)^{b \cdot f(x)} |x\rangle |b\rangle , \tag{75}$$

where $x \in \{0,1\}^n, b \in \{0,1\}$.

The following lemma will be used in the next two propositions.

Lemma 1. *For $b \in \{0,1\}$, we have*

$$H |b\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b |1\rangle). \tag{76}$$

Proof. By direct calculation. □

Proposition 5 (Phase kickback trick)**.** *We have $U_f = (\mathbb{1}_{2^n} \otimes H) O_f (\mathbb{1}_{2^n} \otimes H)$.*

Proof. Write $\mathbb{1}$ for $\mathbb{1}_{2^n}$. Then

$$\begin{aligned}
&|x\rangle |b\rangle \overset{\mathbb{1}\otimes H}{\mapsto} |i\rangle \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b |1\rangle) && \text{by Eq. (76)} \\
\overset{O_f}{\mapsto} &\frac{1}{\sqrt{2}} |x\rangle (|f(x)\rangle + (-1)^b |f(x) \oplus 1\rangle) \\
\overset{\mathbb{1}\otimes H}{\mapsto} &\frac{1}{\sqrt{2}} |x\rangle (\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{f(x)} |1\rangle) + (-1)^b \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{f(x)\oplus 1} |1\rangle)) \\
= &\frac{1}{2} |x\rangle ((1 + (-1)^b) |0\rangle + (-1)^{f(x)}(1 - (-1)^b) |1\rangle) \\
= &(-1)^{f(x)\cdot b} |x\rangle |b\rangle ,
\end{aligned}$$

as required. □

Proposition 6. *For all $x \in \{0,1\}^n$,*

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle , \tag{77}$$

where $H^{\otimes n} \coloneqq H \otimes \cdots \otimes H$ (n times) and $x \cdot y \coloneqq \sum_{i=1}^n x_i y_i$.

Proof. We have

$$\begin{aligned} H^{\otimes n} |x\rangle =& H |x_1\rangle \otimes \cdots \otimes H |x_n\rangle \\ =& \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1} |1\rangle) \otimes \cdots \otimes \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_n} |1\rangle) \qquad \text{by Eq. (76)} \\ =& \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle , \end{aligned}$$

as required □

Analysis of the Deutsch-Jozsa algorithm.

$$\begin{aligned} |0^n\rangle |1\rangle &\overset{H^{\otimes n}}{\mapsto} \frac{1}{\sqrt{2^n}} \sum_x |x\rangle |1\rangle \\ &\overset{U_f}{\mapsto} \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} |x\rangle |1\rangle \\ &\overset{H^{\otimes n}}{\mapsto} \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} \sum_y (-1)^{x \cdot y} |y\rangle |1\rangle \\ &= \frac{1}{2^n} \sum_y \Big(\sum_x (-1)^{f(x) + x \cdot y} \Big) |y\rangle |1\rangle \end{aligned}$$

Therefore the probability of measuring $y = 0^n$ is

$$\Pr[0^n] = \frac{1}{2^{2n}} \Big(\sum_x (-1)^{f(x)} \Big)^2 . \tag{78}$$

If f is constant, then $\Pr[0^n] = 1$. If f is balanced, then $\Pr[0^n] = 0$.

Remark 2. Does DJ mean we have a real (unconditional) provable exponential quantum speedup? **No.** To compute f in the real world, a description of $f \colon \{0,1\}^n \to \{0,1\}$ as a circuit[4] must be given. But once we have such a description of f, we may be able to solve the problem faster than time $2^{n-1} + 1$ by analyzing the circuit diagram rather than only evaluating (aka querying) the circuit. Comment: will say a bit more about this next lecture.

If we are *forced* to only query the circuit, then yes, DJ gives an exponential quantum speedup between quantum and randomized query complexity (if we also demand certain correctness). Being forced to *only* query the circuit is the essential assumption of the query model of computation. The real-world model of computation (aka Turing model) does not make this assumption, so DJ does *not* constitute a provable exponential quantum speedup in the real world.

[4] If a circuit sounds abstract, think of this as a program or description. These become circuits once you compile them to run on hardware.