COS 597A: Quantum Cryptography — Princeton University
Lecture 4 (September 24, 2018)
Lecturer: Mark Zhandry — Scribe: Benjamin Kuykendall

Notes for Lecture 4

Announcements

[HW1] is due October 10. Please type up your solutions (LaTeX preferred). Either email your solutions to Prof. Zhandry (preferred) or print them out and hand them in during class by the due date.

1 Summary of Quantum Circuits

Recall: states are elements of $\mathbb{C}^{2^n}$ and we operate on them by unitary matrices. In the circuit model these unitaries are composed of smaller unitary "gates"; we work with a fixed set of gates that operate on one or two qubits. (However, we may use other small unitaries in our descriptions of algorithms. This is fine, since they can be implimented using a small number of gates).

Consider the gate set given below. The gates are presented in two ways: in linear algebra notation and by their actions on the elements of the computational basis. Either suffices to give the transformation, and the later will be useful in analysis.

Hadamard $\quad H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \quad H|b\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^b|1\rangle\right)$

Phase gate $\quad R = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix} \quad R|b\rangle = e^{i(\pi/4)b}|b\rangle$

Controlled NOT $\quad \text{CNOT} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix} \quad \text{CNOT}|b,c\rangle = |b, b\oplus c\rangle$

In this model we proved that any function $f : \{0,1\}^n \to \{0,1\}$ yields a unitary

$$U_f|x,b\rangle = |x, b\oplus f(x)\rangle\,.$$

Further, if f can we written as a circuit of S NAND gates, then U_f can be written as a circuit of kS Toffoli gates. This result shows that any classical algorithm can be implemented as a quantum algorithm. Today, we go further, showing quantum algorithms that are stronger than their classical counterparts.

2 Deutsch-Jozsa problem

All of the problems we study today are *oracle* problems; that is, given black-box access to a function f, we hope to learn something about f. Black-box access can model a number of different scenarios. In summary, whenever you have access to arbitrary outputs of f but not a description of it, you have black-box access.

In classical algorithms, it is clear how to define this formally: you specify a string x and then receive the value $f(x)$ from the oracle. In quantum computing, we want an oracle model where you can submit a quantum state $|x\rangle$. By analogy to the work we did last week, define a quantum oracle as follows.

Classical oracle:

$$x \to \boxed{f} \to f(x)$$

Quantum oracle:

$$\sum_{b,c} \alpha_{b,c} |b, c\rangle \to \boxed{U_f} \to \sum_{b,c} \alpha_{b,c} |b, c \oplus f(b)\rangle$$

2.1 1-bit Problem

Problem 1 (1-bit Deutsch-Jozsa)**.** *Given oracle access to* $f : \{0,1\} \to \{0,1\}$ *determine if* f *is constant, i.e. output* $f(0) \oplus f(1)$*.*

Classical: need exactly two queries. Given only one query, we learn $f(b)$, but without another query, we know nothing about $f(0) \oplus f(1)$.

Quantum: one quantum query suffices. The basic idea is to use the linearity of the quantum oracle to get both $f(0)$ and $f(1)$ at once.

First attempt

1. Prepare $|\phi_1\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle + |1\rangle\Big)\,|0\rangle$.
2. Submit $|\phi_1\rangle$ to the oracle and receive $|\phi_2\rangle$.
3. Measure.

Note: in order to prepare $|\phi_1\rangle$ just compute $H\,|1\rangle \otimes |0\rangle$.

It is clear that $|\phi_2\rangle$ contains both answers in superposition:

$$\begin{aligned}|\phi_2\rangle &= U_f\,|\phi_1\rangle \\ &= \frac{1}{\sqrt{2}}\Big(U_f\,|0,0\rangle + U_f\,|1,0\rangle\Big) \\ &= \frac{1}{\sqrt{2}}\Big(|0,0\oplus f(0)\rangle + |1,0\oplus f(1)\rangle\Big) \\ &= \frac{1}{\sqrt{2}}\Big(|0,f(0)\rangle + |1,\oplus f(1)\rangle\Big).\end{aligned}$$

However, once we measure we lose one of the answers:

With probability 1/2: observe $0, f(0)$ and state collapses to $|0, f(0)\rangle$.

With probability 1/2: observe $1, f(1)$ and state collapses to $|1, f(1)\rangle$.

As written, this is no better than the classical algorithm. This algorithm can be improved by transforming $|\phi_2\rangle$ further, but it will not solve the problem with probability one. A little bit more ingenuity is required.

Deutsch-Jozsa algorithm (1-bit case)

1. Prepare $|\phi_1\rangle = \frac{1}{2}\Big(|0\rangle + |1\rangle\Big)\Big(|0\rangle - |1\rangle\Big)$.
2. Submit $|\phi_1\rangle$ to the oracle and receive $|\phi_2\rangle$.
3. Apply H to the first qubit to get $|\phi_3\rangle$
4. Partially measure the first qubit.

Note: in order to prepare $|\phi_1\rangle$ just compute $H\,|0\rangle \otimes H\,|1\rangle$.

Proof. Again, $|\phi_2\rangle$ will contain information about both $f(0)$ and $f(1)$; however, it will be in a different format this time.

$$\begin{aligned}|\phi_2\rangle &= U_f\,|\phi_1\rangle \\ &= \frac{1}{2}\Big(U_f\,|00\rangle - U_f\,|01\rangle + U_f\,|10\rangle - U_f\,|11\rangle\Big) \\ &= \frac{1}{2}\Big(|0,0\oplus f(0)\rangle - |0,1\oplus f(0)\rangle + |1,0\oplus f(1)\rangle - |1,1\oplus f(1)\rangle\Big) \\ &= \frac{1}{2}\Big(|0\rangle\left(|f(0)\rangle - |1\oplus f(0)\rangle\right) + |1\rangle\left(|f(1)\rangle - |1\oplus f(1)\rangle\right)\Big).\end{aligned}$$

Noting that $|b\rangle - |1 \oplus b\rangle = (-1)^b\big(|0\rangle - |1\rangle\big)$

$$\begin{aligned}|\phi_2\rangle &= \frac{1}{2}\Big((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\Big) \otimes \Big(|0\rangle - |1\rangle\Big) \\ &= \Big(\frac{(-1)^{f(0)}}{2}\Big)\Big(|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle\Big) \otimes \Big(|0\rangle - |1\rangle\Big).\end{aligned}$$

Recalling $H|b\rangle = |0\rangle + (-1)^b|1\rangle$ we can write this as

$$|\phi_2\rangle = \Big(\frac{(-1)^{f(0)}}{2}\Big)\Big(H|f(0)\oplus f(1)\rangle\Big) \otimes \Big(|0\rangle - |1\rangle\Big).$$

Now recalling $HH = I$ we can easily compute the next state. Ignoring the second qubit and the phase write

$$|\phi_3\rangle = H\Big(H|f(0)\oplus f(1)\rangle\Big) = |f(0)\oplus f(1)\rangle.$$

Upon measuring the first qubit we get output $f(0) \oplus f(1)$ with probability one. □

This algorithm has a certain weirdness to it: U_f is supposed to put information from f into the second qubit of the state. However, we only measured the first qubit. This effect is intrinsic to entanglement: "writing" to one qubit can effect the whole state.

This particular trick is known as *phase kickback* and can always be used to transfer the result of an oracle query to the phase. To summarize:

$$U_f\Big(|b\rangle \otimes H|1\rangle\Big) = (-1)^{f(b)}|b\rangle \otimes H|1\rangle.$$

2.2 n-bit Problem

Problem 2 (Deutsch-Jozsa)**.** *Given oracle access to $f : \{0,1\}^n \to \{0,1\}$ decide if f is constant or balanced (meaning $\#f^{-1}(0) = \#f^{-1}(1)$).*

Note this is a natural extension of the 1-bit problem since all non-constant 1-bit functions are balanced.

Classical deterministic: $\Theta(2^n)$ queries needed: even if the first $2^n/2$ queries give the same result, f could still be either constant or balanced.

Classical randomized: a constant number of random queries gives a constant success probability (e.g. 3 queries gives success probability 7/8).

Quantum: one quantum query suffices to succeed with probability one. The basic idea is the same.

Notation: use $\bigotimes$ to denote repeat tensor products, i.e.

$$\bigotimes_{i\in[n]} |a_i\rangle := |a_1\rangle \otimes |a_2\rangle \cdots \otimes |a_n\rangle \,.$$

Use $(\cdot)^{\otimes n}$ for a product of a state with itself n times, i.e.

$$|\phi\rangle^{\otimes n} := \bigotimes_{i\in[n]} |\phi\rangle \,.$$

Deutsch-Jozsa algorithm (full)

1. Prepare $|\phi_1\rangle = \Big(H\,|0\rangle\Big)^{\otimes n} \otimes H\,|1\rangle$.
2. Submit $|\phi_1\rangle$ to the oracle and receive $|\phi_2\rangle$.
3. Apply H to each of the first n qubits independently to get $|\phi_3\rangle$.
4. Partially measure the first n qubits; output `constant` when the result is 0^n and `balanced` otherwise.

Proof. Note that our initial state can be re-written as a sum over all bitstrings:

$$\begin{aligned}
|\phi_1\rangle &= \Big(H\,|0\rangle\Big)^{\otimes n} \otimes H\,|1\rangle \\
&= \Big(\frac{1}{\sqrt{2}}\Big(|0\rangle + |1\rangle\Big)\Big)^{\otimes n} \otimes H\,|1\rangle \\
&= 2^{-n/2} \sum_{x\in\{0,1\}^n} |x\rangle \otimes H\,|1\rangle \,.
\end{aligned}$$

Putting this characterization together with our knowledge of phase kickback, it is easy to compute the next state.

$$\begin{aligned}
|\phi_2\rangle &= U_f 2^{-n/2} \sum_x |x\rangle \otimes H\,|1\rangle \\
&= 2^{-n/2} \sum_x (-1)^{f(x)}\,|x\rangle \otimes H\,|1\rangle \\
&= \left(2^{-n/2} \sum_x (-1)^{f(x)}\,|x\rangle\right) \otimes H\,|1\rangle
\end{aligned}$$

Ignore the last qubit. Now we apply H to each qubit

$$\begin{aligned}
|\phi_3\rangle &= H^{\otimes n}\left(2^{-n/2}\sum_x (-1)^{f(x)}|x\rangle\right) \\
&= 2^{-n}\sum_x (-1)^{f(x)}\bigotimes_{i\in[n]} H|x_i\rangle \\
&= 2^{-n}\sum_x (-1)^{f(x)}\bigotimes_{i\in[n]}\Big(|0\rangle + (-1)^{x_i}|1\rangle\Big).
\end{aligned}$$

Note that each bitstring occurs once in the product. The phase is determined by the inner product mod 2 as follows.

$$\begin{aligned}
|\phi_3\rangle &= 2^{-n}\sum_x (-1)^{f(x)}\sum_{y\in\{0,1\}^n}(-1)^{\langle x,y\rangle}|y\rangle \\
&= 2^{-n}\sum_y\sum_x (-1)^{f(x)+\langle x,y\rangle}|y\rangle\,.
\end{aligned}$$

Finally, we measure. Since we only care about the case where the measurement yields $y = 0^n$, compute that coefficient

$$\begin{aligned}
\alpha_{0^n} &= 2^{-n}\sum_x (-1)^{f(x)+\langle x,0^n\rangle} \\
&= 2^{-n}\sum_x (-1)^{f(x)} \\
&= 2^{-n}\left(\sum_{x:f(x)=0} 1 + \sum_{x:f(x)=1}(-1)\right) \\
&= 2^{-n}\Big(\big(\#f^{-1}(0)\big) - \big(\#f^{-1}(1)\big)\Big).
\end{aligned}$$

When f is balanced $\alpha_{0^n} = 0$ so $y = 0^n$ will be observed with probability zero.
When f is constant $\alpha_{0^n} = \pm 1$ so $y = 0^n$ will be observed with probability one. □

One way to interpret this algorithm is to note that $H^{\otimes n}$ is a *Discrete Fourier Transform* on $\{0,1\}^n$. In this way, the state after making the query can be considered as a signal in high-dimensional space, and the Fourier transform moves this information into the computational basis. Explicitly,

$$H^{\otimes n}|x\rangle = 2^{-n/2}\sum_y (-1)^{\langle x,y\rangle}|y\rangle\,.$$

3 Simon's Problem

This problem has a similar flavor to the Deutsch-Jozsa problem, but the advantage is even larger; as we will see, a quantum computer has an exponential advantage over even a randomized classical one in this problem.

Problem 3 (Simon's problem)**.** *Given oracle access to* $f : \{0,1\}^n \to \{0,1\}^n$ *such that for some* $s \in \{0,1\}^n$

1. $\forall x \in \{0,1\}^n$, $f(x \oplus s) = f(x)$ *and*
2. $f(x) = f(y)$ *implies* $y \in \{x, x \oplus s\}$,

find the value of s*.*

Classical randomized: $\Theta(2^{n/2})$. The algorithm chooses $\Theta(2^{n/2})$ values of x randomly and evaluates f. If it finds $f(x) = f(y)$ with $x \neq y$ it knows $s = x \oplus y$. This collision occurs with good probability by "birthday paradox" analysis.

Quantum: $O(n)$ queries.

Simon's algorithm:

1. Prepare $|\phi_1\rangle = 2^{-n/2} \sum_x |x\rangle \otimes |0^n\rangle$.
2. Submit $|\phi_1\rangle$ to the oracle and receive $|\phi_2\rangle$.
3. Partially measure the last n qubits giving y and $|\phi_3\rangle$.
4. Apply H to each of the first n qubits independently to get $|\phi_4\rangle$.
5. Measure again to obtain z.
6. Repeat obtaining $z_1, \ldots, z_m$ until the span of the z_i has dimension $n-1$; output the remaining vector outside of the span.

Proof. The oracle is now queried *without* using phase kickback; the result is simply

$$|\phi_2\rangle = 2^{-n/2} \sum_x |x, f(x)\rangle .$$

Measurement results in a random value of $y = f(x)$ giving, up to normalization

$$|\phi_3\rangle = \sum_{x : f(x) = y} |x\rangle |y\rangle .$$

However by the promise on f, this is only two values

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\Big(|x\rangle + |x \oplus s\rangle\Big)\,|y\rangle\,.$$

Dropping the last n qubits and applying the H gates (i.e. taking the DFT) yields

$$\begin{aligned}
|\phi_3\rangle &= \frac{1}{\sqrt{2}}\left(2^{-n/2}\sum_z (-1)^{\langle x,z\rangle}\,|z\rangle + 2^{-n/2}\sum_z (-1)^{\langle x\oplus s,z\rangle}\,|z\rangle\right) \\
&= 2^{-n/2-1}\sum_z \Big((-1)^{\langle x,z\rangle} + (-1)^{\langle x\oplus s,z\rangle}\Big)\,|z\rangle \\
&= 2^{-n/2-1}\sum_z (-1)^{\langle x,z\rangle}\Big(1 + (-1)^{\langle s,z\rangle}\Big)\,|z\rangle\,.
\end{aligned}$$

Since $\langle s,z\rangle = 0 \iff s \perp z$ we have

$$\begin{aligned}
|\phi_3\rangle &= 2^{-n/2-1}\sum_{z\perp s} (-1)^{\langle x,z\rangle}(1+1)\,|z\rangle + 2^{-n/2-1}\sum_{z\not\perp s} (-1)^{\langle x,z\rangle}(1-1)\,|z\rangle \\
&= 2^{-n/2+1}\sum_{z\perp s} (-1)^{\langle x,z\rangle}\,|z\rangle\,.
\end{aligned}$$

Thus the final measurement always yields a random z such that $z \perp s$, allowing us to use linear algebra to find s after getting sufficiently many $(n-1)$ linearly-independent values of z in this manner. □