Lecture 6: Basic applications of quantum computing

Rajat Mittal

IIT Kanpur

Armed with the knowledge of postulates of quantum mechanics and circuits, we can already understand some applications of quantum computing. In this lecture, we will talk about two of them: first one is called *quantum teleportation* and other is an algorithm to show quantum advantage (known as *Deutsch-Jozsa algorithm*, a generalization of Deutsch's algorithm).

1 Quantum Teleportation

I would like to thank the author of the website http://algassert.com. They have a nice explanation of swapping and teleportation. This section is motivated by this website.

Let us look at another application of the fourth postulate, specifically entanglement, called *quantum teleportation*. You might have noticed that a qubit is much harder to describe than a classical bit, i.e., a qubit takes two complex numbers to describe but a bit just needs a binary value.

Suppose two parties, Alice and Bob, want to exchange a qubit. Unfortunately, they only have access to a classical communication channel and not a quantum channel. It seems that it should not be possible to exchange a qubit as that requires infinite bits of information to describe.

In other words, to transfer a quantum bit, we should have some form of quantum communication. Surprisingly, using entanglement and classical communication, we can transfer a qubit without using any form of quantum communication!! That means, Alice and Bob need access to a classical channel and share a pair of entangled states. This pair does not (should not too) depend upon the qubit to be transferred. This protocol is called *quantum teleportation*.

Looking at it from a different perspective, suppose Alice and Bob have quantum computers but *don't* have a channel which can transfer quantum bits. Using entanglement, we can transfer quantum bits from one party to another with the help of only classical communication. This protocol is called the quantum teleportation protocol.

Swapping two quantum bits: Before we try our hand on quantum teleportation, let us look at an easier problem. We are given two qubits and we want to swap their states (obviously, we are not allowed to physically change their places).

Exercise 1. How will you do it classically?

The easiest way, classically, is to have a temporary variable and then swap the states. The problem is, we are not allowed to copy states in quantum computation. There is a trick to achieve the swap classically, without using copy. Given that x and y are two bits, notice the following set of operations.

$$\begin{aligned} x &= x \oplus y \\ y &= y \oplus x \\ x &= x \oplus y \end{aligned}$$

Convince yourself that the following operation swaps the two bits. Notice that the procedure above requires x (and y) to be bits, so that $\oplus$ operation is defined. We want to achieve the same thing quantumly.

Exercise 2. What is the quantum equivalent of $\oplus$ gate?

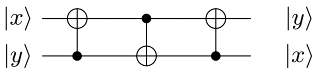

Fig. 1. Circuit for swapping two quantum bits

The quantum equivalent of classical XOR gate is the quantum CNOT gate. So, our circuit will instead consist of three CNOT's.

Again, check that the circuit in Fig. 1 works for all classical values of x and y. The linearity of quantum mechanics ensures that this circuit works for all states.

Exercise 3. Take two arbitrary quantum states and show that Fig. 1 performs swap through direct calculation.

Protocol for teleporting a quantum bit: For teleportation, we need to transfer a qubit from Alice to Bob. The problem is, one qubit is with Alice and one with Bob. So, it is not possible to apply CNOT on these two qubits.

Exercise 4. Show that the first CNOT is redundant if we start with $|0\rangle$ state on Bob's part.

So, we want to get the following circuit.

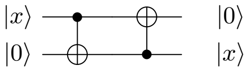

Fig. 2. Simplified circuit when Bob's qubit is $|0\rangle$

The idea is to use an intermediate qubit and use that to transfer the state of Alice to state of Bob. This will only require CNOT's between Alice's qubit and intermediate qubit, or between intermediate qubit and Bob's qubit (this trick is classical).

Exercise 5. Find the classical circuit which takes $x, y, z \rightarrow x, y, z \oplus x$ using XOR operations where no gate can be used between x and z.

Hint: Use 4 XOR gates

The question remains, who will have the intermediate qubit. It is kept with Alice, and CNOTs between intermediate and Bob's qubit is simulated using entanglement and classical communication. The circuit becomes

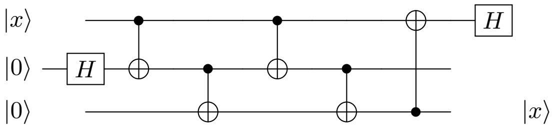

Fig. 3. Introducing the intermediate qubit

Notice that a Hadamard gate at the beginning of second qubit (the initial state of second qubit is not important) and at the end of the third qubit does not change anything. The last two operations can be changed by switching from CNOT to CZ gate. We switch to CZ because control and target can be switched for CZ without affecting anything. Please convince yourself that the circuit below is equivalent.

Once again the first CNOT can be switched (using Hadamard) and then CZ can be removed (control is $|0\rangle$).

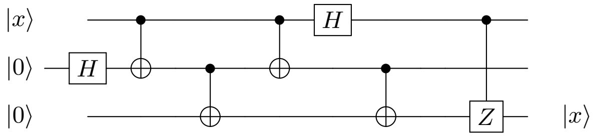

Fig. 4. Switching between CNOT and CZ

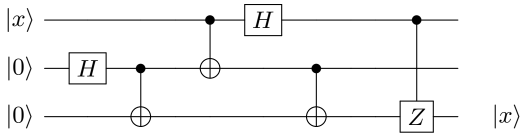

Fig. 5. Switching between CNOT and CZ again

Note the last two gates have only control qubits on Alice's part, they can be simulated using measurement and then sending the measurement results to Bob (by Alice). Also, The first Hadamard gate and the first gate between Alice's intermediate qubit and Bob's qubit is equivalent to having entangled qubits (it does not depend upon the state to be transferred). The full protocol is given below.

As mentioned before, this protocol requires the use of entanglement. Alice and Bob can meet before and keep one part (qubit) of the Bell state with each of them. Suppose, Alice wants to transfer state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ to Bob. Notice that we have chosen a completely arbitrary state to transfer to Bob.

Suppose the state Alice wants to transfer, $|\psi\rangle$, is the first qubit and her part of Bell state is the second qubit. Alice applies CNOT gate to these two qubits. Remember, CNOT gate is a 2-qubit gate, which applies NOT gate to the second qubit if and only if the first qubit is in state $|1\rangle$.

Exercise 6. Show that CNOT is unitary operator.

Then she applies Hadamard gate to her first qubit.

Exercise 7. What is the state of three qubits now?

It can be shown that the resulting state is,

$$\frac{1}{2}\left(|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)\right).$$

Now Alice measures her two qubits and sends them to Bob.

Exercise 8. Convince yourself that Bob can recover $|\psi\rangle$ using Pauli operators.

This completes the quantum teleportation. Alice is able to transfer one quantum bit using two classical bits of communication.

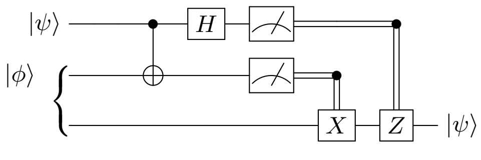

Fig. 6. Final protocol (here $|\phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is the Bell state).

Exercise 9. We said that we can transfer and not copy the quantum bit. Why?

In quantum computing we can't copy qubits, this is known as *no-cloning* theorem.

2 Query model

Before we begin to take a look at quantum algorithms, let us take a look at the *query model.* Most of the algorithms in the quantum world are query algorithms, or are described as query algorithms. The two best known algorithms, Shor's (the critical subroutine for order-finding) and Grover's, can both be put in this format.

The query format is different from the standard format because the input is not directly given to the algorithm. Let us assume that we want to compute a function $\{0,1\}^n \to \{0,1\}$. The input x is an n bit string $\{0,1\}^n$, then the algorithm has access to a black-box which on an input $i \in [n]$ outputs the bit x_i. This is described by an oracle O_x,

$$O_x|i,b\rangle \to |i, b \oplus x_i\rangle.$$

Exercise 10. Prove that the oracle O_x is unitary.

Note 1. This is in the same spirit as the transformation $(x,y) \to (x, y \oplus f(x))$.

Specifically, to obtain the i^{th} bit x_i, we input $|i,0\rangle$ into the oracle and get the output on the second part of the register. The first part of the register is the *address* and the second part is called the *target.* Since our alphabet is $\{0,1\}$, the target will be one qubit.

The dimension of the address part should be enough to specify the index i. Most of the time we will choose $n = 2^k$, so the index i will be a k bit string and hence the address part will be k qubits.

Note 2. Since there is a difference between query and standard model, the exponential separation between quantum and classical model do not transfer to exponential separation in the standard model.

Sometimes a different oracle is used for querying the input instead of the one mentioned above. It is called the *phase oracle* (as it puts the phase depending upon the input bit x_i),

$$O'_x|i,b\rangle = (-1)^{bx_i}|i,b\rangle.$$

You will show that the query oracle and phase oracle are equivalent and only differ by an application of Hadamard to the target part of the register.

Exercise 11. Show that,

$$O_x|i,-\rangle = (-1)^{x_i}|i,-\rangle.$$

Exercise 12. Why is this global phase not useless?

Note 3. Since these are quantum oracles, we assume that we can query in superposition. In other words, the state of the address part (as well as the target part) can be in superposition.

You can also check that,

$$O_x|i,+\rangle = |i,+\rangle.$$

Exercise 13. Using the facts above, show that the oracle O_x and O'_x are *equivalent*, in the sense that one can be converted into another.

There is another possible definition of phase query oracle, instead of $i \in [n]$, consider $i \in \{0,1,2,\cdots,n\}$. Then,

$$O''_x|i\rangle = (-1)^{x_i}|i\rangle,$$

Where we assume $O''_x|0\rangle = |0\rangle$.

Exercise 14. Show that if we only take $i \in [n]$ for the above oracle (don't have 0 as the address state), then we can't distinguish between x and its complement.

We will use either of the three mentioned oracles depending upon the application, remember that all three of them are equivalent.

As mentioned before, most of the time, we will choose $n = 2^k$. Many a times, the input $x \in \{0,1\}^n$ is instead interpreted as a function f from $\{0,1\}^k$ to $\{0,1\}$. This means, every index $i \in [n]$ is interpreted as a k bit binary string with x_i being considered as the function value $f(i)$.

This interpretation shows another reason why query model is important. Suppose there is a subroutine for f and that subroutine is very expensive. Then, query model translates to the task of computing another function through this subroutine, where we want to minimize the applications of this subroutine.

Exercise 15. Convince yourself that this is just the relabelling of the input.

3 Deutsch-Jozsa algorithm

We are ready for our first algorithm after learning the circuit model. This is going to be an extension of Deutsch's algorithm, the small 1 bit problem we saw before.

Exercise 16. If you don't remember, look at the notes for Deutsch's problem.

The Deutsch-Jozsa problem is a simple generalization of Deutsch's problem, given a string $x \in \{0,1\}^n$ (where $n = 2^k$), find whether

- all x_i's are same,
- or string x is balanced, i.e., exactly half the x_i's are 1 and other half of them are 0.

You are promised that the input falls in one of the cases mentioned above. From the discussion about viewing the input as being the output of a function f, this problem is equivalent to determining whether the corresponding function to the input is balanced or constant.

Exercise 17. Show that the task of determining $f(0) = f(1)$ is same as the above problem when $n = 2$.

We are given the query oracle,

$$O_x|i, b\rangle = (-1)^{bx_i}|i, b\rangle,$$

where i is a k length binary string. So, the state space of the oracle is $k + 1$ qubits.

We can rephrase the Deutsch-Jozsa problem in terms of phases,

- All x_i are same, i.e., $|\sum_i (-1)^{x_i}| = 2^k$.
- String x is balanced, i.e., $|\sum_i (-1)^{x_i}| = 0$.

Before we describe the algorithm, let us take a look at the action of Hadamard on $|i\rangle$. As an assignment, show that,

$$H^{\otimes k}|i\rangle = \frac{1}{\sqrt{2^k}} \sum_j (-1)^{i.j}|j\rangle,$$

where $i.j$ is the bitwise inner product between two binary strings.

Exercise 18. What is the inverse of $H^{\otimes n}$?

Abusing the notation $|0\rangle = |00\cdots0\rangle$, then

$$H^{\otimes k}|0\rangle = \frac{1}{\sqrt{2^k}} \sum_j |j\rangle.$$

This also means: if $|\psi\rangle = \sum_j \alpha_j|j\rangle$ then $\langle\psi|H^{\otimes k}|0\rangle = \frac{1}{\sqrt{2^k}} \sum_j \alpha_j$.

In other words, Hadamard spreads the amplitude to every state when applied to $|0\rangle$. It also collects the amplitude to state $|0\rangle$ when applied to a state $|\psi\rangle$ (Hadamard is its own inverse).

Exercise 19. Before looking at the circuit below, can you guess the Deutsch-Jozsa algorithm?

From the observations about the Hadamard gate, the Deutsch-Jozsa algorithm can be inferred. We start with $|0,1\rangle$, where the first part of the register is a k bit string. Using Hadamard, we can transfer it to equal superposition of all index states,

$$\frac{1}{\sqrt{2^k}}\sum_j |j,1\rangle.$$

Then we apply the phase query oracle,

$$\frac{1}{\sqrt{2^k}}\sum_j (-1)^{x_j}|j,1\rangle.$$

Again, using Hadamard, we can collect the amplitudes on state $|0\rangle$.

Exercise 20. Show, if the input is constant then we get state $|0\rangle$, else if the input is balanced then the amplitude on the state $|0\rangle$ is zero.

Measuring in the standard basis, if we get state $|0\rangle$ then algorithm will answer *constant* else it will answer *balanced*.

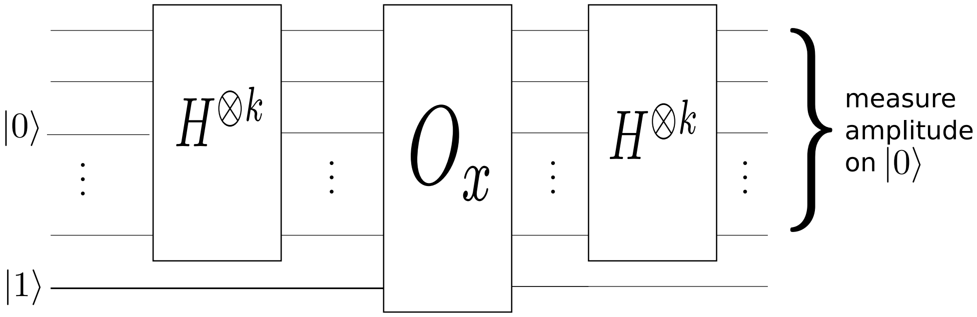

Fig. 7. Deutsch's Algorithm

Exercise 21. Convince yourself that the circuit given in Fig. 7 works.

Notice that this algorithm is without error. Any classical deterministic algorithm will require at least $O(n)$ queries of the classical oracle to determine the correct answer, while this algorithm takes only one query to the quantum oracle.

Exercise 22. Why will any classical deterministic algorithm require $O(n)$ queries?

4 Assignment

Exercise 23. Read about super-dense coding. What is the relation between teleportation and super-dense coding?

Exercise 24. Let $|\psi\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$ be a quantum state. Find a quantum state orthogonal to $|\psi\rangle$. Write the basis change operator from standard basis to these new states.

Exercise 25. Write the matrix representation of the gate which takes $|00\rangle$ to $|00\rangle$, $|01\rangle$ to $|11\rangle$, $|10\rangle$ to $|10\rangle$ and $|11\rangle$ to $|01\rangle$. What gate is this? What is its action on $(\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

Exercise 26. Create a circuit using Hadamard gate and CNOT gate which creates $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ from $|00\rangle$ state.

References