

Lecture 3

*Lecturer: Henry Yuen**Scribes: Seyed Sajjad Nezhadi, Angad Kalra
Nora Hahn, David Wandler*

1 Overview

In Lecture 3, we started off talking about Quantum Teleportation and were introduced to Quantum Computation. We explore the Quantum Circuit model, various quantum gates, application of quantum entanglement, and learn to do a step-by-step analysis of quantum algorithms.

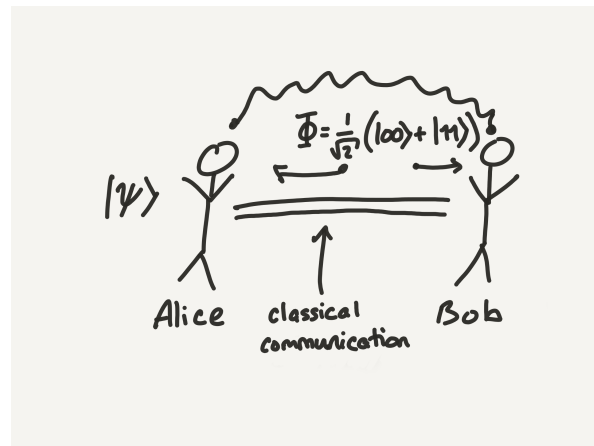
2 Quantum Teleportation

Quantum Teleportation is a protocol for transferring quantum states using only classical communication and entanglement. To do this we construct and analyze our first quantum circuit.

2.1 Quantum Teleportation

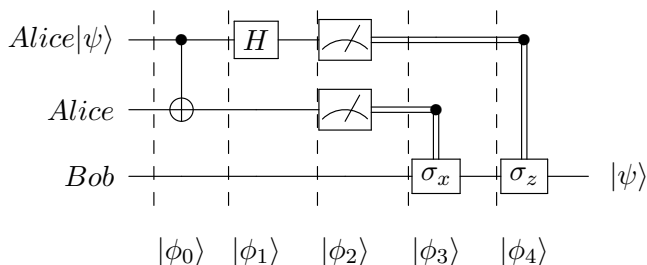
Imagine Alice and Bob are very far apart but have the means of classical communication (eg. phone line). Alice wishes to communicate her qubit, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \in \mathbb{C}^2$, with Bob. However, physically transferring the qubit would be difficult as quantum states are fragile. Even if she knew the classical description of her qubit (i.e. the amplitudes α and β) and wanted to tell Bob, Alice could be required to send infinitely many bits as α and β could be transcendental numbers.

However, with quantum entanglement, we can do this much more efficiently. We will show as long as Alice and Bob each hold 1 qubit of the entangled EPR pair $|\Phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \in \mathbb{C}^2$, we can teleport $|\psi\rangle$ with only two classical bits of communication!




2.2 Circuit Implementation

Quantum circuits represent applications of gates (unitary operations) and measurements to the qubits. The first two qubits are for Alice and the third for Bob. The circuit wires move forward with time. At each time step i we get the quantum state $|\phi_i\rangle$.



Gate Descriptions

1. *CNOT* – 2 qubit gate: $\text{CNOT } |a, b\rangle = |a, a \oplus b\rangle$ where a is the **control** and b is the **target**. In the circuit diagram, CNOT is represented by the black circle, connected by a line to an \oplus symbol. The black circle represents the control qubit, and the \oplus is on the target qubit.
2. *Hadamard* – single qubit unitary: $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. This single-qubit unitary has the following behavior on the standard basis: $H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and $H|1\rangle = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.
3.  – Measurement in the standard basis $\{|0\rangle, |1\rangle\}$

Alice carries out the gates up to the measurement part. She obtains two classical bits (that is what the double-lines coming out of the measurement gates indicate), called z (top wire) and x (middle wire). She relays these two classical bits to Bob over their classical communication channel.

Bob, after receiving (z, x) , will first apply a *controlled Pauli X* gate depending on the value of x . If $x = 0$, then he does nothing to his qubit (or in other words, applies the identity). If $x = 1$, then he applies the Pauli X gate, which is a single qubit bit flip: $\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. This single-qubit unitary has the following behavior on the standard basis: $\sigma_X|0\rangle = |1\rangle$, and $\sigma_X|1\rangle = |0\rangle$.

Then, if $z = 0$, he does nothing (i.e. applies the identity) to his qubit; otherwise, if $z = 1$, then he applies the Pauli Z gate, which is: $\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. This single-qubit unitary has the following behavior on the standard basis: $\sigma_Z|0\rangle = |0\rangle$, and $\sigma_Z|1\rangle = -|1\rangle$.

Note: Here, one can think of the controlled Pauli X gate as being equivalent to a CNOT gate where the control qubit is classical.

2.3 Circuit Analysis

We claim that, after the protocol is completed, Bob's qubit (the third qubit of the circuit) is in the state $|\psi\rangle$ – the state that Alice originally received! To show this, we will compute the state of the circuit at each time step.

$$|\phi_0\rangle = |\psi\rangle \otimes |\Phi\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle) \quad (1)$$

$$|\phi_1\rangle = \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle) \text{ by applying CNOT gate to the first two qubits.} \quad (2)$$

$$\begin{aligned} |\phi_2\rangle &= H|\phi_1\rangle = \frac{1}{\sqrt{2}}(\alpha H|0\rangle|00\rangle + \alpha H|0\rangle|11\rangle + \beta H|1\rangle|10\rangle + \beta H|1\rangle|01\rangle) \\ &= \frac{1}{\sqrt{2}}(\alpha|+\rangle|00\rangle + \alpha|+\rangle|11\rangle + \beta|-\rangle|10\rangle + \beta|-\rangle|01\rangle) \\ &= \frac{1}{2}(\alpha|000\rangle + \alpha|100\rangle + \alpha|011\rangle + \alpha|111\rangle + \beta|010\rangle - \beta|110\rangle + \beta|001\rangle - \beta|101\rangle) \end{aligned} \quad (3)$$

Step 3: Alice measures both her qubits to obtain outcomes (z, x) . Suppose she gets:

- $(z = 0, x = 0)$. To get the post-measurement state, only look at terms with "00" in first two positions. Unnormalized, the post-measurement state is

$$\frac{1}{2}|00\rangle \otimes (\alpha|0\rangle + \beta|1\rangle) = \frac{1}{2}|00\rangle \otimes |\psi\rangle$$

Since Bob receives $(z = 0, x = 0)$, he doesn't apply σ_X nor σ_Z , and his qubit is in the state $|\psi\rangle$. This outcome occurs with probability $1/4$.

- $(z = 0, x = 1)$. The unnormalized post-measurement state is $\frac{1}{2}|01\rangle \otimes (\beta|0\rangle + \alpha|1\rangle)$. Bob applies σ_X to his qubit, yielding

$$\sigma_x(\beta|0\rangle + \alpha|1\rangle) = |\psi\rangle$$

for Bob's qubit. This outcome also occurs with probability $1/4$.

- $(z = 1, x = 0)$. The unnormalized post-measurement state is $\frac{1}{2}|10\rangle \otimes (\alpha|0\rangle - \beta|1\rangle)$. Bob applies σ_Z to his qubit, yielding

$$\sigma_z(\alpha|0\rangle - \beta|1\rangle) = |\psi\rangle$$

for Bob's qubit. This outcome also occurs with probability $1/4$.

- $(z = 1, x = 1)$. The unnormalized post-measurement state is $\frac{1}{2}|11\rangle \otimes (-\beta|0\rangle + \alpha|1\rangle)$. Bob applies σ_X , and then σ_Z to his qubit, yielding

$$\sigma_z(\sigma_x(-\beta|0\rangle + \alpha|1\rangle)) = \sigma_z(\alpha|0\rangle - \beta|1\rangle) = |\psi\rangle$$

for Bob's qubit. This outcome also occurs with probability $1/4$.

Therefore with probability $P = 4 \times \frac{1}{4} = 1$ the third qubit is $|\psi\rangle$. Bob always gets Alice's qubit!

A few notes. One thing you may notice is that this seems reminiscent of cloning a state. However, this does not violate the No-Cloning Theorem (which we saw in Lecture 1), because at the end of the protocol, only Bob has the state $|\psi\rangle$, and Alice only has the measurement outcomes.

Another question that arises is, whether teleportation can be achieved by sending fewer than two classical bits. It turns out two classical bits are necessary! Also, *some* communication is necessary; otherwise this would mean that Alice could transmit information to Bob instantaneously. In other words, if Alice tries to teleport quantum states to Bob without sending any information, Bob would not be able to recover the state $|\psi\rangle$ (in fact, from his point of view, he just has a uniformly random bit).

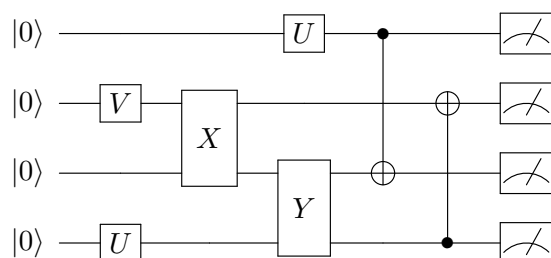
3 Quantum Computation

3.1 The Quantum Circuit Model

A **quantum computation** is anything that can be computed by a quantum circuit. Quantum Teleportation was an example of quantum computation. For our purposes, we will define a quantum circuit to be a combination of single- and two-qubit gates, as well as measurements in the standard basis. One could also consider models of quantum circuits with say three-qubit gates, but this does not really affect the model.

Furthermore, when discussing quantum algorithms, we will assume that the measurements all occur at the very end of the circuit. This is not true of the quantum teleportation circuit, but that is for the “communication setting”, where one party has to measure in order to communicate with another party. For algorithms, we can defer all the measurements to the end. This is without loss of generality in the algorithms setting, as by something called the “Principle of Deferred Measurement” we can push any intermediate measurements to the end and have an equivalent quantum circuit.

Here is how we interpret the circuit:



- The n qubits usually start off in the $|0\rangle$ state (but not always).
- Time runs from left to right.
- At each discrete time step, a gate is applied to at most 2 qubits.
- A gate is a unitary operation.

Thus, if we ignore the measurements at the end of a quantum circuit, every quantum circuit C corresponds to a unitary transformation acting on n qubits.

Question: Why does this capture quantum computation?

Given n qubits, what can happen to it? According to the postulates of QM that we talked about, there are measurements and unitary transformations. The quantum circuit has the measurements, but what about the unitaries? The unitaries in a quantum circuit act non-trivially on a few qubits, but a general unitary on n qubits might act on all n qubits in a very complicated fashion.

This next Theorem justifies why we can restrict our attention to two-qubit unitaries.

Theorem 1. *Every unitary U on n qubits can be written as a product of two-qubit gate $\{g_i\}$*

$$U = g_T \cdots g_2 g_1.$$

for some T .

Proof. Proof can be found in Chapter 4 of Nielsen and Chuang. □

Important note. In the Theorem above, when we say g_i is a two-qubit gate, that does not mean that g_i is two-qubit unitary in the product $g_T \cdots g_2 g_1$. That wouldn't make sense, as a two-qubit unitary is a 4×4 matrix and U is a $2^n \times 2^n$ matrix.

Rather, we mean that each g_i looks like $\hat{g}_i \otimes I$ where \hat{g}_i is a 4×4 unitary matrix, and I is the identity matrix on $\mathbb{C}^{2^{n-2}}$. We also have to pay attention to how the tensor product is interpreted, too. For example, if g_i represents a two-qubit gate acting on qubits 1 and 2, then $g_i = \hat{g}_i \otimes I$ is interpreted as usual. If g_i acts on qubits 3 and 4, then we can write $g_i = I_{1,2} \otimes \hat{g}_i \otimes I_{5,\dots,n}$ where $I_{1,2}$ is the identity on qubits 1 and 2, and $I_{5,\dots,n}$ is the identity on qubits 5 through n . What if g_i acts on qubits 1 and 3 (or any non-consecutive pair of qubits)? There is unfortunately no nice way to write this, so we just write $g_i = \hat{g}_i \otimes I$ where we interpret the identity as acting on qubits 2, 4, 5, ..., n .

The point is, one needs to pay attention which qubits a gate acts on.

4 Quantum Computation and Quantum Circuit Model

4.1 Discrete, universal gate sets

Consider a general circuit, that means the circuit consists of arbitrary two qubit gates. How many qubit gates are there which you can apply? Infinite many! But let's suppose that you can only apply gates from some discrete, finite set $\Lambda \in \{\text{two-qubit unitaries}\}$. Is there any chance to construct a general circuit only with gates from Λ ? If the set Λ is chosen carefully, the circuits with gates from Λ can approximate any general circuit. And in this case Λ is called a universal gate set. Hereafter, the meaning of the terms *approximate* and *universal gate set* will be specified. Notice that the final measurements are omitted for now, such that circuits are unitaries.

Definition 2 (ε -approximation). *The circuit C ε -approximates circuit C' if*

$$E(C, C') = \max_{|\psi\rangle} \|(C - C')|\psi\rangle\| \leq \varepsilon.$$

Definition 3 (universal gate set). *A gate set Λ is **universal** if for all general circuits C the following statement holds:*

$\forall \varepsilon > 0 \exists$ a Λ -circuit C' (gates only from Λ) s.t. $E(C, C') \leq \varepsilon$.

The attempt to approximate any general circuit out of gates from some finite set Λ seems to be impossible. But the proof idea resembles the relation between the sets of rational and real numbers. The rational numbers form a dense subset of the real numbers. Analogously, the idea is to show that $Z(\Lambda) = \{\text{all possible } \Lambda\text{-circuits}\} \subseteq \{\text{all } n\text{-qubit unitaries}\}$ is dense in the set of all n -qubit unitaries.

In the following, an example for an universal gate set is given.

Example. Consider the discrete, finite set

$$\Lambda = \{H, S, T, CNOT\},$$

where H (Hadamard gate), S (phase gate) and T ($\pi/8$ gate) [NC11] are single-qubit gates and CNOT is the only two-qubit gate. More precisely, the gates are defined as follows.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Consider a universal gate set. We have seen that there exists a Λ -circuit C' which ε -approximates the general circuit C . But up to now we don't know anything about the size of C' . The following theorem is one of the most important results of quantum computing; it will allow us to conclude that there is always a circuit C' which is not much bigger than C .

Theorem 4 (Solovay-Kitaev Theorem). *Let Γ denote a set of single-qubit unitaries. Suppose that*

- Γ generates a dense subgroup of $SU(2)$

- Γ is closed under inverse.

Then any single-qubit unitary U can be ε -approximated by a product of at most $\log^d(1/\varepsilon)$ gates from the set Γ for some constant $1 \leq d < 4$.

Proof. A proof can be found in the appendix of, e.g., Nielsen and Chuang. □

Corollary 5. Let Λ be a gate set such that $\Lambda = \{CNOT\} \cup \Gamma$ where Γ that satisfies the conditions of the Solovay-Kitaev theorem. Then for any circuit C , there exists a Λ -circuit C' that ε -approximates C and the size of C' (the number of gates) is at most the size of C times $\log^d(|C|/\varepsilon)$.

Proof. It turns out that any circuit C consisting of (arbitrary) two-qubit gates has an *equivalent* circuit C' consisting of single-qubit gates and CNOT gates (a proof can be found in Nielsen and Chuang). Using the Solovay-Kitaev theorem, we can approximate every single-qubit gate in C' up to $\varepsilon/|C|$ accuracy using $\log^d(|C|/\varepsilon)$ gates from Γ . □

How do we apply this Corollary to the gate set described above? It does not appear to be closed under inverse, after all. However, we can augment Λ so that it has inverses: notice that $S^{-1} = S^3$ and $T^{-1} = T^7$. Adding this in will make Λ satisfy the requisite conditions.

In the original theorem the constant was approximated by $d \approx 3.97$. Over the years, the estimate has been improved and today it is proved that the statement even holds for $d = 1$.

This theorem is very important for quantum computing in practice because developed quantum algorithms often contain various two (or more) qubit gates. But a real quantum computer cannot handle an algorithm which uses any kind of gates. Due to the theorem, it is possible to compile the code down to the gate set of the used machine. Furthermore, there already are fast algorithms to compute C' out of C .

4.2 Oracle access to functions

Two important types of problems which shall be solved are decision and search problems because many problems can be phrased in the same way. Therefore a lot of the quantum algorithms in this course will be of one of these types.

The input for both problems is an unknown function $f : X \rightarrow Y$ but the questions vary as you can see in the following.

- **decision:** Does f have some property p ? \rightarrow yes/no
- **search:** Find $x \in X$ s.t. $f(x) = 1$

Picture of the query model

Now we want to see how these quantum algorithms are working. The quantum algorithms get access to the unknown function f through query access. As you can see in the following picture, the circuits get access to another unitary which is called the **oracle to f** and shortened as U_f .

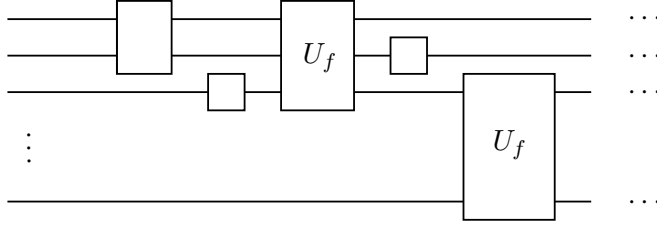


Figure 1: A picture of the query model

But what is U_f precisely? How does the oracle behave? As the name *oracle* suggests, we can ask for the answer but cannot know how it works inside.

The oracle can be expressed as a unitary operator. Without loss of generality, assume the boolean function $f : \{0, 1\}^m \rightarrow \{0, 1\}$. Consider a m -qubit x as input register and b as target qubit. Then the oracle behaves as $(m + 1)$ -qubit unitary as follows.

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

Generally, if the oracle U_f is applied to a function $f : \{0, 1\}^m \rightarrow \{0, 1\}^n$, that means

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

where x is a m -qubit register and b is a n -qubit register. The notation $b \oplus f(x)$ is the n -qubit string such that the i 'th bit is $b_i \oplus f(x)_i$ (the parity of b_i and the i 'th bit of the output $f(x)$).

Hence, U_f is a $2^{m+n} \times 2^{m+n}$ unitary.

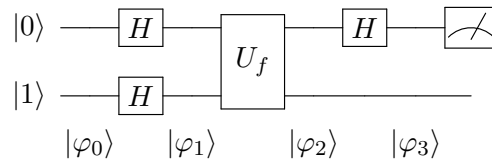
5 Quantum Algorithms

In this section we start to look at particular algorithms that can be implemented on quantum computers and how they improve on classical algorithms.

5.1 Deutsch's Algorithm

This algorithm solves the decision problem: *given $f : \{0, 1\} \rightarrow \{0, 1\}$, decide if f is constant or not constant*. This is a very simple problem, we could just check the two possible inputs, but Deutsch's algorithm is able to do it using just one query!

The algorithm is given by the following circuit:



To prove that this algorithm solves the decision problem, we follow through the circuit and write the quantum state at each step. To start we have

$$|\varphi_0\rangle = |0\rangle \otimes |1\rangle$$

Then we apply the Hadamard operator to both qubits:

$$\begin{aligned} |\varphi_1\rangle &= H|0\rangle \otimes H|1\rangle = |+\rangle \otimes |-\rangle \\ &= \frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle) \end{aligned}$$

Then we apply the oracle operator, U_f :

$$\begin{aligned} |\varphi_2\rangle &= \frac{1}{2} (|0, 0 + f_0\rangle - |0, 1 + f_0\rangle + |1, 0 + f_1\rangle - |1, 1 + f_1\rangle) \\ &= \frac{1}{2} (|0, f_0\rangle - |0, \bar{f}_1\rangle + |1, f_1\rangle - |1, \bar{f}_1\rangle) \end{aligned}$$

Notice that here we used the shorthand $f_i = f(i)$ and used \bar{f}_i to indicate the opposite of f_i . From here, we again apply the Hadamard operator to the first qubit:

$$\begin{aligned} |\varphi_3\rangle &= \frac{1}{2} (|+\rangle|f_0\rangle - |+\rangle|\bar{f}_0\rangle + |-\rangle|f_1\rangle - |-\rangle|\bar{f}_1\rangle) \\ &= \frac{1}{2\sqrt{2}} (|0\rangle \otimes (|f_0\rangle - |\bar{f}_0\rangle + |f_1\rangle - |\bar{f}_1\rangle) + |1\rangle \otimes (|f_0\rangle - |\bar{f}_0\rangle - |f_1\rangle + |\bar{f}_1\rangle)) \end{aligned}$$

Now, we measure the first qubit. There are two possibilities for the distribution of outcomes. If f is constant, then $f_0 = f_1$, so the state is:

$$|\varphi_3\rangle = \frac{1}{2\sqrt{2}} (|0\rangle \otimes (2|f_0\rangle - 2|\bar{f}_0\rangle) + |1\rangle \otimes 0) = \frac{1}{\sqrt{2}} |0\rangle \otimes (|f_0\rangle - |\bar{f}_0\rangle)$$

Therefore, we will measure 0 with probability 1. On the other hand, if f is not constant, then $f_1 = \bar{f}_0$, so the state is:

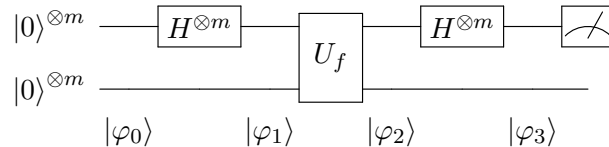
$$|\varphi_3\rangle = \frac{1}{\sqrt{2}} |1\rangle \otimes (|f_0\rangle - |\bar{f}_0\rangle)$$

In this case, we measure 1 with probability 1. Thus, the algorithm solves the problem of deciding whether or not f is constant.

5.2 Simon's Algorithm

Simon's algorithm is a precursor to Shor's algorithm, that was created by U of T grad, Daniel Simon. The search problem that the algorithm solves takes as its input a function $f : \{0, 1\}^m \rightarrow \{0, 1\}^m$ that is guaranteed to have the property that $\exists s \in \{0, 1\}^m$ such that $\forall x, y \in \{0, 1\}^m f(x) = f(y) \iff x = y$ or $x = y \oplus s$. The problem is to find s .

Classically, this can only be done by checking various pairs until you find two that collide. This takes at least $\sqrt{2^m}$ queries. Simon's algorithm, on the other hand, only needs $\mathcal{O}(m)$ queries. It runs on a $2m$ -qubit quantum computer. The following subroutine is the basis of Simon's algorithm:



Notice that the notation $|0\rangle^{\otimes m}$ means $|0\rangle_1 \otimes |0\rangle_2 \otimes \dots \otimes |0\rangle_m$ and $H^{\otimes m}|0\rangle^{\otimes m}$ means we apply H to each of the m qubits. Also, since f outputs to $\{0, 1\}^m$ we now need m qubits for the output. Just

as in Deutsch's algorithm, we look at the state of the qubits after each step in the circuit. Our initial state is

$$|\varphi_0\rangle = |0\rangle^{\otimes 2m}$$

We then apply the Hadamard gate to the first m qubits:

$$\begin{aligned} |\varphi_1\rangle &= (H^{\otimes m}|0\rangle^{\otimes m}) \otimes |0\rangle^{\otimes m} \\ &= \frac{1}{2^{m/2}} \left(\sum_{x \in \{0,1\}^m} |x\rangle \right) \otimes |0\rangle^{\otimes m} \end{aligned}$$

After this we apply the oracle operator

$$|\varphi_2\rangle = \frac{1}{2^{m/2}} \sum_{x \in \{0,1\}^m} |x\rangle |f(x)\rangle$$

Finally, we reapply the Hadamard operator to the first m qubits

$$\begin{aligned} |\varphi_3\rangle &= \frac{1}{2^{m/2}} \sum_{x \in \{0,1\}^m} (H^{\otimes m}|x\rangle) |f(x)\rangle \\ &= \frac{1}{2} \sum_{x \in \{0,1\}^m} \sum_{y \in \{0,1\}^m} (-1)^{x \cdot y} |y\rangle |f(x)\rangle \end{aligned}$$

where $x \cdot y = \sum_i x_i y_i \pmod 2$. The last line is left as an exercise for the reader to confirm. Using this result to find s is left for the next lecture.

TO BE CONTINUED

References

- [NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.