CSE 434 : Introduction to Quantum Computation

University of Washington

Andrew Chen

Spring 2026

Hello and welcome! This is my lecture notes on CSE 434: Introduction to Quantum Computation. As the course title suggests, this course is an undergraduate-level exposure to many quantum computation theories and techniques. The professor is **Andrea Coladangelo**, and we meet TTh at **10:00 am** for lectures. There are no required textbooks for this course. Also note that theorem names might not necessarily be accurate; it's probably just whatever my textbook / professor said it is.

The goal of these lecture notes is to write **understandable** math. As the great Albert Einstein put it, "If you can't explain it to a six year old, then you don't understand it yourself". The hope is that anyone coming across these notes (like you!) will be able to at least take away the gist of these concepts. Should you find any errors in my mathematics, please contact me at zchen66@uw.edu

Contents

List of Definitions

List of Theorems

1 Lecture 01: Mar. 31st

Course logistics. Introduction to quantum computation. Double-slit experiment. Complex numbers overview. Linear algebra review.

1.1 Course logistics

This course will go over the basics of quantum computation with an emphasis on theory. Students will be assessed through 3 ways – homeworks (of which there are 6), a take-home midterm exam, and an in-person final exam.

The first half of the class will focus on quantum *information*, and the second half will ocus on quantum *computation*.

1.2 Quantum computation overview

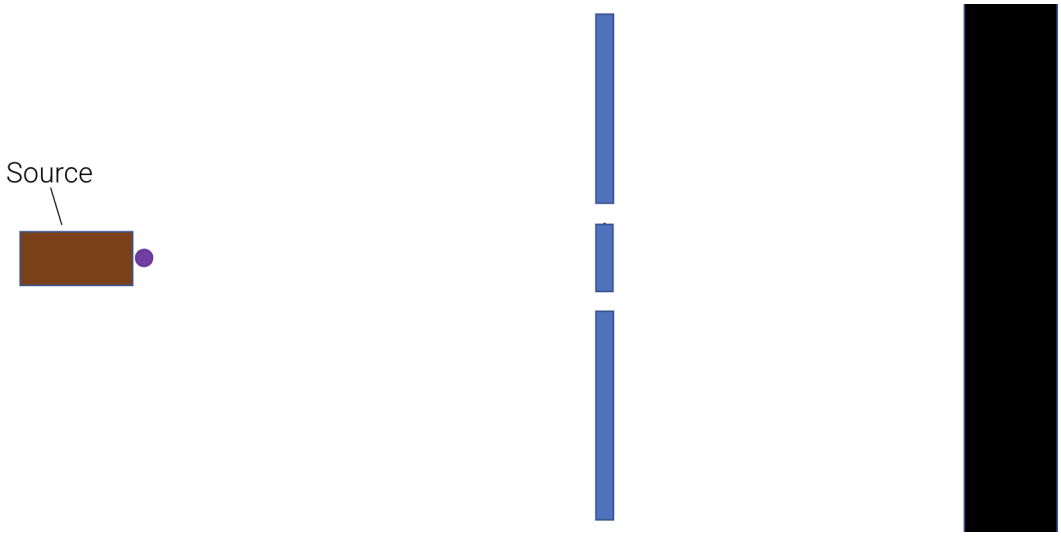
To begin, quantum computation is a new paradigm for computation that is based on the laws of *quantum mechanics*. But quantum mechanics sucks and no one understands it (famously), so why do we care??

Well, there are actually *many* applications, which includes concepts in fields like chemistry, machine learning, cryptography, etcetc.

To address the elephant in the room, many of the quantum algorithms / concepts are actually simple heuristics at this time, since humanity don't yet have a quantum computer large enough to actually compute these.

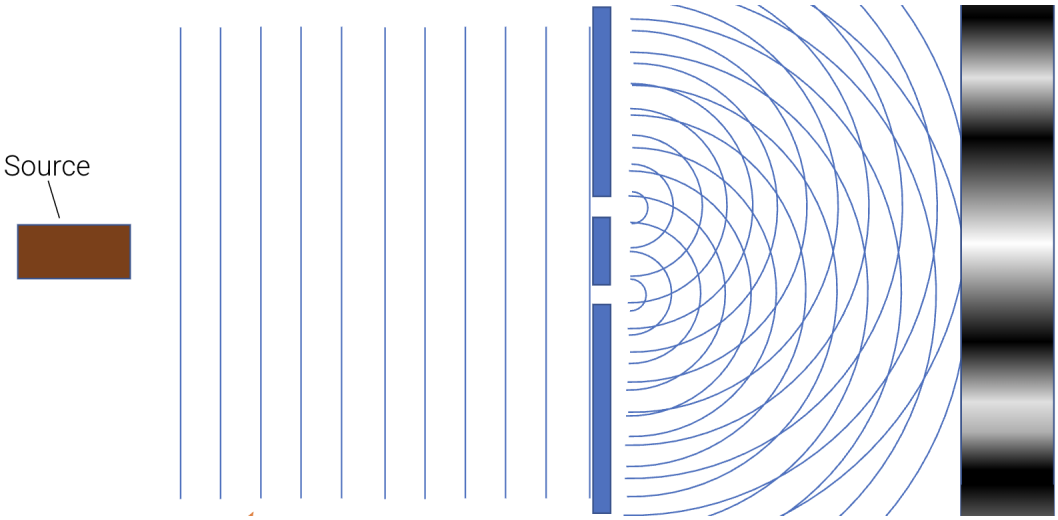
1.3 The double-slit experiment

Here's the set-up:



To begin, let's assume the source is sending ping pong balls through the slits. Intuitively, only some will make it through the slits, and leave a mark on the detector.

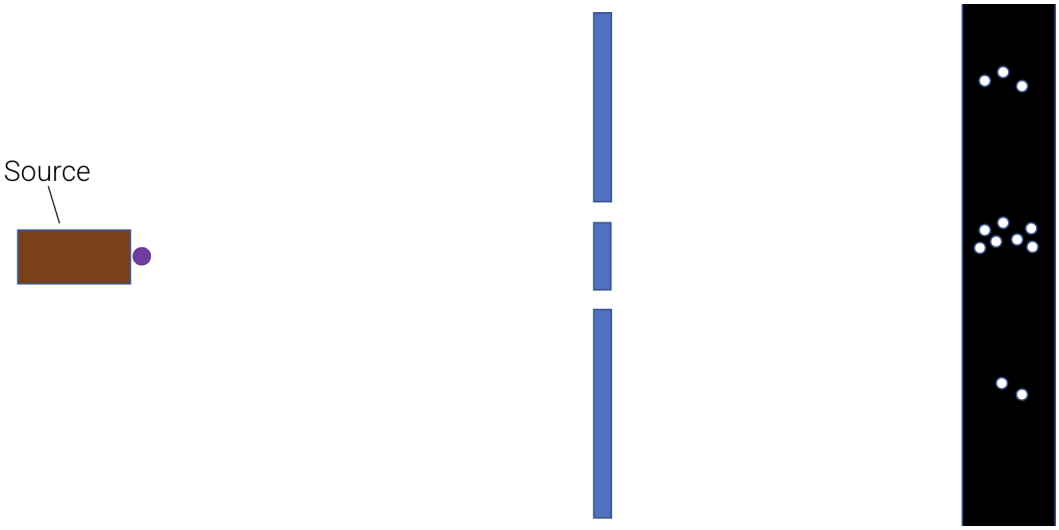
Now let's think. What if the source is sending water waves instead? Take a look.



The waves collide through the two slits, creating circular waves which forms constructive and destructive interferences. As a result, our detector will mark the highest peaks and lowest valleys. This is what people refer to as the **interference pattern**.

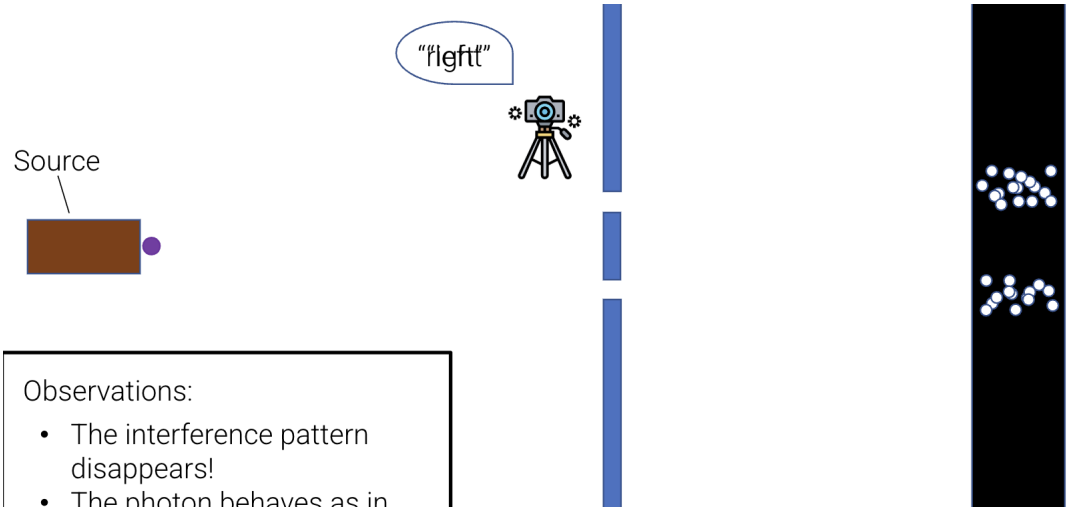
Great. So far it's all been pretty expected patterns. Let's take a look at a third case now, where the source is sending *photons* (tiny packets of light). We can think of it like a very tiny ping pong ball.

So what would we expect to see now? Well, under the assumption that it's a tiny tiny ping pong ball, then we'd expect it to behave as such right?



WRONG. Surprisingly perhaps, it creates an interference pattern much like the case with water waves. But slightly different from the water wave, because it's "tiny ping pong balls", we see discrete hits on the detector rather than a gradient of waves.

Since we're sending photons one-by-one, the only plausible explanation of this is that the photon is somehow... interfering with itself? Meaning it passes through both slits? Let's test that.



So all of a sudden... the photons stopped interfering with itself. What an annoying ahh photon. The underlying message is:

Quantum mechanics is the theory that is able to explain this absurdity.

Quantum computation is all about orchestrating this interference in such a way that the resulting "pattern" tells us something useful (like the solution of a problem). Trippyyyyyyyy...

There are many of these quantum systems that exists. But in this course, we abstract away all the nasty stupid physics and just say

Definition (qubit)**.** A quantum system with two degrees of freedom.

1.4 Complex numbers and linear algebra review

Okay, cool part's over. Time to do math. To define complex numbers, we first have to define an "imaginary unit" i. By definition, $i^2 = -1$, so we can think of it as "$i = \sqrt{-1}$", but strictly speaking this is not correct.

Building off of this, imaginary numbers are just real multiples of i, so $y \cdot i$ for any $y \in \mathbb{R}$.

Building off even further, complex numbers are numbers of the form $a + ib$, where $a, b \in \mathbb{R}$. We refer to a as the "real part" and b as the "imaginary" part.

One way to represent these numbers is through a 2D plane called the **complex plane**, where for some $z = a + ib$, we represent it with the coordinate point (a, b).

There are now some cool properties. Let $z = a + ib$ and $z' = a' + ib'$. We define the **magnitude** as

$$|z| = \sqrt{a^2 + b^2}$$

Thinking in the complex plane, that is the length of the segment that connects the origin to z. Furthermore, **addition** is defined as

$$z + z' = (a + a') + i(b + b')$$

In the complex plane, this is represented as vector addition. Continuing on, **multiplication** is defined to be commutative, associative, and distributive:

$$\begin{aligned} z \cdot z' &= (a + ib) \cdot (a' + ib') \\ &= aa' + iab' + iba' + i^2 bb' \\ &= aa' + iab' + iba' - bb' \\ &= (aa' - bb') + i(ab' + ba') \end{aligned}$$

Lastly, we define an operation somewhat unique to the complex number called **conjugation**. We define a conjugate $\bar{z}$ of z to be

$$\bar{z} := a - ib$$

notice the sign of the imaginary part has been flipped. One nice thing about the conjugate is that

$$z \cdot \bar{z} = (a + ib)(a - ib) = a^2 + b^2 = |z|^2$$

We now pivot to a linear algebra review. In this class, we'll consider vectors in $\mathbb{C}^n$. Let

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \quad w = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$$

All addition, scalar multiplication, and norms and defined as usual. I'm lazy and I won't write it out.

Wait actually I'll write out the norm. Since each element is in $\mathbb{C}$, we actually define the norm of a vector v to be

$$\|v\| = \sqrt{|v_1|^2 + |v_2|^2 + \cdots + |v_n|^2}$$

this is because we want to require the norm to be a real number. Similarly, inner products are defined as

$$\langle v, w \rangle = \overline{v_1} \cdot w_1 + \cdots + \overline{v_n} \cdot w_n$$

Inner products are perhaps the most important thing we will need to have DOWN in this class (allegedly). But why did we define it with the conjugates of the first elements?

Well that's because we need it to satisfy the following 3 properties:

1. $\langle v, v \rangle = 0$ if and only if $v = 0^n$.
2. Inner product is "linear" (up to complex conjugation) in both arguments: let $u, v, w \in \mathbb{C}^n$ and $\alpha, \beta \in C$. Then,

$$\langle u, \alpha \cdot v + \beta \cdot w \rangle = \alpha \cdot \langle u, v \rangle + \beta \cdot \langle u, w \rangle$$

   Similarly,

$$\langle \alpha v + \beta w, u \rangle = \bar{\alpha} \langle v, u \rangle + \bar{\beta} \langle w, u \rangle$$

3. Inner product is "symmetric" (up to complex conjugation):

$$\langle u, v \rangle = \overline{\langle v, u \rangle}$$

2 Lecture 02: Apr. 2nd

Linear algebra review: orthogonality, linear combinations, basis & orthonormal basis, linear transformations (and inverse linear transformations), **unitary transformations**. Qubits, representations of qubits, ket notation, qubit measurement and Born's rule.

2.1 Linear algebra review (cont'd)

To begin lecture, we picked up the linear algebra review from the previous lecture. Recall that in the previous lecture, we defined the "inner product" of two vectors $v, w \in \mathbb{C}^n$, and some important properties regarding the inner product.

One thing that wasn't mentioned last time is that we can alo express the norm of a vector using the inner product.

$$\|v\|^2 = |v_1|^2 + |v_2|^2 = \overline{v_1}v_1 + \overline{v_2}v_2 = \langle v, v\rangle$$

Let's continue. We now define orthogonality and linear combination.

Definition (Vector orthogonality)**.** Two vectors $v, w \in \mathbb{C}^n$ are **orthogonal** if $\langle v, w\rangle = 0$.

Definition (Linear combination)**.** We say $u \in \mathbb{C}^n$ is a **linear combination** of $v, w \in \mathbb{C}^n$ if there exists $\alpha, \beta \in \mathbb{C}$ such that

$$u = \alpha \cdot v + \beta \cdot w$$

Definition (Basis)**.** A set of vectors $B = \{u_1, \ldots, u_k\} \subseteq \mathbb{C}^n$ is called a **basis** if:

1. The vectors of B generate all vectors in $\mathbb{C}^n$ via linear combinations. In other words, for any $v \in \mathbb{C}^n$, there exists $\alpha_1, \ldots, \alpha_k$ such that

$$v = \alpha_1 \cdot u_1 + \cdots + \alpha_k u_k$$

2. The vectors of B are **linearly independent** – no vectors in the set can be expressed as a linear combination of the others.

A couple facts about bases:

Fact 1: Any basis of $\mathbb{C}^n$ has size n.
Fact 2: Any set of n *linearly independent* vectors is a basis of $\mathbb{C}^n$.
Fact 3: The *standard basis* of $\mathbb{C}^n$ is

$$e_1 = \begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}, \quad e_2 = \begin{bmatrix}0\\1\\\vdots\\0\end{bmatrix}, \quad \ldots, \quad e_k = \begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix}$$

Let's do a quick example.

Example 2.1. How do we find the coefficients α, β such that

$$\begin{bmatrix}3\\2\end{bmatrix} = \alpha \cdot \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} + \beta \cdot \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$$

The traditional way that we've all learned in a typical linear algebra class is to solve via a linear system of equations. But there actually is a simpler way.

Notice how $\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$ and $\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$ are unit vectors (have norm 1) and are orthogonal (form a basis in $\mathbb{C}^2$). We can take advantage of this.

Suppose the expression holds for some $\alpha, \beta \in \mathbb{C}$. Take the inner product with both sides:

$$\left\langle \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}3\\2\end{bmatrix} \right\rangle = \alpha \cdot \left\langle \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} \right\rangle + \beta \cdot \left\langle \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-2\end{bmatrix} \right\rangle = \alpha = \frac{5}{\sqrt{2}}$$

Similarly on the other side:

$$\left\langle \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}, \begin{bmatrix}3\\2\end{bmatrix} \right\rangle = \beta = \frac{1}{\sqrt{2}}$$

This works because we can take mega advantage of the linearity of the inner product operation.

We now talk about linear transformations.

Definition (Linear transformations)**.** A map $T : \mathbb{C}^n \to \mathbb{C}^n$ is a **linear transformation** if, for all $v, w \in \mathbb{C}^n$ and all $\alpha, \beta \in \mathbb{C}$,

$$T(\alpha \cdot v + \beta \cdot w) = \alpha \cdot T(v) + \beta \cdot T(w)$$

Some examples include:

- Identity: $T(v) = v$ for all v
- Rescaling: $T(v) = \alpha \cdot v$ for all v where $\alpha \in \mathbb{C}$
- Reflection about x-axis: $T\left(\begin{bmatrix}v_1\\v_2\end{bmatrix}\right) = \begin{bmatrix}v_1\\-v_2\end{bmatrix}$

This brings about an interesting question: how many linear transformations T are there such that $T\left(\begin{bmatrix}1/\sqrt{2}\\1/\sqrt{2}\end{bmatrix}\right) = \begin{bmatrix}1\\i\end{bmatrix}$ and $T\left(\begin{bmatrix}1/\sqrt{2}\\-1/\sqrt{2}\end{bmatrix}\right) = \begin{bmatrix}1\\0\end{bmatrix}$.

There actually is only one, by linearity! Recall that any vector $u \in \mathbb{C}^n$ can be written as

$$u = \alpha \cdot \begin{bmatrix}1/\sqrt{2}\\1/\sqrt{2}\end{bmatrix} + \beta \cdot \begin{bmatrix}1/\sqrt{2}\\-1/\sqrt{2}\end{bmatrix}$$

for some α, β, since the latter two vectors form a basis.

$$\begin{aligned} T(u) &= T\left(\alpha \cdot \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} + \beta \cdot \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}\right) \\ &= \alpha \cdot T\left(\begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}\right) + \beta \cdot T\left(\begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}\right) \\ &= \alpha \cdot \begin{bmatrix} 1 \\ i \end{bmatrix} + \beta \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{aligned}$$

Let's talk more about linear transformations.

Definition (Inverse of a linear transformation)**.** The inverse of a linear transformation $T : \mathbb{C}^n \to \mathbb{C}^n$ is a linear transformation, denoted ${}^{-1}T$, such that

$${}^{-1}T \circ T = T \circ^{-1} T = I$$

where I is the identity transformation. This menas for all $v \in \mathbb{C}^n$,

$${}^{-1}T(T(v)) = T({}^{-1}T(v)) = v$$

Now, there is a class of linear transformations that's particularly important in quantum information. These are "unitary transformations".

Definition (Unitary transformations)**.** A linear transformation is **unitary** if it maps any orthonormal basis to another orthonormal basis.

In other words, for any orthonormal basis $B = \{u_1, \ldots, u_n\}$, we have that $U(B) = B' = \{U(u_1), \ldots, U(u_n)\}$ is also an orthonormal basis.

Definition (Matrix representation)**.** For a linear transformation $T : \mathbb{C}^n \to \mathbb{C}^n$, its matrix representation is the $n \times n$ matrix with complex entries M_t such that, for all $v \in \mathbb{C}^n$

$$T(v) = M_t \cdot v$$

there $M_t \cdot v$ is the usual matrix-vector product.

Holy review this guy just taught all of basic linalg in one and a half lecture. Let's *finally* get to the good stuff.

2.2 What is a qubit?

Recall from last lecture that we defined a **qubit** to be the simplest quantum system, with just two degrees of freedom. It's the fundamental unit of quantum information that *generalizes* the classical bit.

To be more specific, recall that a *bit* is defined to be one of two possible discrete values $\in \{0, 1\}$. Now, a qubit is defined to be a vector in $\mathbb{C}^2$, which is a value in the 2-dimensional vector space of possible states. More precisely perhaps, the state of a qubit is a *unit vector* in $\mathbb{C}^2$.

Out of all possible qubit states, there are two special states for which we interpret as representing the "classical" states 0 and 1. These are

$$|0\rangle \coloneqq \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle \coloneqq \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Note that these two vectors form an orthonormal basis, which we will refer to as the "standard basis". And since they form a basis, any other qubit space can be represented as a linear combination of these two vectors.

Thus, in general, the state of a qubit is denoted

$$|\psi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

where $|\alpha|^2 + |\beta|^2 = 1$, since all qubits must have norm 1.

Naturally, we now can wonder how these quantum states change, or "evolve". They do so in one of the two following ways:

1. via a unitary transformation
2. via a **measurement**.

Measurement = evolve. This is because we cannot simple "read" a quantum state. There is no physical procedure that, given some qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ reads out α and β.

The axioms of quantum mechanics say that the only way to extract "classical" information from a quantum state is to make a "measurement". But how do we "measure" a quantum state?

It is an inherently probabilistic procedure that takes a quantum state and returns a *classical outcome* (and a post-measurement state). It must obey **Born's rule**.

Theorem 2.1 (Born's rule).
If we measure the state of $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we get

- *outcome "0" with probability $|\alpha|^2$*
- *outcome "1" with probability $|\beta|^2$*

The quantum state after *the outcome then strictly becomes $|0\rangle$ if the outcome was "0", and $|1\rangle$ otherwise.*

In other words, the state after the measurement "collapses" to the state corresponding to the observed outcome.

As a sanity check, recall that all quantum states have norm 1: $|\alpha|^2 + |\beta|^2 = 1$ since $|\psi\rangle$ is norm 1. So the probabilities described by Born's rule sum to 1.

3 Lecture 03: Apr. 7th

Uniform superposition, born's rule. Measurements in orthonormal bases, bra-ket inner product notation, Hadamard basis, uncertainty principle, globas phase, relative phase.

3.1 What is a qubit? (cont'd)

Picking up from last time, recall that the state of a qubit is a unit vector in $\mathbb{C}^2$, with the two classical states being $|0\rangle := \begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle := \begin{bmatrix}0\\1\end{bmatrix}$.

Taking advantage of the vector properties of states, we write the general state of a qubit to be

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix}\alpha\\\beta\end{bmatrix}$$

where, crucially, $|\alpha|^2 + |\beta|^2 = 1$. An example of this is the **uniform superposition** $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$.

Furthermore, there are two ways for a qubit to evolve is either through a (1) unitary transformation or a (2) measurement following **Born's rule**. We will mostly focus on the latter in this lecture.

We started the lecture with some examples on qubit measurement.

Example 3.1. Suppose we measure the state $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. What is the probability distribution over outcomes?

We simply square each α and β and get that we get "0" with probability $\frac{1}{2}$ and "1" with probability $\frac{1}{2}$.

Now suppose we measured the qubit and got the outcome "0", and we measure again. What happens?

We will now get "0" with probability 1, because upon the first measure, our qubit collapses to the state $|0\rangle$. Any subsequent measures will give "0".

3.2 Measurement in a different basis

The motivation of this subsection is that in the scope of quantum theory, there is nothing particularly special about the standard basis. We simply "prefer" it because it mirrors that of the classical theory.

In reality, nothing really changes when we switch to a different orthonormal basis. For example, when we measure in the basis $\{|b_0\rangle, |b_1\rangle\}$, we are simply asking "are you $|b_0\rangle$ or $|b_1\rangle$?". Because of this realization, we should update Born's rule.

Theorem 3.1 (Born's rule, revisited).
Let $|\psi\rangle = \alpha'|b_0\rangle + \beta'|b_1\rangle$. If we measure $|\psi\rangle$ in the basis $\{|b_0\rangle, |b_1\rangle\}$, we get

- *Outcome b_0 with probability $|\alpha'|^2$*
- *Outcome b_1 with probability $|\beta'|^2$*

Remark. One may notice that in the above case, we called the outcomes b_0 and b_1 instead of the traditional "0" or "1". There is nothing particularly special about these labels, and they are just names given to the two possible outcomes.

As such, in future notation, we will (almost) always just refer to the outcomes as "0" or "1".

We can write the above more compactly by writing α' and β' in terms of $|\psi\rangle, |b_0\rangle, |b_1\rangle$. How?

$$\alpha' = \langle b_0|\psi\rangle, \quad \beta' = \langle b_1|\psi\rangle$$

Here, $\langle b_0|\psi\rangle$ refers to the inner product between $|b_0\rangle$ and $|\psi\rangle$, in that order.

So then, we can once again rewrite Born's rule:

Theorem 3.2 (Born's rule, revisited, again).
If we measure $|\psi\rangle$ in the basis $\{|b_0\rangle, |b_1\rangle\}$, we get

- *Outcome "0" with probability $|\langle b_0|\psi\rangle|^2$*
- *Outcome "1" with probability $|\langle b_1|\psi\rangle|^2$*

Going back a bit, why is it important that we measure in an *orthonormal* basis? Because we have to guarantee that $|\alpha'|^2 + |\beta'|^2 = 1$. Let's take a look at an example.

Example 3.2. What happens if we measure $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ in basis $\{\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\}$?
Well, we'd get outcome "0" with probability 1.

It seems like we're doing using this $\frac{1}{\sqrt{2}}$ term a lot. Let's define some new notation.

Definition (Hadamard basis)**.** We denote

$$|+\rangle \coloneqq \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \qquad |-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

The basis $\{|+\rangle, |-\rangle\}$ is referred to as the **Hadamard basis**.

Let's do a thought experiment now. What happens if we measure $|0\rangle$ in the Hadamard basis? Notice that

$$|0\rangle = \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) + \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}|+\rangle + \frac{1}{\sqrt{2}}|-\rangle$$

Meaning we get outcome "0" with probability $\frac{1}{2}$, and probability "1" with probability $\frac{1}{2}$. But more simply perhaps, we could've just used Born's rule to say

$$\Pr[\text{“0”}] = |\langle +|0\rangle|^2 = \frac{1}{2}$$
$$\Pr[\text{“1”}] = |\langle -|0\rangle|^2 = \frac{1}{2}$$

The same thing holds for $|1\rangle$ in the Hadamard basis (trust me bro).

So notice what we've just discovered: if the measurement outcome in the standard basis is deterministic (i.e. the state is $|0\rangle$ or $|1\rangle$), then the outcome in the Hadamard basis is *uniformly random*! And vise versa.

Furthermore, after the Hadamard basis measurement, the state collapses to $|+\rangle$ or $|-\rangle$. This means we lose *all* information regarding whether we started with $|0\rangle$ or $|1\rangle$.

In some sense, the standard basis and the Hadamard basis are "incompatible" with each other, and measurements in both basis cannot be performed simultaneously. This is the heart of the **Uncertainty Principle**.

> **Intuition.** *Think of how we cannot know the position and momentum at the same time. This is exactly what it is! We cannot measure some quantity in both basis at the same time.*

Let's do more examples!

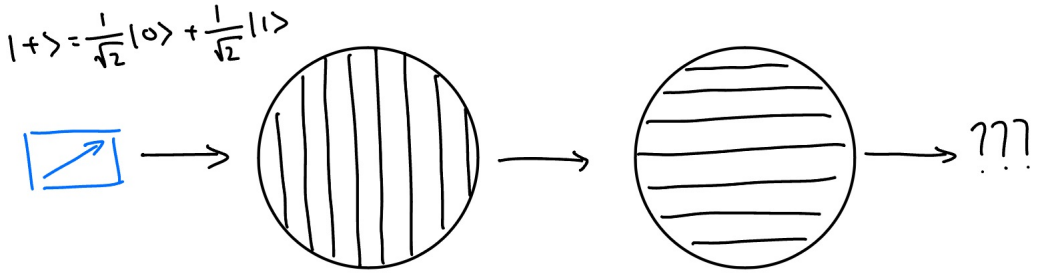
Example 3.3. What happens if we measure $\sqrt{\frac{2}{3}}|0\rangle + \sqrt{\frac{1}{3}}|1\rangle$ in the Hadamard basis?

Recall Born's rule. We have

$$\begin{aligned}\Pr[\text{“0”}] &= \left|\langle +| \left(\sqrt{\frac{2}{3}}|0\rangle + \sqrt{\frac{1}{3}}|1\rangle\right)\right|^2 \\ &= \left|\sqrt{\frac{2}{3}}\langle +|0\rangle + \sqrt{\frac{1}{3}}\langle +|1\rangle\right|^2\end{aligned}$$

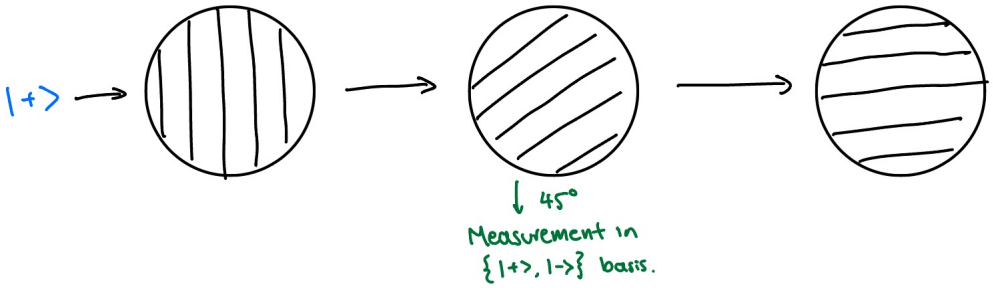
Let's connect all this to some physical examples perhaps. We mentioned in the first lecture that a representation of a qubit is seen in the polarization of a photon. A photon can be "vertically" or "horizontally" polarized (and anything in between).

Then consider the following experiment with polaroid filters, where each polaroid filter is a measurement in some basis (depending on the filter). Here, "vertically" is a filter for $|0\rangle$, and "horizontally" is a filter for $|1\rangle$.



Here, with probability 1, the photon does not come out at the end. This is because if the photo goes through the first filter, it has collapsed to $|0\rangle$ (the "vertical" state), meaning it will not be measured by the second polaroid filter.

However, now consider this add-on to the experiment:



Now the photon will come out with probability $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$! This is because:

1. The photon will go through the first filter with probability 1/2, and in the case it does go through, it collapses to $|0\rangle$
2. The photon in state $|0\rangle$ will go through the second filter with probability 1/2, in which case it collapses to $|+\rangle$
3. The photon in state $|+\rangle$ will go through the third filter with probability 1/2 again, in which case it collapses to $|1\rangle$

This is an amazing result that shatters our classical view of the world! How is it that, by adding *more* restrictions via filters, we end up *increasing* the probability of the photon surviving! This is the power of measuring in different bases.

*****

Let's talk a bit now about phases. We once again start with a thought experiment: is there any difference between $|\psi\rangle$ and $-|\psi\rangle$?

The short answer is: not a physical one. Since for any basis $\{|b_0\rangle, |b_1\rangle\}$, if $|\psi\rangle = \alpha|b_0\rangle + \beta|b_1\rangle$, then $-|\psi\rangle = -\alpha|b_0\rangle - \beta|b_1\rangle$. And since $|\alpha| = |-\alpha|$, $|\beta| = |-\beta|$, the distributions over outcomes is identical.

To generalize this, let $w \in \mathbb{C}$ with $|w| = 1$. We call w a **global phase** with polar representation $e^{i\theta}$. Now in general, for some $|\psi\rangle$, we treat $|\psi\rangle$ and $w \cdot |\psi\rangle$ as the same state for all intents and purposes.

> **Intuition.** *Multiplication by a* global phase *does not change the probability distribution over the outcomes of $|\psi\rangle$. Essentially, it acts like a rotation of the vector in $\mathbb{C}^2$ (a unitary transformation!)*

Continuing on this thought experiment, are $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ the same?

Of course not! You can distinguish them perfectly by measuring in the $\{|+\rangle, |-\rangle\}$ basis, in which case the first state will have outcome "0" with probability 1, and the second state will have outcome "1" with probability 1.

We refer to the extra minus sign on one of the states as a **relative phase**. Soon we will see that these relative phases play a very important role in quantum computation.

4 Lecture 04: Apr. 9th

Unitary evolution, identity map, Pauli X, Pauli Y, Pauli Z, quantum gates, Hadamard gate, quantum circuits. Elitzur-Vaidman tester, photon bomb experiment. Quantum key distribution, unconditional security.

4.1 Unitary evolution

So far, we've only talked about quantum states using one qubit. But we'll see today in lecture that even with a one-qubit system, there still are some very interesting applications.

But first, let's talk about the *other* way that a quantum state can evolve: through *unitary evolution*.

Let $U : \mathbb{C}^2 \to \mathbb{C}^2$ be some unitary transformation. Given a potentially unknown state $|\psi\rangle$, it is possible to "apply U to $|\psi\rangle$". This will result in the state $U(|\psi\rangle)$ (later to be denoted as $U|\psi\rangle$).

As a sanity check: do unitaries map valid quantum states to other valid quantum states? In other words, is $U|\psi\rangle$ norm 1? YES! Recall from homework 1 that unitary transformations preserve inner products. This means

$$\|Uv\|^2 = \langle Uv, Uv\rangle = \langle v, v\rangle = \|v\|^2$$

In other words, unitaries preserve norm as well. Let's look at some example:

Example 4.1. The identity map I, defined as $I|\psi\rangle = |\psi\rangle$ for all $|\psi\rangle$.

The "Pauli X", or "quantum NOT gate", map X, defined as $|0\rangle \mapsto |1\rangle$ and $|1\rangle \mapsto |0\rangle$. In general,

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

The "Pauli Z" map Z, defined as $|0\rangle \mapsto |0\rangle$, $|1\rangle \mapsto -|1\rangle$. In general,

$$Z(\alpha|0\rangle + \beta|1\rangle) = \alpha Z|0\rangle + \beta Z|1\rangle = \alpha|0\rangle - \beta|1\rangle$$

The "Pauli Y" map Y, defined as $|0\rangle \mapsto i|1\rangle$, $|1\rangle \mapsto -i|0\rangle$.

Terminology: We'll often refer to unitaries as "quantum gates", analogous to "logical gates" from classical computation.

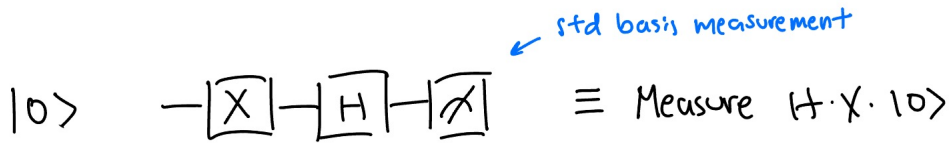
So far, the unitaries we saw maps standard basis states to standard basis states. Here's a more interesting example.

Example 4.2. The "Hadamard gate" H:

$$|0\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = |+\rangle \qquad |1\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = |-\rangle$$

*****

Now that we have gates defined, let's talk about drawing a quantum circuit.

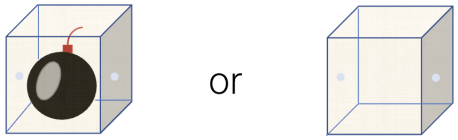


This is typically how a quantum circuit is drawn. In this specific case, we have

$$|0\rangle \overset{X}{\mapsto} |1\rangle \overset{H}{\mapsto} \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \overset{obs}{\mapsto} \begin{cases} \text{“0” with probability } \frac{1}{2} \\ \text{“1” with probability } \frac{1}{2} \end{cases}$$

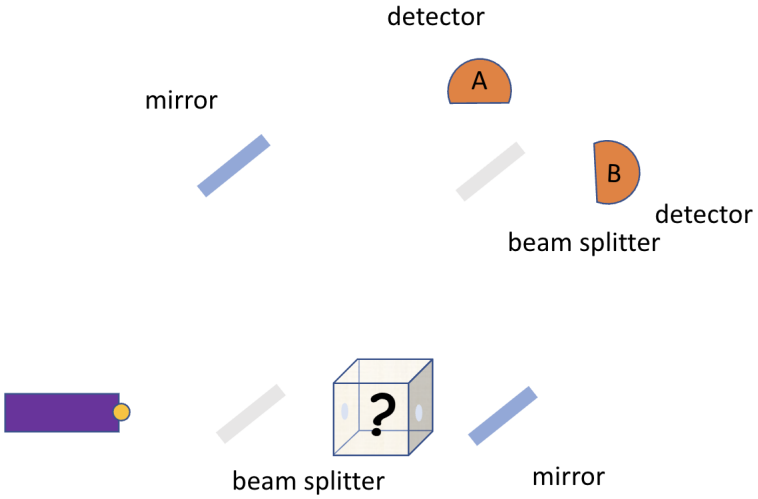
4.2 Elitzur-Vaidman tester

We start with a thought experiment. There may or may not be a bomb inside a box. There are holes on either side where a photon can pass through.



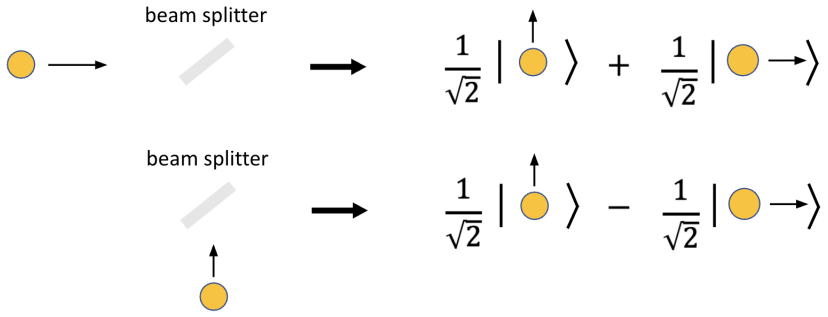
However, here's the catch. If we pass a photon through the box and the bomb is there, we get blown up. The question is: can we determine if the bomb exists without having the possibliity of getting blown to smithereens?

Not classically. But consider the following quantum set-up.



To interpret this, the photon is now a quantum state, and the beam splitters as quantum gates. Here, it's not that the beam splitter is "reflecting with prob. 1/2 and letting through with prob. 1/2", but rather, it's spliting the *quantum state* into a superposition of two.

Perhaps we can formalize this a bit (note that none of these diagrams are mine, they're copied from Andrea's lecture slides):

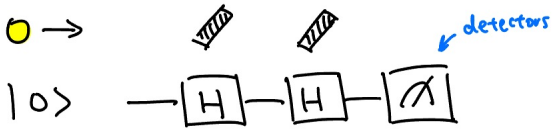


Now doesn't this look like a hadamard gate! Remember that $|0\rangle$ and $|1\rangle$ are just labels we put onto different orthogonal quantum states of a qubit. In this case, we can label "forward travel" as $|0\rangle$ and "upward travel" as $|1\rangle$. Then, rewriting a few things lets us see that

$$|0\rangle \xrightarrow{\text{beam splitter}} \frac{1}{\sqrt{2}}|1\rangle + \frac{1}{\sqrt{2}}|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = |+\rangle$$
$$|1\rangle \xrightarrow{\text{beam splitter}} \frac{1}{\sqrt{2}}|1\rangle - \frac{1}{\sqrt{2}}|0\rangle = -\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = -|-\rangle = |-\rangle$$

Note the "global phase" at the end of the second expression, allowing us to generalize this beam splitter as exactly the Hadamard gate! We now split the experiment into two cases:

Case 1: the box is empty. Using this knowledge, we can recreate the experiment set-up as the following quantum circuit:

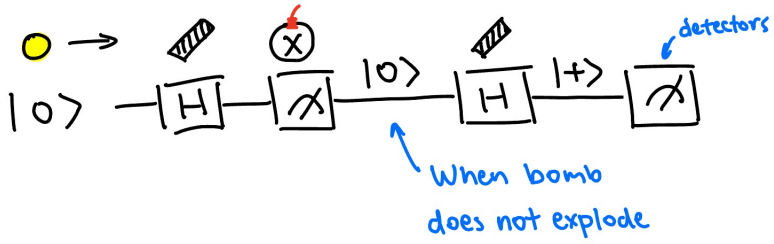


This lets us do the math:

$$\begin{aligned}
|0\rangle &\overset{H}{\mapsto} \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\
&\overset{H}{\mapsto} \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) + \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \\
&\mapsto \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|0\rangle - \frac{1}{2}|1\rangle \\
&= |0\rangle
\end{aligned}$$

We should notice that due to the superposition of the photon, regardless of the beam splitters themselves, the photon will *always* end up traveling "forward" and being detected by detector B, given that the box is empty.

Case 2: the box contains the bomb. Here's where the magic happens. We once again recreate the experiment set-up as a quantum circuit:



This gives us a very interesting phenomenon:

1. When the photon passes through the first Hadamard gate, it becomes a superposition of the "forward" and "upward" states.
2. Upon observation of the photon at the bomb, the qubit collapses to the "forward" state with probability $\frac{1}{2}$, which would lead to the bomb exploding.
3. *However*, if the qubit collapses to the "upward" state, then the bomb will not explode, and the photon continues to travel until it reaches the second Hadamard gate.
4. From here, since the state of the photon isn't in superposition, there will be no "interference" happening, and the second Hadamard gate will once again split the photon into a superposition of the "forward" and "upward" states
5. Finally, upon observation at the detectors, the photon will have a 1/2 probability of being detected by either detector.

Here's the catch: since we previously concluded that "empty box implies detected by B", now if the photon is detected by A, we will know *for certain* that there exists a bomb in the box, without triggering the explosion at all.

To conclude case 2 and the overall experiment,

$$\text{If the box contains the bomb: } \rightarrow \begin{cases} \text{BOOM with probability } \frac{1}{2} \\ \textbf{detected by A with probability } \frac{1}{4} \\ \text{detected by B with probability } \frac{1}{4} \end{cases}$$

4.3 Quantum key distribution

Cryptography!!! We now talk about quantum key distribution. Here's the scenario: there are two parties alice and bob, and the goal is to agree on a uniformly random secret key.

But before we get to that actually, suppose both bob and alice already has some agreed-upon secret key. Given this, they are able of achieving "unconditional" security – the given ciphertext will appear uniformly random to some adversary eve, *regardless* of how much computing power eve has.

But here's the catch: the process of key-distribution itself is *not possible* with unconditional security... classically.

dun dun dunnnnnn! to be continued (we ran out of time in lecture).

5 Lecture 05: Apr. 14th

Quantum key distribution (QKD), BB84 protocol. States of many qubits, uniform superposition, EPR pair, generalized Born's rule for n qubit states. Composition of quantum systems, norm consistency.

5.1 Quantum key distribution (cont'd)

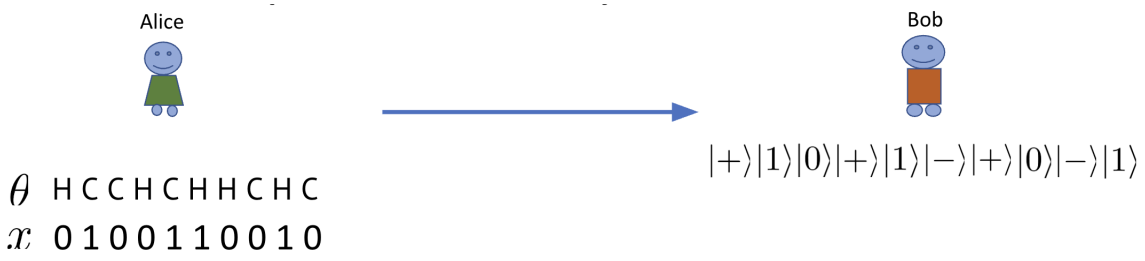
We began lecture with a recap and continuation of quantum key distribution – a phenomenon posible with only a single qubit state. Recall that if alice and bob both agree on some secret key, then information theoretically, they can achieve "unconditional security".

But unfortunately, it's not possible to distribute keys with the same level of unconditional security in the classical world, since eve can just read the public communication. Let's see if quantum computation can achieve this.

Imagine now alice sends *quantum states* to bob, namely, uniformly one at a time from $|0\rangle, |1\rangle, |+\rangle, |-\rangle$. Now the only way for eve to extract information is to make a measurement in some basis.

Upon measurement, the state will collapse, meaning bob must receive the same state that eve had observed. But there is a caveat: if eve happens to measure in the wrong basis, she disturbs the state, which means alice and bob can detect eavesdropping. This notion of possible detection inspired [Bennett, Brassad '84] to conclude that key distribution with unconditional security *is* possible quantumly.

To begin, alice first samples some string θ that represents which basis each qubit of the string should be measured in. She then *only sends* the qubit string x to bob.



Bob receives x' and now samples some θ' uniformly randomly. Note that x' should agree with x at all locations where $\theta' = \theta$, assuming there is no adversary making measurements along the way.

But that's a bad assumption to make, right? Eve could have measured x bit by bit, and have disturbed a subset of the states of x'. We now need a way for alice and bob to "double check" the consistency of their bit strings.

To do this, alice and bob share θ and θ' with each other. They then individually compare θ and θ', and discard all locations where $\theta_i \neq \theta'_i$, since the measurementst done in those

locations will be inconsistent.

From there, alice picks a random subset of locations where $\theta_i = \theta_i'$ and sends these indices to bob. Bob then sends back the result of the measurement in those indices.

Lastly, alice checks if bob's measurement in those indices are consistent with her own. If they are, then alice and bob will use the remaining shared locations as the key.

We didn't have time to get super deep into this, but intuitively, this is secure because eve wouldn't know which qubits are going to be involved in the consistency check. Think about it this way:

1. Suppose for a single qubit, alice and bob choose the same basis (say the standard basis, where alice sends 0).
2. Eve doesn't know this basis, so she randomly pick a basis. If she guessed right, she'd measure and get 0, then resend $|0\rangle$ to bob.

   However, if she guessed incorrectly and measures in Hadamard, she force the definitive $|0\rangle$ that alice sent into superposition: either $|+\rangle$ or $|-\rangle$.
3. Suppose eve guessed incorrectly, then bob now has a 50% chance of measuring 0, and a 50% chance of measuring 1.

Putting it all together, there is a $1/2$ chance eve picks the wrong basis, and of that half, there is another $1/2$ chance bob gets the wrong measurement. This means $1/2 \cdot 1/2 = 1/4$ of the time, bob's measurement will be inconsistent to alice's original qubit.

This is where consistency checks come in: if alice and bob happen to check this bit, then eve has a $3/4$ chance of getting away with it.

However, if eve gets greedy and measures many qubits (say 10 qubits) to try to recover the key, and if those 10 qubits are apart of the consistency check between alice and bob, then eve's chance of survival drops to $(3/4)^{10} \approx 0.056$. This exponentially decays as more qubits are measured.

5.2 States of many qubits

Now we begin talking about systems involving many qubits, and for most applications beyond this point, we will need these many-qubit states.

So far, single-qubits are defined $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, which is a superposition of the classical "0" and "1". We now define an n-qubit state as a superposition of n-bit strings.

$$\sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

Formally, the state of n qubits is a unit vector in $\mathbb{C}^{2^n}$. We denote the standard basis as

$$\{|x\rangle : x \in \{0,1\}^n\}$$

where $|x\rangle$ represents the i-th standard basis vector if x is the i-th string in lexicographic order. And just like a single qubit, we can represent the $|x\rangle$'s as representing the classical states.

Then, the general state of n qubits is

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$$

for some $\alpha_x \in \mathbb{C}$, where $\sum_x |\alpha_x|^2 = 1$.

Just to put this into scale, a 500-bit string is described fully with... 500 bits. Duh. But, a 500-qubit state takes 2^{500} complex numbers to describe! This is all a lot to wrap our heads around.

Example 5.1. Two qubits: $\mathbb{C}^4$. The standard basis states are $|00\rangle, |10\rangle, |01\rangle, |11\rangle$. In general,

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

There is a special example of this called the **EPR pair**, where $|\Psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. We will see that this is quite special.

There is a generalized Born's rule now, for n state qubits.

Theorem 5.1 (Born's rule, generalized).
If we measure $|\psi\rangle = \sum_x \alpha_x|x\rangle \in \mathbb{C}^{2^n}$ in the standard basis, we get outcome "x" with probability $|\alpha_x|^2$.

More generally, if we measure $|\psi\rangle$ in the orthonormal basis $\{|b_1\rangle, \ldots, |b_{2^n}\rangle\}$, we get outcome "$b_i$" with probability $|\langle b_i|\psi\rangle|^2$.

The state after obtaining "b_i" is $|b_i\rangle$.

5.3 Composing quantum systems

Let $|\psi_1\rangle, |\psi_2\rangle$ be single-qubit states. What is the two-qubit state corresponding to "the first qubit in state $|\psi_1\rangle$, and second qubit in state $|\psi_2\rangle$"? In other words,

$$|\psi_1\rangle|\psi_2\rangle \xrightarrow{?} \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

Theorem 5.2 (Composition of quantum systems).
Suppose $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$. Then, we write $|\psi_1\rangle|\psi_2\rangle$ to denote the two-qubit state

$$\alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \alpha_2\beta_1|10\rangle + \beta_1\beta_2|11\rangle$$

Claim. This will always be norm 1.

Proof.

$$\begin{aligned}
\||\psi_1\rangle|\psi_2\rangle\|^2 &= \|\alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \alpha_2\beta_1|10\rangle + \beta_1\beta_2|11\rangle\|^2 \\
&= |\alpha_1 \cdot \alpha_2|^2 + |\alpha_1\beta_2|^2 + |\beta_1 \cdot \alpha_2|^2 + |\beta_1 \cdot \beta_2|^2 \\
&= |\alpha_1|^2 \cdot (|\alpha_2|^2 + |\beta_2|^2) + |\beta_1|^2 \cdot (|\alpha_2|^2 + |\beta_2|^2) \\
&= |\alpha_1|^2 + |\beta_1|^2 \\
&= 1
\end{aligned}$$

□

6 Lecture 06: Apr. 16th

Linearity of composition, tensor product of quantum states, entanglement, product states. Unitary evolution, unitary transformations, tensor product of unitaries, tensor exponentiation, uniform superposition, multi-qubit gates, CNOT gate, CZ gate. Born's rule for partial measurements, re-normalization.

6.1 Composing quantum systems (cont'd)

We began with a quick review of the composition of two quantum states. A quick remark:

Remark. Composition satisfies "linearity": for all $a, b, c, d \in \mathbb{C}$, and all $|\psi_1\rangle, |\psi_2\rangle, |\phi_1\rangle, |\phi_2\rangle$ single-qubit states, we have

$$(a|\psi_1\rangle + b|\psi_2\rangle)(c|\phi_1\rangle + d|\phi_2\rangle) = ac|\psi_1\rangle|\phi_1\rangle + ad|\psi_1\rangle|\phi_2\rangle + bc|\psi_2\rangle|\phi_1\rangle + bd|\psi_2\rangle|\phi_2\rangle$$

Example 6.1. Let $|\psi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and $|\psi_2\rangle = |1\rangle$.

What is $|\psi_1\rangle|\psi_2\rangle$? It's $\frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|11\rangle$.

Sometimes, we will also see the notation $|\psi_1\rangle \otimes |\psi_2\rangle$, called "tensor product" of two quantum states.

Remark. $|\psi_1\rangle|\psi_2\rangle$ is very different from $\langle\psi_1|\psi_2\rangle$.

What about the converse of the previous theorem? Are all two-qubit states of the form $|\psi_1\rangle|\psi_2\rangle$? NO! Consider the EPR pair $|\Phi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. It cannot be written as $|\psi_1\rangle|\psi_2\rangle$ for any $|\psi_1\rangle, |\psi_2\rangle$.

This leads us to the concept of entanglement.

Definition (Entanglement)**.** A two-qubit state $|\phi\rangle$ is **entangled** if it *cannot* be written in the form $|\psi_1\rangle|\psi_2\rangle$.

States that are not entangled are called "product" states.

As a quick fun fact, almost all states are entangled. There are 2^n parameters needed to describe an n-qubit states, and really only $2n$ parameters needed to describe product states.

The composition of multiple qubit states can be generalized beyond just two qubits:

Let $|\psi_1\rangle, \ldots, |\psi_n\rangle$ where $|\psi_i\rangle = \alpha_0^i|0\rangle + \alpha_1^i|1\rangle$. Then,

$$|\psi_1\rangle|\psi_2\rangle \cdots |\psi_n\rangle = \sum_{x \in \{0,1\}^n} \beta_x |x\rangle$$

where $\beta_x = \alpha_{x_1}^1 \cdot \alpha_{x_2}^2 \cdots \alpha_{x_n}^n$.

6.2 Unitary evolution of multi-qubit states

States of many qubits also evolve in one of two ways:

1. Measurement
2. Unitary transformation

We will focus on unitary evolutions in this subsection. Let $U : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ be a unitary transformation. Given some (unkonwn) state $|\psi\rangle \in \mathbb{C}^{2^n}$, we can "apply" U to $|\psi\rangle$. This results in the state $U|\psi\rangle$.

BUT! What if we want to apply X to the first qubit of a state, and Z to the second qubit? This brings us to define

Theorem 6.1 (Tensor product of unitaries).
For single qubit gates A and B, we write $A \otimes B$ to denote a unitary such that

$$A \otimes B|\psi_1\rangle|\psi_2\rangle = (A|\psi_1\rangle) \otimes (B|\psi_2\rangle)$$

for all single-qubit states $|\psi_1\rangle, |\psi_2\rangle$.

A very fair question: is this definition sufficient to specify $A \otimes B$ entirely?

Answer: Yes. This specifies how $A \otimes B$ acts on $|0\rangle|0\rangle = |00\rangle, |01\rangle, |10\rangle, |11\rangle$, ie on the standard basis.

Example 6.2. Find the following tensor product of unitaries:

$$X \otimes I|01\rangle = X|0\rangle \otimes I|1\rangle = |1\rangle|1\rangle = |11\rangle$$

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

Some notation: if we "exponentiate A" using tensor product n times, then

$$\underbrace{A \otimes A \otimes \cdots \otimes A}_{n \text{ times}} \equiv A^{\otimes n}$$

$$\underbrace{|\psi\rangle \otimes |\psi\rangle \otimes \cdots \otimes |\psi\rangle}_{n \text{ times}} \equiv |\psi\rangle^{\otimes n}$$

Example 6.3. What is $H^{\otimes n}|0\rangle^{\otimes n}$?

$$\begin{aligned} H^{\otimes n}|0\rangle^{\otimes n} &= |+\rangle^{\otimes n} \\ &= \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \\ &= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \end{aligned}$$

Notice that this is still the uniform superposition!

The natural question that will sound very familiar is: are *all* two-qubit gates of the form $A \otimes B$? NO! Same counting argument from above.

Let's now go over some examples of canonical multi-qubit gates:

Example 6.4. We look at multi-qubit gates:

- `CNOT` gate ("Controlled NOT"): flips the second qubit only if the first qubit is 1.

$$\begin{aligned} \texttt{CNOT}|00\rangle &= |00\rangle \\ \texttt{CNOT}|01\rangle &= |01\rangle \\ \texttt{CNOT}|10\rangle &= |11\rangle \\ \texttt{CNOT}|11\rangle &= |10\rangle \end{aligned}$$

- CZ gate ("Controlled Z"): applies the Z gate to the second qubit only if the first qubit is 1.

$$\begin{aligned} CZ|00\rangle &= |00\rangle \\ CZ|01\rangle &= |01\rangle \\ CZ|10\rangle &= |10\rangle \\ CZ|11\rangle &= -|11\rangle \end{aligned}$$

6.3 Partial measurements

Another question that may naturally arise is: Is it possible to measure just one qubit of a state of n-qubits? YES!

Theorem 6.2 (Born's rule for partial measurements ($n = 2$)).
Let $|\psi\rangle$ be a two-qubit state

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

If we measure the first qubit of $|\psi\rangle$ in the standard basis, we get

- *Outcome "0" with probability* $|\alpha_{00}|^2 + |\alpha_{01}|^2$,
  and the post-measurement state $\dfrac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$
- *Outcome "1" with probability* $|\alpha_{10}|^2 + |\alpha_{11}|^2$,
  and the post-measurement state $\dfrac{\alpha_{10}|10\rangle + \alpha_{11}|11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}$

Example 6.5. What if we measure the first qubit of $\frac{1}{\sqrt{3}}|00\rangle + \frac{1}{\sqrt{3}}|01\rangle + \frac{1}{\sqrt{3}}|11\rangle$?

We get outcome "0" with probability $\frac{1}{3} + \frac{1}{3} = \frac{2}{3}$. The post measurement state collapses to

$$\frac{\frac{1}{\sqrt{3}}(|00\rangle + |01\rangle)}{\sqrt{\frac{2}{3}}} = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle$$

And outcome "1" with probability $\frac{1}{3}$. The post measurement state collapses to $|11\rangle$.

7 Lecture 07: Apr. 21st

Full measurement vs partial measurement, Born's rule for partial measurements in arbitrary bases. Non-local games, magic square game, genuine quantumness, non-determinism, true randomness.

7.1 Partial measurements (cont'd)

Recall from last week that we've just seen Born's rule for partial measurements, which is that *kinda* ugly thing where we have

- Outcome "0" with probability $|\alpha_{00}|^2 + |\alpha_{01}|^2$,

  and the post-measurement state $\dfrac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$
- Outcome "1" with probability $|\alpha_{10}|^2 + |\alpha_{11}|^2$,

  and the post-measurement state $\dfrac{\alpha_{10}|10\rangle + \alpha_{11}|11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}$

Picking from the example that we eneded lecture with, we saw that

Example 7.1. When we measure the first qubit of $\frac{1}{\sqrt{3}}|00\rangle + \frac{1}{\sqrt{3}}|01\rangle + \frac{1}{\sqrt{3}}|11\rangle$, we get outcome "0" with probability $\frac{1}{3} + \frac{1}{3} = \frac{2}{3}$. The post measurement state collapses to

$$\frac{\frac{1}{\sqrt{3}}(|00\rangle + |01\rangle)}{\sqrt{\frac{2}{3}}} = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle$$

And outcome "1" with probability $\frac{1}{3}$. The post measurement state collapses to $|11\rangle$.

What happens now if we measure the second qubit? Well, it would be conditioned on the first measurement, meaning

- If the first outcome was "0", we get "0" with prob 1/2, and "1" with prob 1/2
- If the first outcome was "1", we get "1" again with probability 1.

Putting everything together, we'd get

- "00" with probability $\frac{2}{3} \cdot \frac{1}{2} = \frac{1}{3}$
- "01" with probability $\frac{2}{3} \cdot \frac{1}{2} = \frac{1}{3}$
- "11" with probability $\frac{1}{3} \cdot 1 = \frac{1}{3}$

Remark. The distribution over all outcomes is exactly the same as if we had just made a *full* standard basis measurement on $|\psi\rangle$.

It turns out, it is of course entirely possible to make partial measurements in an arbitrary basis.

Theorem 7.1 (Born's rule for partial measurements in an arbitrary basis). *Let $\mathcal{B} = \{|b_0\rangle, |b_1\rangle\}$ be some orthonormal basis. Let $|\psi\rangle = \alpha|b_0\rangle|\phi_0\rangle + \beta|b_1\rangle|\phi_1\rangle$ If we measure the first qubit of $|\psi\rangle$ in basis $\mathcal{B}$, we get*

- *"0" with probability $|\alpha|^2$, giving us $|b_0\rangle|\phi_0\rangle$*
- *"1" with probability $|\beta|^2$, giving us $|b_1\rangle|\phi_1\rangle$*

Remark. Measuring the first qubit of $|\psi_1\rangle|\psi_2\rangle$ in some basis $\mathcal{B}$ is equivalent to:

1. Measuring $|\psi_1\rangle$ in $\mathcal{B}$
2. Composing the post-measurement state with $|\psi_2\rangle$

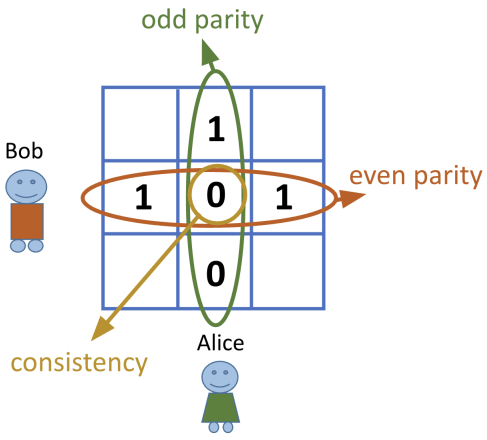
7.2 Non-local games

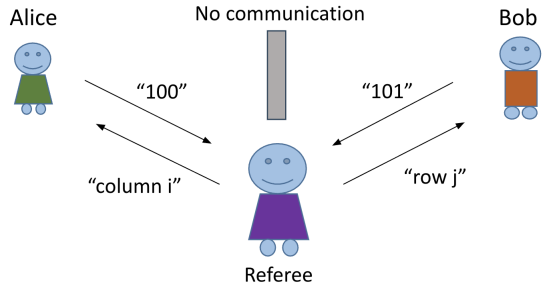
We start a new topic! This is the first application of entanglement and multiple-qubit states. A non-local game involves a referee, and two players Alice and Bob, both of whom are trying to win the game *cooperatively*, but have no way of communication with each other.

One example is the "magic square game". In this game, the referee will send a number $i \in \{1, 2, 3\}$ to alice, representing the column of a 3×3 square. Alice is then expected to send back 3 bits, which will fill the column top-down. Additionally, the sum of the column must be *odd*.

Similarly, the referee sends a number $j \in \{1, 2, 3\}$ to bob, representing a row. Bob is also expected to send back 3 bits, which will fill the row left-right. Additionally, the sum of the row must be *even*.

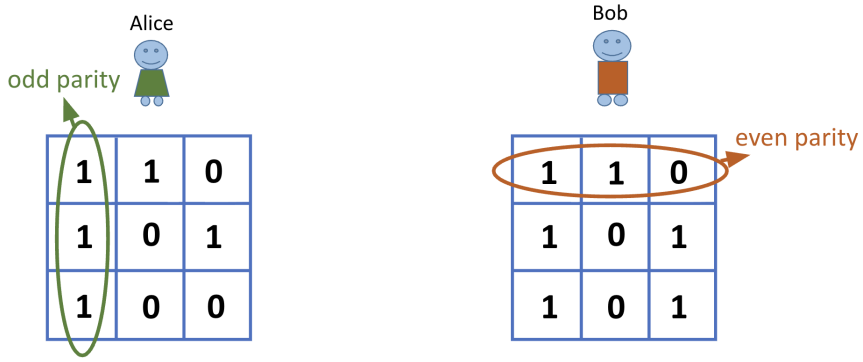
In order to win, both the column and row parity conditions must be met. In addition, the overlapping square must be *consistent*.





What would be the optimal strategy here for alice and bob? Remember that they're free to communicate prior to the game starting.

One strategy could be to establish their own 3×3 squares, satisfying their respective parity constraints, while making sure as many squares are consistent as possible. This strategy will win with probability $\frac{8}{9}$.



Now, does there exists a "perfect strategy"? A consistent 3×3 square that satisfies *all* of the parity constraints, eliminating that [3][3] inconsistency?

No. There are no magic squares.

Proof. (sketch)

If consistent across columns, then all 3 columns have odd sum, resulting in an overall odd sum of the 3×3 square.

If consistent across rows, then all 3 rows have even sum, resulting in an overall even sum of the 3×3 square. A contradiction! □

So the best strategy achieves a winning probability of $\frac{8}{9}$... in the classical world.

It turns out, as we've all expected, with quantum resources Alice and bob can win with probability 1. To do this, prior to the game starting, they created an EPR pair $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, where alice takes one qubit, and bob takes the other.

The basic idea is that, alice will make a measurement to her qubit in some basis dependent on the input from the referee. We won't dive into the strategy just yet, but there are a few consequences of such a strategy existing.

First, this game serves as a "test" for quantumness. To do this, imagine two devices playing this MAGIC SQUARE game. To ensure the two boxes don't communicate, we can place them very very far away, such that it would defy the speed of light if they did communicate.

In this case, observing a winning probability greater than $\frac{8}{9}$ certifies the presence of **genuine quantumness**.

Second, this game serves as "proof" of non-determinism. We boutta get philosophical. Is nature deterministic? If we fixed all initial conditions of the universe, would everything happening afterwards also be deterministic?

As in, if physical systems evolve according to the laws of physics, and if we fix the state of every particle in a system, then theoretically everything that happens later can just be derived from a set of equations.

The "probabilities" we encounter in our everyday life is simply a proxy for our lack of information. Throwing a die? If we knew the velocity and momentum of every single particle in the die and the system, we should theoretically predict the outcome of the die with probability 1, not 1/6.

Similarly, in the magic square game played by two boxes, even if we had information about the entire state of the system (including both boxes playing the magic square game), any deterministic, predictable strategy *must* have winning strategy at most 8/9.

But wait. Since there exists a quantum strategy that succeeds with probability 1, this means the strategy cannot be deteriministic! The answers cannot possibly exist before the qusetions are asked, regardless of how much information is available. This peeks at the idea of non-determinism and true randomness existing within the environment.

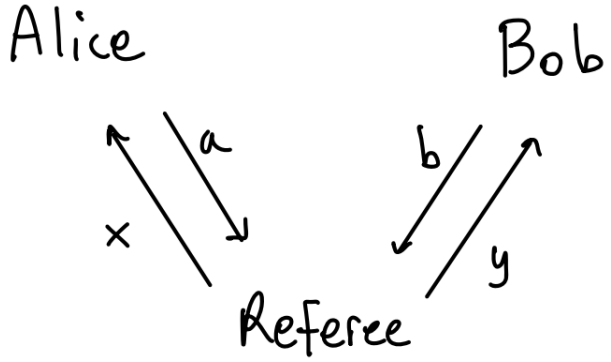
8 Lecture 08: Apr. 23rd

CHSH game, generalized non-local games, quantum strategies, CHSH optimality.

8.1 The CHSH game

Last lecture, we saw this remarkable non-local game called the "Magic square" game, for which there is a perfect quantum strategy that beats all classical strategies. Today, we'll look at a much simpler game, and actually dive into the quantum strategies involved.

Before that, let's clean up our notations a bit.



A non-local game (like the diagram above) is specified as:

- $\mathcal{X}$:= the set of possible questions for alice
- $\mathcal{Y}$:= the set of possible questions for bob
- $\mathcal{A}$:= the set of possible answers of alice
- $\mathcal{B}$:= the set of possible answers of bob
- A *winning condition:* a function of all inputs that is specified

$$V(x, y, a, b) = \begin{cases} 1 & \text{“win”} \\ 0 & \text{“lose”} \end{cases}$$

- A distribution over questions: a pair in $\mathcal{X} \times \mathcal{Y}$. This is usually the uniform distribution over $\mathcal{X} \times \mathcal{Y}$.

Example 8.1. For the magic square game, $\mathcal{X} = \mathcal{Y} = \{1, 2, 3\}$ (specifies a column / row), $\mathcal{A} = \mathcal{B} = \{0, 1\}^3$ (entries of a column / row).

The optimal classical winning probability is $w_c = \frac{8}{9}$. The optimal quantum winning probability is $w_q = 1$.

Now, it's time to actually introduce the **CHSH game**. In this game, we have that $\mathcal{X} = \mathcal{Y} = \mathcal{A} = \mathcal{B} = \{0, 1\}$. The winning condition in defined as $a \oplus b = x \cdot y$.

Notice that the RHS of the aforementioned winning condition is 0 unless $x = y = 1$. So we can expand and write the winning condition as

x	y	win condition
0	0	$a = b$
0	1	$a = b$
1	0	$a = b$
1	1	$a \neq b$

What is the optimal classical strategy? If alice and bob always answer 0, then $\Pr[\text{win}] = \frac{3}{4}$. Is this optimal?

Proof. Recall that random strategies do not help, so we restrict our proof to the set of all deterministic strategies. How many deterministic strategies are there?

For each $x \in \{0, 1\}$ denote alice's answer as a_x. Similarly, for each $y \in \{0, 1\}$ denote bob's answer as b_y. Then, the strategy is entirely specified by a_0, a_1, b_0, b_1, and there are two possible values for each, giving us a total of 16 strategies.

Now, factoring the winning conditions, we see that a strategy wins if and only if

- $a_0 \oplus b_0 = 0$
- $a_0 \oplus b_1 = 0$
- $a_1 \oplus b_0 = 0$
- $a_1 \oplus b_1 = 1$

But I claim that there are no solutions to this systems of equations. Notice

$$(a_0 \oplus b_0) \oplus (a_0 \oplus b_1) = 0 \qquad \Rightarrow \qquad b_0 \oplus b_1 = 0$$

and

$$(a_0 \oplus b_1) \oplus (a_1 \oplus b_1) = 1 \qquad \Rightarrow \qquad b_0 \oplus b_1 = 1$$

a contradiction! Thus, we conclude that alice and bob must fail at least 1 of the 4 possible pairs of questions, implying $\Pr[\text{“win”}] \leq \frac{3}{4}$, proving the desired claim. □

Alice and Bob can do better with a *quantum strategy.* But before that, what exactly *is* a quantum strategy?

A quantum strategy is specified by

- A state $|\psi\rangle_{AB}$ on some number $m_A + m_B$ qubits (think of it as alice holding the first m_A qubits, and bob holding the latter m_B qubits)

- Orthonormal bases as strategies, where

$$A_x, x \in \mathcal{X} \qquad \text{each } A_x \text{ is a basis of } m_A \text{ qubits}$$

$$B_y, y \in \mathcal{Y} \qquad \text{each } B_y \text{ is a basis of } m_B \text{ qubits}$$

The corresponding strategy then is:

1. Alice and bob share $|\psi\rangle_{AB}$
2. Alice, upon receiving question $x \in \mathcal{X}$, measures her m_A qubits in basis A_x and returns the outcome a as answers
3. Bob, upon receiving question $y \in \mathcal{Y}$, measures his m_B qubits in basis B_y and returns the outcome b as answers

8.2 Optimal quantum strategy for CHSH

In this strategy, define $|\psi\rangle_{AB} = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. How should we pick A_0, A_1, B_0, B_1?

Example 8.2. A_0, A_1, B_0, B_1 = standard basis. Here, Alice and Bob always return identical answers as a result of measing the EPR pair, giving us $\Pr[\text{“win”}] = \frac{3}{4}$.

So that doesn't work. Let's generalize: suppose Alice measures in basis $\{|s_0\rangle, |s_1\rangle\}$ and Bob measures in basis $\{|t_0\rangle, |t_1\rangle\}$. Suppose further that these bases only contain real-valued vectors. Then, what is $\Pr[a = b]$?

$$|\Phi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \overset{\text{HW3}}{=} \frac{1}{\sqrt{2}}|s_0\rangle|s_0\rangle + \frac{1}{\sqrt{2}}|s_1\rangle|s_1\rangle$$

From here, we have $\Pr[a = b] = \Pr[a = b = 0] + \Pr[a = b = 1]$. Notice that now we have

$$\Pr[a = b = 0] = \Pr[a = 0] \cdot \Pr[b = 0 \mid a = 0] = \frac{1}{2} \cdot \Pr[b = 0 \mid a = 0]$$

Upon Alice obtaining “0”, the state collapses to $|s_0\rangle|s_0\rangle$. From here, by born's rule, when Bob measures the second qubit in some arbitrary basis, he will get

$$\Pr[b = 0 \mid a = 0] = |\langle t_0|s_0\rangle|^2 = \||t_0\rangle\|^2 \cdot \||s_0\rangle\|^2 \cdot \cos^2(\theta) = \cos^2(\theta)$$

where θ is the angle between the two vectors $|t_0\rangle$ and $|s_0\rangle$. Substituting back into the original equation, we have that

$$\Pr[a = b = 0] = \frac{1}{2}\cos^2(\theta)$$

Similarly, $\Pr[a = b = 1] = \frac{1}{2}\cos^2(\theta')$, where θ' is the angle between $|s_1\rangle, |t_1\rangle$. But how are θ and θ' related?

Notice that $\theta' = \theta$ or $\theta' = \theta + \pi$ (neither case changes the value of cos), since $|s_1\rangle$ is orthogonal to $|s_0\rangle$, and $|t_1\rangle$ is orthogonal to $|t_0\rangle$. Plugging everything back now, we have

$$\Pr[a = b] = \cos^2(\theta)$$

Sanity check: if $\theta = 0$, alice and bob willl get $a = b$ with probability 1. We know then the nuance of this strategy is in **correctly picking** A_0, A_1, B_0, B_1.

We take

- $A_0 = \{|0\rangle, |1\rangle\}$. We call this "angle 0" (referring to the angle between the x-axis and the vectors)
- $A_1 = \{|+\rangle, |-\rangle\}$. We call this "angle $\pi/4$".

This is not intuitive at all, but it's the only two bases we know so let's just stick to it. How should we then pick B_0, B_1? Well, recall that we're trying to somehow play around with values of cos depedent on the angle between the bases.

Suppose $B_0 = \pi/8, B_1 = 3\pi/8$. Then, we have the angles between

- A_0 and B_0: $\pi/8$. Wins with probabiliy $\cos^2(\pi/8) \approx 0.85$.
- A_0 and B_1: $3\pi/8$. Wins with probabiliy $\cos^2(3\pi/8) \approx 0.15$.
- A_1 and B_0: $\pi/8$. Wins with probabiliy $\cos^2(\pi/8) \approx 0.85$.
- A_1 and B_1: $\pi/8$. Wins with probabiliy $1 - \cos^2(\pi/8) \approx 0.15$.

This gives us a overall winning proabibily of, like, $1/2$. But we are onto something I think. Let's isntead take $B_0 = \pi/8, B_1 = -\pi/8$. Then, we have the angles between

- A_0 and B_0: $\pi/8$. Wins with probabiliy $\cos^2(\pi/8) \approx 0.85$.
- A_0 and B_1: $-\pi/8$. Wins with probabiliy $\cos^2(-\pi/8) \approx 0.15$.
- A_1 and B_0: $\pi/8$. Wins with probabiliy $\cos^2(\pi/8) \approx 0.85$.
- A_1 and B_1: $3\pi/8$. Wins with probabiliy $1 - \cos^2(3\pi/8) = \sin^2(3\pi/8) = \cos^2(\pi/8) \approx 0.85$.

This gives us

$$\Pr[\text{"win"}] = \cos^2\left(\frac{\pi}{8}\right) \approx 0.85$$

and this clearly beats the classical winning probability of 0.75!

But wait a second, **where's this advantage coming from?** We could have chosen B_1 to be aligned with A_0. Then,

$$\begin{aligned} A_0 \text{ and } b_1 \approx 0 \quad &\Rightarrow \quad \text{better} \\ A_1 \text{ and } b_1 \approx \pi/4 \quad &\Rightarrow \quad \text{worse} \end{aligned}$$

Instead we chose $B_1 = \pi/8$. Is this trade-off worth it? Naively, we suppose no, because like the "average" angle is about the same.

However, crucially, probabilities are *squares* of amplitudes: they depend on $\cos^2(\theta)$, not $\cos(\theta)$. So increasing the angle between A_0 and B_0 from 0 to now $\pi/8$, we don't lose as much as we'd gain from increasing the angle between A_1 and B_1 from $\pi/4$ to now $3\pi/8$.

9 Lecture 09: Apr. 28th

Local hidden variable theory, incompleteness of quantum mechanics, quantum mechanics isn't LHV, true randomness, certifiable randomness. Quantum teleportation protocol, Bell basis.

9.1 CHSH (cont'd)

Seeing as our midterm is happening right now, the professor began with a contextualization of the fact that we're going to be moving onto the latter half of the course – with more quantum *computations*, involving quantum algorithms, or classical algorithms that get a speed-up quantumly.

Then, he gave us the bad news: quantum teleportation won't be as cool as it sounds ("one of the only things in this class that isn't as cool as it sounds" – andrea). But before we get to that, we first need to wrap up some perspectives on the CHSH game.

Classical Newtonian physics is a "local hidden variable" theory. **Hidden variable** means that there is an underlying variable λ which describes the exact properties of the entire system. This variable would allow us to erdict *any* future process with certainty. **Local** refers to actions taken at any point in space depending only on the information available at that point in space.

Is quantum mechanics "local"? Many would say no (non-local games exist, duh), but that is actually quite the misnomer. Quantum mechanics is "local" in the sense that the choice of basis to measure in at some point in space should only depend on information available at that point.

In a local hidden variable theory, Alice and Bob's strategy for CHSH would look like:

- Alice's answer is a function $a(x, \lambda)$
- Similarly, Bob's answer is a function $b(y, \lambda)$

But that's *just* a deterministic strategy, which implies $\Pr[\text{win}] \leq \frac{3}{4}$. We saw last time that there exists an quantum strategy such that $\Pr[\text{win}] = \cos^2(\pi/8) \approx 0.85$. This would mean that quantum mechanics is either not "local" or not "hidden variable".

Locality is super reasonable as we've discussed, which must imply quantum mechanics is not "hidden variable". This is actually quite the remarkable result!

This may not seem very surprising – after all, we started the course by learning **Born's rule**, which is dominated by probabilitiies. And it could have been that quantum mechanics, as we know it, is "incomplete". In the sense that there is a LHV theory that subsumes quantum mechanics. Einstein believed this, and he was quite unhappy with the probabilistic nature of quantum mechanics.

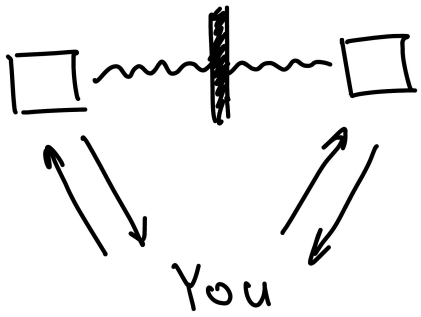
BUT! Einstein did not know about the CHSH game. The significance of this game, then, is that it serves as a "test" to "prove" that quantum mechanics isn't a LHV theory. In

fact, the first "loophole-free" experimentation of the quantum strategy winning with $> 3/4$ probability was performed in 2015! Nobel prize in physics a few years later.

One application of the CHSH game is the idea of "certifiable randomness". We discussed last time how CHSH game demonstrates *true randomness*, but even further than that, CHSH is a protocol that allows us to extract some randmness in a certifiable way (even without trusting the devices we are interacting with).

Here is a candidate protocol: we have a box, and we ask for it to be measuring $|+\rangle$ in the standard basis. It *could* be that when we measure, we get either $|0\rangle$ or $|1\rangle$ with probability $1/2$ each. But since we don't trust the box, it could also be that the box is serving us some pre-determined string, which is not "true" randomness.

Here's a high-level idea:



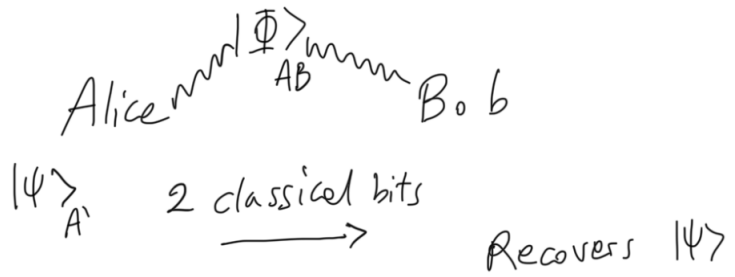
We have two boxes instead, and make them play the CHSH game many times. We then estimate the winning probability: if the probability > 0.75, then we have "true randomness" in Alice and Bob's answers. Because no classical randomness will be able to fool us with greater probability than $3/4$.

9.2 Quantum teleportation

We begin with the definition:

Definition (Quantum teleportation)**.** A protocol that allows Alice to transmit an arbitrary qubit state to Bob by only communicating classical bits.

The requirement: Alice and Bob share an EPR pair.



More importantly, Alice will no longer have $|\psi\rangle$ after bob recovers it. But before we get into how it works, let's first define some new stuff.

Definition (Bell basis)**.** This is an orthonormal basis of two qubits {

$$\begin{aligned}
&\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \quad \text{outcome "00"} \\
&\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle \quad \text{outcome "01"} \\
&\frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle \quad \text{outcome "10"} \\
&\frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle \quad \text{outcome "11"}
\end{aligned}$$

}. These are all orthonormal (trust me bro)

We can now get into the protocol:

Setup: alice and bob share an EPR pair $|\Phi\rangle_{AB}$ (one qubit each denoted A and B).

Input: $|\psi\rangle_{A'} \in \mathbb{C}^2$, making the overall state $|\psi\rangle_{A'}|\Phi\rangle_{AB}$.

Protocol:

1. Alice performs a Bell basis measurement on $A'A$. Let the outcome of the measurement be (a, b).
2. Alice sends (a, b) to bob.
3. Bob applies the gate $X^a Z^b$ to qubit B.

Claim: Bob's output is $|\psi\rangle$.

*****

Okay, this all seems a bit too magical. Let's justify it now. Recall born's rule for partial measurement. Let $|\phi\rangle$ be a 3-qubit state. Let $\mathcal{B} = \{|b_1\rangle, |b_2\rangle, |b_3\rangle, |b_4\rangle\}$ be an orthonormal basis of $\mathbb{C}^4$.

Question: What happens if we measure the first two qubits $|\phi\rangle$ in basis $\mathcal{B}$?

Answer: Write $|\phi\rangle$ expressing the first 2 qubits in basis $\mathcal{B}$.

$$|\phi\rangle = \alpha_1|b_1\rangle|\psi_1\rangle + \alpha_2|b_2\rangle|\psi_2\rangle + \alpha_3|b_3\rangle|\psi_3\rangle + \alpha_4|b_4\rangle|\psi_4\rangle$$

Then, we get

- Outcome "1" with probability $|\alpha_1|^2$, giving us outcome $|b_1\rangle|\psi_1\rangle$.
- Outcome "2" with probability $|\alpha_2|^2$, giving us outcome $|b_2\rangle|\psi_2\rangle$.
- Outcome "3" with probability $|\alpha_3|^2$, giving us outcome $|b_3\rangle|\psi_3\rangle$.
- Outcome "4" with probability $|\alpha_4|^2$, giving us outcome $|b_4\rangle|\psi_4\rangle$.

Putting things into context now, let $|\psi\rangle_{A'} = \alpha|0\rangle + \beta|1\rangle$. Then, the joint 3-qubit state:

$$\begin{aligned}
|\psi\rangle_{A'}|\Phi\rangle_{AB} &= (\alpha|0\rangle + \beta|1\rangle)(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle) \\
&= \frac{\alpha}{\sqrt{2}}|000\rangle + \frac{\alpha}{\sqrt{2}}|011\rangle + \frac{\beta}{\sqrt{2}}|100\rangle + \frac{\beta}{\sqrt{2}}|111\rangle \\
&= \text{... (being forced to do this in HW4 bruh)}
\end{aligned}$$

Once we expand via the magic of HW4, we claim that

$$\begin{aligned}
|\psi\rangle_{A'}|\Phi\rangle_{AB} = &\frac{1}{2}\left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle\right)_{A'A}|\psi\rangle_B \\
&+ \frac{1}{2}\left(\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle\right)_{A'A}(Z|\psi\rangle_B) \\
&+ \frac{1}{2}\left(\frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle\right)_{A'A}(X|\psi\rangle_B) \\
&+ \frac{1}{2}\left(\frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle\right)_{A'A}(ZX|\psi\rangle_B)
\end{aligned}$$

Notice that, no matter the outcome, Bob collapses to $|\psi\rangle, Z|\psi\rangle, X|\psi\rangle, ZX|\psi\rangle$.

10 Lecture 10: May 5th

Quantum computation, auxiliary qubits, quantum circuits, universal set of classical gates, exponential scaling of gates, universal set of quantum gates, Solovay-Kitaev theorem.

10.1 Basics of quantum computation

Welcome back! The past lecture(s) have been dedicated to a midterm review and midterm. We now begin the second half of the quarter, where we focus on quantum *computation* rather than *information*.

So what *is* quantum computation? It's typically outlined by the following steps:

1. Input: $|x_1\rangle \cdots |x_n\rangle |0\rangle \cdots |0\rangle$, where $x \in \{0,1\}^n$ is the classical input to the problem. The $|0\rangle$ tagged at the end are used as auxiliary qubits.
2. Apply a sequence of 1 and 2-qubit states to the input
3. Measure a subset of qubits (this gives a classical output)

To make this a bit more concrete, let $f : \{0,1\}^n \to \{0,1\}^m$ be a function thta we wish to "compute" (for example, input a n-bit string and output a 0 if prime and 1 if not).

We say that a *classical circuit* C computes f if: for all $x \in \{0,1\}^n$, $C(x) = f(x)$.

Conversely, we say that a **quantum circuit** C computes f if, for all $x \in \{0,1\}^n$,

$$\Pr[C(x) = f(x)] \geq 0.99$$

Since quantum measurements are entirely based on probabilistic outcomes. In fact, the 0.99 threshold is quite arbitrary – any constant over 1/2 should be sufficient, given that we just execute the circuit a bunch of times.

With quantum computation, we are interested in solving problems faster on quantum computers than classical computers, so the focus is on efficiency: how the number of gates scale with n.

*****

One natural question that arises is why we restrict circuits to only have 1 and 2-qubit gates? We've previously shown that n-qubit gates are very well defined! Well, this is because of hardware limitations. Generally speaking, the more qubits a gate acts on, the more difficult it is to implement.

Naturally, we may wonder if 1 and 2-qubit gates even sufficient to capture the full power of quantum computation, and the answer is, not surprisingly, of course they are. Let's take a look at this in the classical case:

Definition (Universal set of classical gates)**.** A set of classical gates S is universal if any function $f : \{0,1\}^n \to \{0,1\}^m$ can be computed by a circuit consisting of gates in S

Example 10.1. Some examples include {AND, OR, NOT}, or {NAND}.

But there is an issue with the universal set being too small. For *most* functions f, the number of (classical) gates a circuit needs to compute f ends up scaling *exponentially* with n.

How about the quantum case?

Definition (Universal set of quantum gates)**.** A set of quantum gates S is universal if, for all n, for all unitaries U on n qubits, for any precision $\epsilon > 0$, there is a quanutm circuit consisting of finitely many gates from S that "approximates" U to precision ϵ, i.e., there exists circuit C such that for any $|\psi\rangle$,

$$\|C|\psi\rangle - U|\psi\rangle\| \leq \epsilon$$

But why is this a good notion of approximation? We'll see in HW4 that if $\||\phi_1\rangle - |\phi_2\rangle\| \leq \epsilon$, then for any choice of measurement,

$$|\Pr[\text{“0”} \mid \phi_1] - \Pr[\text{“0”} \mid \phi_2]| \leq 2\epsilon$$

So distinguishing between two qubits that are close, is for all intents and purposes, the same.

Now similar to the classical cases, there indeed are small universal quantum gate sets.

Example 10.2. $\{H, \text{CNOT}, T\}$ is universal, where $T : |0\rangle \to |0\rangle$, and $T : |1\rangle \to e^{i\pi/4}|1\rangle$.

And again much like the classical case, most unitaries require $\exp(n)$ gates to approximate. Given this, we are especially interested in the subset of $\text{poly}(n)$-sized quantum circuits that compute functions that (we think) require $\exp(n)$ size circuits classically.

But there is another scaling factor here. Since we introduced this precision factor ϵ, how does the size of the circuits scale with n and ϵ? For example, if the scaling is $2^{1/\epsilon}$... that would be pretty bad.

Theorem 10.1 (Solovay-Kitaev Theorem (informal)).
Let S be any universal quantum gate set. Any n-qubit unitary can be approximated to any precision ϵ using

$$O\left(4^n \cdot \log^c\left(\frac{1}{\epsilon}\right)\right)$$

where c is a constant.

11 Lecture 11: May 7th

Invertibility of unitaries, reversible implementation of classical functions, bounding number of gates, uncomputing garbage. Deutsch's algorithm, constant vs balanced, black-box access.

11.1 Reversible computation

We began with a quick recap of quantum computation. The next natural question is: does quantum computation subsume classical comuptation? That is, can we at least quantumly compute everything that's able to be computed classically?

We start with a canonical example: how does a quantum computer even compute an AND gate? To discuss this, we first have to point out that the AND gate is *not* invertible, since it maps 2 bits to 1. But *all* unitaries are invertible! So how can we implement an AND gate only using unitaries?

Let's think about a "reversible" way of implementing an AND gate – one that takes 3 bits to 3 bits. For example,

$$x_1x_2b \mapsto x_1x_2\ b \oplus \text{AND}(x_1, x_2)$$

This is a pretty good canonical construction. Let's generalize this a bit. For any function $f : \{0,1\}^n \to \{0,1\}^m$, it can be implemented in a *reversible* way, mapping $n+m$ bits to $n+m$ bits:

$$x_1 \cdots x_n y \mapsto x_1 \cdots x_n y \oplus f(x_1, \ldots, x_n)$$

Now to bring everything back to the quantum world, let's consider the linear transformation U_f on $n+m$ qubits defined as

$$|x_1\rangle \cdots |x_n\rangle|y\rangle \mapsto |x_1\rangle \cdots |x_n\rangle|y \oplus f(x_1, \ldots, x_n)\rangle$$

This begs the question: is U_f a unitary? Yes, because it just permutes the standard basis elements (each element being the input $n+1$ qubit state).

To bring it back to our original question: in principle, any classical gate can be implemented on a quantum computer, we just need a few extra qubits. But this leads to another question: what about efficiency? Specifically, if f is clasically computed using T gates, how many quantum gates are needed to implement U_f?

One possible implementation: we take each classical gate and convert it to its "reversible" version. We then implement each reversible classical gate using some unitary built from the universal set $S = (H, CNOT, T)$. From here, using the theorem discussed previously, we would need $O\left(T \log^c\left(\frac{T}{\epsilon}\right)\right)$ gates.

Let's formalize this implementation. Let $C = G_T \cdots G_1$ be a classical circuit computing

$$f : \{0,1\}^n \to \{0,1\}^m$$

To obtain a quantum circuit computing f to some desired precision $\epsilon > 0$ using some universal set of gates S, do the following:

1. For each $i \in \{1, \dots, T\}$, let $\widetilde{G}_i$ be the unitary that implements G_i reversibly:

$$\underbrace{|x\rangle}_{k_i}\underbrace{|y\rangle}_{\ell_i} \longmapsto \underbrace{|x\rangle}_{k_i}\underbrace{|y \oplus G_i(x)\rangle}_{\ell_i}$$

2. Now consider the quantum circuit

$$\widetilde{C} = \widetilde{G}_T \cdots \widetilde{G}_1$$

   acting on $n + L$ qubits, where

$$L = \ell_1 + \ell_2 + \cdots + \ell_T$$

   (implement any "duplication" of wires using CNOT)

3. Implement each gate $\widetilde{G}_i$ using gates from the universal set S. It suffices for each implementation to be to precision ϵ/T (since error accumulates additively).

Overall, it implements the unitary

$$\underbrace{|x_1 \cdots x_n\rangle}_{n}\underbrace{|0 \cdots 0\rangle}_{L} \longmapsto \underbrace{|x_1 \cdots x_n\rangle}_{n}\underbrace{f(x_1, \dots, x_n)}_{m}\underbrace{|\cdots\rangle}_{L-m}$$

Couple things to note: (1) we can always apply SWAP gates to that the output of f comes first, and (2) the last $L-m$ qubits may contain some "garbage" $g(x_1, \dots, x_n)$, which includes intermediate values obtained during computation.

But ideally, we'd like the last $L-m$ qubits to be in the $|0 \cdots 0\rangle$ state (or a state independent of $x_1 \cdots x_n$). This is not just cosmetic, but rather crucial.

Fact. The "garbage" can always be easily "uncomputed", so we get quantum circuit acting as

$$|x_1 \cdots x_n\rangle|y\rangle|0 \cdots 0\rangle \longmapsto |x_1 \cdots x_n\rangle|y \oplus f(x_1, \dots, x_n)\rangle|0 \cdots 0\rangle$$

Because of this, going forward we omit the auxiliary qubits and write

$$|x_1 \cdots x_n\rangle|y\rangle \longmapsto |x_1 \cdots x_n\rangle|y \oplus f(x_1, \dots, x_n)\rangle$$

11.2 Deutsch's algorithm

We now move onto discussing our first quantum algorithm. It is one of the very first algorithms to provide evidence of the power of *quantum* algorithms over *classical*.

The problem. Determine if a given function $f : \{0, 1\} \to \{0, 1\}$ is constant or balanced. f is *constant* if $f(0) = f(1)$, otherwise, it's *balanced*.

The function f is given to us with "black-box access", that is, we get to query some x and see $f(x)$. Clasically, it is necessary and sufficient to query f twice to determine if it's constant or balanced. We claim that Deutsch's algorithm can do this with *one* query.

But let's back it up a bit: what does it mean to query a function as a black-box in the quantum setting? Naturally, we can define it as querying some $|x\rangle$ and get $|f(x)\rangle$ in return.

However, there is a big issue with this in that it's not guaranteed to be a unitary. Suppose f is constant, then it would map both $|0\rangle$ and $|1\rangle$ maps to $|0\rangle$, which is clearly not an invertible operation.

Luckily, we've just discussed this earlier. We can define f in a reversible way. By querying $|x\rangle|y\rangle$, we pass the query through some unitary U_f which gives us the output $|x\rangle|y \oplus f(x)\rangle$.

But now a natural worry is that is this a different kind of classical algorithm? By giving us two inputs, does this somehow make the quantum algorithm more powerful?

Remark. No. It's easy to see that a classical algorithm with 2 inputs x, y and outputs $x, y \oplus f(x)$ would still require 2 queries.

So then, what *does* quantum computation buy us? Consider the following, different procedures, where we query f in some superposition:

1. Put the first qubit in superposition:

$$\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes |0\rangle \xmapsto{U_f} \frac{1}{\sqrt{2}}U_f|00\rangle + \frac{1}{\sqrt{2}}U_f|10\rangle$$

   This is the same as saying

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle \xmapsto{U_f} \frac{1}{\sqrt{2}}|0\rangle|f(0)\rangle + \frac{1}{\sqrt{2}}|1\rangle|f(1)\rangle$$

   Crucially, notice now that the output contains information about *both* $f(0)$ and $f(1)$! The conversation is now about how we should "extract" this information. Because notice that, upon measurement of the first qubit, it would collapse to $|0\rangle|f(0)\rangle$ or $|1\rangle|f(1)\rangle$ with probability $1/2$ each...

2. Put the second qubit in superposition, rather than the first:

$$\begin{aligned}
|x\rangle|-\rangle = \frac{1}{\sqrt{2}}|x\rangle|0\rangle - \frac{1}{\sqrt{2}}|x\rangle|1\rangle &\xmapsto{U_f} \frac{1}{\sqrt{2}}U_f|x\rangle|0\rangle - \frac{1}{\sqrt{2}}U_f|x\rangle|1\rangle \\
&= \frac{1}{\sqrt{2}}|x\rangle|0 \oplus f(x)\rangle - \frac{1}{\sqrt{2}}|x\rangle|1 \oplus f(x)\rangle \\
&= |x\rangle\left(\frac{1}{\sqrt{2}}|0 \oplus f(x)\rangle - \frac{1}{\sqrt{2}}|1 \oplus f(x)\rangle\right) \\
&= (-1)^{f(x)}|x\rangle|-\rangle
\end{aligned}$$

It should be interesting to note that we get a global phase in terms of $f(x)$, although it's still just a global phase, which doesn't help us very much.

3. Naturally, the very next thing to try is to put *both* qubits in superposition. This gives us

$$\begin{aligned}|+\rangle|-\rangle &\xmapsto{U_f} \frac{1}{\sqrt{2}}(-1)^{f(0)}|0\rangle|-\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}|1\rangle|-\rangle \\ &= \left(\frac{1}{\sqrt{2}}(-1)^{f(0)}|0\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}|1\rangle\right) \otimes |-\rangle \\ &= \begin{cases} |+\rangle \text{ if } f(0) = f(1) \\ |-\rangle \text{ if } f(0) \neq f(1) \end{cases}\end{aligned}$$

up to a global phase.

Procedure 3 allows us to distinguish the two states perfectly!

12 Lecture 12: May 12th

Deutsch's algorithm. Simon's algorithm, exponential speed-up, period-finding on boolean functions, quantum subroutine + classical post-processing, uniform superposition.

12.1 Deutsch's algorithm (cont'd)

We started lecture with a quick recap of Deutsch's algorithm. Recall the problem: given black-box access to some function $f : \{0,1\} \to \{0,1\}$, determine if it's constant, i.e. $f(0) = f(1)$, or balanced, i.e. $f(0) \neq f(1)$.

Clasically, 2 queries to the black-box is necessary and sufficient for figuring out the structure of f. We argue that we can do this quantumly with only 1 query. Further recall that to implement f *reversibly*, we defined the black-box as a unitary U_f, where $|x\rangle|0\rangle \overset{U_f}{\longmapsto} |x\rangle|f(x)\rangle$.

We had several ideas in how to implement this:

1. Try putting the input qubit in superposition. As a result, both outputs are simultaneously computed and put in uniform superposition, but we don't have a way of extracting either one, since measuring the first qubit will cause the second to collapse.

$$|+\rangle|0\rangle \overset{U_f}{\longmapsto} \frac{1}{\sqrt{2}}|0\rangle|f(0)\rangle + \frac{1}{\sqrt{2}}|1\rangle|f(1)\rangle$$

   Thus, in order to win, we need the oracle to somehow evaluate $f(x)$ without leaving it in the aux qubit.

2. Try putting the aux qubit in superposition. Normally, we can think of the oracle as applying a NOT gate on the aux qubit whenever $f(x) = 1$, since $|0 \oplus f(x)\rangle = |1\rangle$. However, when we put the aux qubit in superposition, the output aux qubit will, interestingly, stay the same up to a global phase:

$$|x\rangle|-\rangle \overset{U_f}{\longmapsto} |x\rangle \left(\frac{1}{\sqrt{2}}|0 \oplus f(x)\rangle - \frac{1}{\sqrt{2}}|1 \oplus f(x)\rangle \right) = (-1)^{f(x)}|x\rangle|-\rangle$$

   Essentially, whenever $f(x) = 1$, a global phase of -1 will be applied. Although this still doesn't help us very much, since global phases don't change the probability distribution upon measurement...

3. Try putting *both* qubits in superposition. This is essentially using idea 2, but on both inputs at once. This gave us

$$|+\rangle|-\rangle \overset{U_f}{\longmapsto} \frac{1}{\sqrt{2}}(-1)^{f(0)}|0\rangle|-\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}|1\rangle|-\rangle = \begin{cases} |+\rangle|-\rangle & \text{if } f(0) = f(1) \\ |-\rangle|-\rangle & \text{if } f(0) \neq f(1) \end{cases}$$

   allowing for perfect distinguishing!

This worked because the oracle was able to simultaneously compute both inputs at the same time, so when $f(0) = f(1)$, the same phase would be applied to both inputs. However, when $f(0) \neq f(1)$, then the phase would only be applied to one of the inputs (whichever one $f(x) = 1$).

12.2 Simon's algorithm

We now move onto a new quantum algorithm! Simon's algorithm is a first example of a qunatum algorithm that solves a problem exponentially faster than the best classical randomized algorithm.

The Problem. You are given block-box access to a function $f : \{0,1\}^n \to \{0,1\}^n$ with the promise that there exists an $s \in \{0,1\}^n$ such that

$$f(x) = f(y) \qquad \text{if and only if } y = x \text{ or } y = x \oplus s$$

Find s.

Intuitively, we can think of s as a "shift", that f is "periodic" in a sense. For $s \neq 0^n$, this essentially means f is a 2-to-1 function, where s partitions the input space into pairs.

As usual, we'll think of this problem classically. How many queries to the black-box does it take to solve the problem (with certainty) classically? In the worst case, we'd need to query at least $2^n/2 + 1$ queries, since we could have happened to query exactly the half of the input space that all mapped to unique outputs.

Additionally, there are random algorithms that can succeed "with high probability" with $\Omega(\sqrt{2^n})$ queries.

Quantumly, this is possible with only $O(n)$ queries.

*****

As usual, we model the black-box as a $2n$-qubit unitary U_f acting as $|x\rangle|y\rangle \xmapsto{U_f} |x\rangle|y \oplus f(x)\rangle$. As we'll see in a second, Simon's algorithm consists of

1. A quantum sub-routine to sample from a useful distribution many times
2. Classical post-processing on the samples to recover the secret s

Quantum subroutine: queries U_f once and obtains a uniformly random $y \in \{0,1\}^n$ such that

$$y \cdot s = y_1 \cdot s_1 \oplus \cdots \oplus y_n \cdot c_n = 0$$

Classical post-processing: Receives $m = 100n$ samples $y^{(1)}, \ldots, y^{(m)}$ such that

$$y^{(1)} \cdot s = 0 \qquad \cdots \qquad y^{(m)} \cdot s = 0$$

With high probability, the system of linear equations has only solutions 0^n and s. Solve using Gaussian elimination (or other similar algorithms).

Let's start with the crux of the algorithm: the quantum subroutine.

1. Start with $|0^{2n}\rangle$
2. Apply $H^{\otimes n} \otimes I$
3. Apply U_f
4. Measure the second n qubits (in the standard basis)
5. Apply $H^{\otimes n} \otimes I$
6. Measure the first n qubits (in the standard basis)

To begin the analysis of this subroutine, we first recall that $H^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$. But what about $H^{\otimes n}|x\rangle$? We see that

$$\begin{aligned} H^{\otimes n}|x\rangle &= H|x_1\rangle \otimes H|x_2\rangle \otimes \cdots \otimes H|x_n\rangle \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{x_1}|1\rangle\right) \otimes \cdots \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{x_n}|1\rangle\right) \\ &= \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x_1 \cdot y_1 + x_2 \cdot y_2 + \cdots + x_n \cdot y_n} |y\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle \end{aligned}$$

Keeping these facts in mind, we move to the actual analysis of Simon's algorithm.

$$\begin{aligned} |0^{2n}\rangle &\overset{H^{\otimes n} \otimes I}{\longmapsto} \left(\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \right) \otimes |0^n\rangle \\ &\overset{U_f}{\longmapsto} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle \end{aligned}$$

From here, we measure the second n qubits. This is a measurement of the uniform superposition of all function outcomes, which in other words means we get outcome $z \in \{0,1\}^n$ with probability $\frac{1}{|\text{Range}(f)|}$.

Perhaps more interestingly, the state collapses to

$$\begin{cases} \left(\frac{1}{\sqrt{2}}|x_z\rangle + \frac{1}{\sqrt{2}}|x_z \oplus s\rangle \right) \otimes |z\rangle & \text{if } s \neq 0^n \\ |x_z\rangle \otimes |z\rangle & \text{if } s = 0^n \end{cases}$$

where x_z is one pre-image of z. We now concern ourselves with the first n qubits (since the second n have already been measured). We now break into two cases based on the triviality of the period s.

Case $s = 0^n$. Applying another $H^{\otimes n}$ gives us

$$\overset{H^{[\otimes n]}}{\longmapsto} H^{\otimes n}|x_z\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x_z \cdot y} |y\rangle$$

From here, measuring this n qubit state gives us a uniformly random $y \in \{0,1\}^n$. Indeed, $y \cdot s = 0^n$ for all y, when $s = 0^n$.

Case $s \neq 0^n$. Applying $H^{\otimes n}$ gives us

$$\begin{aligned}
\overset{H^{[\otimes n]}}{\longmapsto} \quad & H^{\otimes n}\left(\frac{1}{\sqrt{2}}|x_z\rangle + \frac{1}{\sqrt{2}}|x_z \oplus s\rangle\right) \\
&= \frac{1}{\sqrt{2}}H^{\otimes n}|x_z\rangle + \frac{1}{\sqrt{2}}H^{\otimes n}|x_z \oplus s\rangle \\
&= \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{x_z \cdot y}|y\rangle + \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{(x_z \oplus s)\cdot y}|y\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} \left(\frac{1}{\sqrt{2}}(-1)^{x_z \cdot y} + \frac{1}{\sqrt{2}}(-1)^{x_z \cdot y + s\cdot y}\right)|y\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} \left(\frac{1}{\sqrt{2}}(-1)^{x_z \cdot y}(1 + (-1)^{s\cdot y})\right)|y\rangle
\end{aligned}$$

From here, notice that the inner-most grouped term, $(1 + (-1)^{s\cdot y})$ takes on a nonzero value only when $s \cdot y = 0$. Otherwise, destructive interference occurs, and the terms cancel out to leave us with 0.

Thus, upon the final measurement, we'll still receive a uniformly random y such that $s \cdot y = 0$.

13 Lecture 13: May 19th

Grover's algorithm, quadratic speed-up, unstructured search, phase oracle, diffusion operator, Grover iteration, Grover's theorem, analysis of Grover's algorithm.

13.1 Grover's algorithm

Over the last lecture, we saw a quantum algorithm that gave us an exponential speed-up over a classical one in *Simon's algorithm*. Today, we'll explore a more modest, quadratic speed-up for the problem of "unstructured search".

Let's define today's problem: Given black-box access to a function $f : \{0,1\}^n \to \{0,1\}$ such that there is a unique x^* satisfying $f(x^*) = 1$, and $f(x) = 0$ for all $x \neq x^*$. Find x^*.

Classical algorithms would require $\Omega(2^n)$ queries to find x^* through brute-force search, and even randomized algorithms do not provide any meaningful speed-up.

What we'll see, however, is that Grover's algorithm will be able to do this in $O(\sqrt{2^n})$ queries! Although upon first glance, this may seem like a modest speedup, but it is broadly applicable. Also, in terms of today's compute, we see that, 1 billion seconds $\approx$ 31 years, while $\sqrt{1 \text{ billion seconds}} \approx 9$ hours. So yeah. Pretty significant stuff.

This speed-up is relative to an oracle (black-box) for f. But why do we consider the problem in the oracle framework? Well, in this model, we can actually prove a lower bound of $\Omega(2^n)$ queries classically.

More precisely, for many problems we care about (NP problems), the function f is both (1) efficiently computable, and (2) hard to invert. Grover's algorithm is the first to break through the strict, exponential bound on these functions.

A word of caution: we might hope to speed up search by querying f on all inputs through a uniform superposition:

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|0\rangle \xmapsto{U_f} \frac{1}{\sqrt{2^n}} \sum_x |x\rangle|f(x)\rangle$$

But even then, the x^* we're looking for would still have an exponentially small amplitude, which is not helpful at all.

With that said, let's finally get into some plausible approaches. We first define some key ingredients we'll need, starting with the phase oracle.

Definition (Phase oracle)**.** For a function f, its phase oracle is the unitary O_f such that

$$O_f|x\rangle \mapsto (-1)^{f(x)}|x\rangle$$

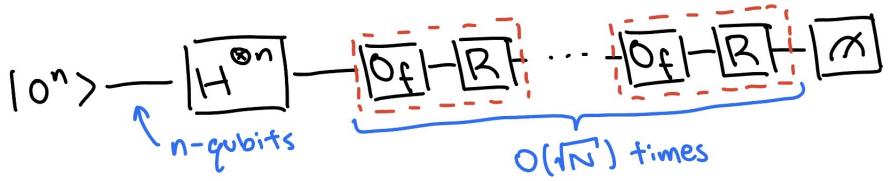
But woah, this seems a bit too powerful no? Why can we take O_f to be our oracle, instead of U_f, without loss of generality?

$$\begin{aligned} U_f|x\rangle|-\rangle &= \frac{1}{\sqrt{2}}U_f|x\rangle|0\rangle - \frac{1}{\sqrt{2}}U_f|x\rangle|1\rangle \\ &= \frac{1}{\sqrt{2}}|x\rangle|f(x)\rangle - \frac{1}{\sqrt{2}}|x\rangle|1\oplus f(x)\rangle \\ &= |x\rangle \otimes \left(\frac{1}{\sqrt{2}}|f(x)\rangle - \frac{1}{\sqrt{2}}|1\oplus f(x)\rangle\right) \\ &= (-1)^{f(x)}|x\rangle|-\rangle \end{aligned}$$

So we've implemented O_f with a single query to U_f, with the help of an auxiliary qubit $|-\rangle$ that was returned to us as $|-\rangle$. Thus, this WLOG substitution of using O_f rather than U_f seems entirely plausible.

Let's finally get into Grover's circuit. Begin by letting $N = 2^n$ (easier to write).

The circuit to implement Grover's algorithm is as follows:



where R is the n-qubit unitary acting as

$$\begin{aligned} &|+\rangle^{\otimes n} \mapsto |+\rangle^{\otimes n} \\ &|\phi\rangle \mapsto -|\phi\rangle \qquad \text{for all } |\phi\rangle \text{ orthogonal to } |+\rangle^{\otimes n} \end{aligned}$$

Theorem 13.1 (Grover's Theorem).
Grover's algorithm outputs the market input, i.e. x^* such that $f(x^*) = 1$, with probability $\frac{1}{100}$ (if one exists)

Remark. The exact probability is insignificant: any constant success probability (independent of n) can be boosted to any $1-\delta$ precision with $O(\log(1/\delta))$ repetitions.

13.2 Understanding Grover's Algorithm

So, what is the algorithm actually doing?

Intuitively, each application of (RO_f) increases the amplitude on $|x^*\rangle$ by $\approx \frac{2}{\sqrt{N}}$. After $\sqrt{N}$ application, amplitude $\approx \sqrt{N}\frac{2}{\sqrt{N}}$, giving us a constant.

Further, recall that R was defined as $|+\rangle^{\otimes n} \mapsto |+\rangle^{\otimes n}$, and $|\phi\rangle \mapsto -|\phi\rangle$ for all $|\phi\rangle$ orthogonal to $|+\rangle^{\otimes n}$. This is simply a reflection about $|+\rangle^{\otimes n}$!

Moreover, notice that O_f acts as $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$. We can rewrite this as

$$\begin{cases} |x\rangle \mapsto |x\rangle & \text{if } f(x) = 0 \\ |x\rangle \mapsto -|x\rangle & \text{if } f(x) = 1 \end{cases}$$

In other words, this is *also* a reflection, about the subspace of non-marked inputs.

With ALL of this in mind, let's finally analyze Grover's algorithm. We start with $|0^n\rangle$ and prepare a uniform superposition

$$|0^n\rangle \overset{H^{\otimes n}}{\longmapsto} \frac{1}{\sqrt{2^n}} \sum_x |x\rangle \coloneqq |U\rangle$$

We can rewrite

$$\begin{aligned} |U\rangle &= \frac{1}{\sqrt{N}}|x^*\rangle + \frac{1}{\sqrt{N}} \sum_{x:f(x)=0} |x\rangle \\ &= \frac{1}{\sqrt{N}}|G\rangle + \sqrt{\frac{N-1}{N}}|B\rangle \end{aligned}$$

where $|G\rangle := |x^*\rangle$, and $|B\rangle := \frac{1}{\sqrt{N-1}} \sum_{x:f(x)=0} |x\rangle$.

From here, we can do another rewrite to say $|U\rangle = \sin\theta|G\rangle + \cos\theta|B\rangle$, where $\theta = \sin^{-1}\left(\frac{1}{\sqrt{N}}\right)$.

Now that we can visualize $|U\rangle$ geometrically in the $|G\rangle, |B\rangle$ basis, what happens after each (RO_f) iteration?

$\overset{O_f}{\longmapsto}$: O_f here is a reflection about $|B\rangle$! In general,

$$O_f(\alpha|G\rangle + \beta|B\rangle) = -\alpha|G\rangle + \beta|B\rangle$$

$\overset{R}{\longmapsto}$: R is a reflection about $|U\rangle$ (the uniform superposition)! In general,

$$R(\alpha|U\rangle + \beta|U^\perp\rangle) = \alpha|U\rangle - \beta|U^\perp\rangle$$

where $|U^\perp\rangle$ is a state orthogonal to $|U\rangle$ in the $|G\rangle, |B\rangle$ basis.

We started with $|U\rangle$ at an angle of θ with $|B\rangle$, and we now have $(RO_f)|U\rangle$, at an angle 3θ with $|B\rangle$. Remember, our goal is to eventually arrive at $|U\rangle = |G\rangle$.

We soon see that each application of (RO_f) sequence rotates by angle 2θ towards $|G\rangle$, which means that after k applications, the state is

$$(RO_f)^k|U\rangle = \sin((2k+1)\theta)|G\rangle + \cos((2k+1)\theta)|B\rangle$$

How large does k need to be to get $\frac{1}{100}$ probability of finding x^*?

Well recall that for small θ, $\sin\theta \approx \theta$, meaning $\theta \approx \frac{1}{\sqrt{N}}$. Also, $\sin((2k+1)\theta) \approx (2k+1)\theta \approx (2k+1)\frac{1}{\sqrt{N}}$.

Thus, picking $k = O(\sqrt{N})$, we have $(2k+1)\frac{1}{\sqrt{N}} = O(1)$. So $O(\sqrt{N})$ queries to O_f are sufficient to find x^* with constant probability.

14 Lecture 14: May 21st

Analysis of Grover's algorithm, optimal iteration count k, implementation of diffusion operator R gate. Intro to Shor's algorithm, hardness of factoring, the period-finding problem.

14.1 Wrap-up Grover's Algorithm

Last lecture, we described in some detail of *Grover's Algorithm*. Recall the premise of the algorithm: given black-box access to $f : \{0,1\}^n \to \{0,1\}$, find some "marked" input x^* such that $f(x^*) = 1$. Further recall that after each application of (RO_f), our vector $|U\rangle$ gets rotated by 2θ in the $|G\rangle, |B\rangle$ plane.

Today, we'll give a formal analysis of this algorithm. First, we write a proof for how to pick the best number of iterations, k, for which we apply

$$(RO_f)^k(\sin\theta|G\rangle + \cos\theta|B\rangle) = \sin((2k+1)\theta)|G\rangle + \cos((2k+1)\theta)|B\rangle$$

Given this, ideally we'd want to pick some $\tilde{k}$ such that $(2\tilde{k}+1)\theta = \frac{\pi}{2}$. This gives us $\tilde{k} = \frac{\pi}{4\theta} - \frac{1}{2}$.

But now there's a problem — $\tilde{k}$ is clearly not an integer! So let's just pick k to be the closest integer to $\tilde{k}$. Then, this gives us

$$\begin{aligned}
\Pr[\text{Fail}] = \cos^2((2k+1)\theta) &= \cos^2((2\tilde{k}+1)\theta + 2(k-\tilde{k})\theta) \\
&= \cos^2(\frac{\pi}{2} + 2(k-\tilde{k})\theta) \\
&= \sin^2(2(k-\tilde{k})\theta) \\
&\leq \sin^2\theta \\
&= \left(\frac{1}{\sqrt{N}}\right)^2 \\
&= \frac{1}{N}
\end{aligned}$$

The probability of failure here is "small". Since $\tilde{k} = \frac{\pi}{4\theta} - \frac{1}{2}$ and $|k - \tilde{k}| \leq \frac{1}{2}$, we have $k \leq \frac{\pi}{4\theta}$. Further, since $\sin\theta \leq \theta$ we have $k \leq \frac{\pi}{4\sin\theta} = \frac{\pi}{4}\sqrt{N}$.

Thus, to achieve a probability of failure bounded by $\frac{1}{N}$, we must pick k on the order of $O(\sqrt{N})$.

*****

Recall our R gate, defined as the following:

$$\begin{cases} |+\rangle^{\otimes n} \mapsto |+\rangle^{\otimes n} \\ |\phi\rangle \mapsto -|\phi\rangle & \text{for all } |\phi\rangle \perp |+\rangle^{\otimes n} \end{cases}$$

The question now is: how do we implement the R gate efficiently? "Efficiently" here means with a poly(n)-size circuit.

To tackle this question, we break it down a bit. Let's ask a simpler question: how do we implement some $\tilde{R}$ defined as

$$\begin{cases} |0^n\rangle \mapsto |0^n\rangle \\ |x\rangle \mapsto -|x\rangle & \text{for all } x \neq 0^n \end{cases}$$

equivalently, we say $R : |x\rangle \mapsto (-1)^{\mathrm{OR}(x)}|x\rangle$. Before we talk about how to implement $\tilde{R}$, let's first talk about how we can implement R given $\tilde{R}$. We see that

$$R := H^{\otimes n}\tilde{R}H^{\otimes n}$$

Not convinced? Let's double check. For input $|+\rangle^{\otimes n}$,

$$|+\rangle^{\otimes n} \overset{H^{\otimes n}}{\longmapsto} |0^n\rangle \overset{\tilde{R}}{\longmapsto} |0^n\rangle \overset{H^{\otimes n}}{\longmapsto} |+\rangle^{\otimes n}$$

Now let $|+_b\rangle$ be defined as $|+\rangle$ if $b = 0$, and $|-\rangle$ if $b = 1$. Now for some arbitrary input $|+_{x_1}\rangle \cdots |+_{x_n}\rangle$, we have

$$\begin{aligned} |+_{x_1}\rangle \cdots |+_{x_n}\rangle &\overset{H^{\otimes n}}{\longmapsto} |x_1 \cdots x_n\rangle \\ &\overset{\tilde{R}}{\longmapsto} (-1)^{\mathrm{OR}(x_1 \cdots x_n)}|x_1 \cdots x_n\rangle \\ &\overset{H^{\otimes n}}{\longmapsto} (-1)^{\mathrm{OR}(x_1 \cdots x_n)}|+_{x_1}\rangle \cdots |+_{x_n}\rangle \end{aligned}$$

Okay, this is good. Now we finally talk about implementing $\tilde{R}$. Here's the plan:

1. Write a circuit to compute $\mathrm{OR}(x)$ reversibly (remember to use aux qubits and uncompute garbage)
2. Convert it to $|x\rangle \mapsto (-1)^{\mathrm{OR}(x)}|x\rangle$ (as in Deutsch's)

We'll be doing this in HW4 on a real quantum computer (woah!)

14.2 Shor's Algorithm

We now dive into another quantum algorithm — **Shor's Algorithm**. Given some n-bit integer $N \in \{1, \ldots, 2^n - 1\}$ where $N = pq$ for some primes p and q, Shor's algorithm factors N in time poly(n).

This is surprising as the best classical algorithm we know of runs in time $O(2^{n^{1/3}})$.

The importance of Shor's algorithm goes without saying, but I'll say it here anyway. All security of the RSA cryptosystem relies on the assumption that factoring is "hard" (i.e. exponential time). Shor's algorithm breaks all of these assumptions.

At the heart of the algorithm, Shor's algorithm solves a problem called "period-finding". Given black-box access to $f : [M] \to [N]$ such that $f(x) = f(x + s)$ for all x. Find s.

In other words, f is s-periodic: $f(x) = f(x + s) = f(x + 2s) = \cdots$. How many queries are needed to find s classically? Naively, we could try all possible s. This results in $O(N) = O(2^n)$.

We can improve this a bit. Imagine if we stumble upon some y such that $f(x) = f(y)$, then we'd know $s \mid y - x$. This suggests that after finding a few collisions, we'd have $k_1 \cdot s$, $k_2 \cdot s$, ... We then compute $\gcd(k_1 s, k_2 s, \dots)$. This is most likely s. Finding one collision takes about time $O(\sqrt{2^n})$, and we'd only need a constant number of them to find s with overwhelming probability. Thus, we say that we can find s in $O(\sqrt{2^n})$.

We'll see next lecture that we can solve period-finding quantumly with $\text{poly}(n)$ queries to f.

15 Lecture 15: May 26th

Period-finding problem, (classical) Fourier Transform, Fourier Transform matrix, roots of unity, magnitude-position independence, period extraction via Fourier Transforms.

15.1 Period-finding & Quantum Fourier Transform

Over the next week, we are going to be dedicating the next three lectures to understand Shor's algorithm. In doing that, we'll be taking a trip down memory lane and reviewing Fourier Transform, then talking about what it means for Fourier transforms to be done quantumly, and lastly putting everything together to form Shor's algorithm.

To begin, recall the problem of *period-finding*: Given black-box access to $f : [M] \to [N]$ that is periodic for some period $s \in [N]$, i.e., $f(x) = f(y)$ iff $y = x + \ell \cdot s$ for some $\ell \in \mathbb{Z}$. The goal: find s.

$$f(x) = f(x+s) = f(x+2\cdot s) = \cdots = f(x+(L-1)\cdot s)$$

As per usual, we will denote $M = 2^m$ and $N = 2^n$. Classically, $O(\sqrt{N}) = O(\sqrt{2^n})$ queries are necessary and sufficient to find the period. We will see that quantumly, we can do this in $O(\log N) = \text{poly}(n)$.

Let's dive into the procedure:

1. Create the state $\frac{1}{\sqrt{M}} \sum_{x\in[M]} |x\rangle|0\rangle$. Here $|x\rangle$ is a m-qubit state, where $x_1 \dots x_m$ is the binary representation of $x \in [M]$. Additionally, $|0\rangle$ is an n-qubit state.
2. Apply $U_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$ (the unitary that computes f reversibly). We get $\frac{1}{\sqrt{M}} \sum_{x\in[M]} |x\rangle|f(x)\rangle$.
3. Measure the second register. Let z be the outcome. Then, the first register will collapse to
$$\frac{1}{\sqrt{L}} \left(|x\rangle + |x+s\rangle + \cdots + |x+(L-1)\rangle\right)$$
where these are all inputs that map to z. Here, L denotes the number of such inputs.

This is looking good! But now, how do we extract s from this state? In other words, given a periodic data sequence, how do you extract its period? Furthermore, how can we do this efficiently on a quantum computer? (our input is very large (size M))

Let's break this down one by one. In order to find periodicity, we use a tool called the Fourier Transform. And to do this on a quantum computer, we just slap the word "quantum" in front of it — a Quantum Fourier Transform.

*****

Let's review: **Fourier Transform.**

Consider the vector representation of the state $\frac{1}{\sqrt{L}}(|x\rangle + |x+s\rangle + \cdots + |x+(L-1)s\rangle)$. This is

$$\frac{1}{\sqrt{L}} \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

This is a 2^m-dimensional vector, where the value "1" appears at positions $x + \ell \cdot s$. Fourier Transform is a linear transformation mapping a periodic vector to another vector in which the magnitude of the k-th entry depends only on s and k (and crucially, not on x).

Let's define this formally.

Definition (Fourier Transform Matrix)**.** The **Fourier Transform Matrix** F_N is the $N \times N$ matrix such that its (j,k)-th entry is

$$(F_N)_{jk} \coloneqq \omega^{j \cdot k}$$

where ω is the N-th root of unity, i.e., $\omega = e^{2\pi i/N}$. Additionally, $j \cdot k$ is multiplication of numbers in $\{0, \ldots, N-1\}$.

Example 15.1.

$$F_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} \omega^{0\cdot 0} & \omega^{0\cdot 1} \\ \omega^{1\cdot 0} & \omega^{1\cdot 1} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

This is exactly the Hadamard gate!

Example 15.2.

$$F_2 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & i^4 & i^6 \\ 1 & i^3 & i^6 & i^9 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

Remark. For any $N \in \mathbb{N}$, F_N is a unitary matrix!

Now let's investigate the effect of F_N on some periodic vector v. Notice that when we multiply the two, we get that for the k-th entry,

$$(F_N \cdot v)_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{k \cdot j} \cdot v_j$$

Suppose now that $v = \frac{1}{\sqrt{L}} \left[0 \cdots 0 \ 1 \ 0 \cdots 0\right]^T$ as described in our period-finding problem. Then, the expression becomes

$$(F_N \cdot v)_k = \frac{1}{\sqrt{N \cdot L}} \sum_{\ell=0}^{L-1} \omega^{k \cdot (x + \ell \cdot s)} = \frac{1}{\sqrt{N \cdot L}} \cdot \omega^{k \cdot x} \cdot \sum_{\ell=0}^{L-1} \omega^{k \cdot \ell \cdot s} \tag{1}$$

Note now that the magnitude of $(F_N \cdot v)_k$ does not depend on x! This is because

$$|(F_N \cdot v)_k| = \left| \frac{1}{\sqrt{NL}} \cdot \omega^{k \cdot x} \cdot \sum_{\ell=0}^{L-1} \omega^{k \cdot \ell \cdot s} \right| = \frac{1}{\sqrt{NL}} \cdot \left| \sum_{\ell=0}^{L-1} \omega^{k \cdot \ell \cdot s} \right|$$

Take a look at the sum. Let's call this sum μ and see that $\mu = 1 + \omega^{k \cdot s} + \omega^{2 \cdot k \cdot s} + \cdots + \omega^{(L-1) \cdot k \cdot s}$. Is $|\mu|$ large or small? Well, $|\mu|$ is large if all the $\omega^{\cdots}$ are close to 1, since they're all roots of unity. Thus, it would depend on the relationship between k and s.

Assume for simplicity that s divides N. This is not true in general, but it will make the intuition much clearer. In this perfect world, what values of k would make μ large?

- *Case 1:* $k \cdot s$ is a multiple of N. Then, $k \cdot s = c \cdot N$. Then, $\omega^{k \cdot s} = \omega^{c \cdot N} = (\omega^N)^c = 1$.

  In this case, all the temrs in μ point in the same direction! We call this phenomenon "constructive interference", and it produces a large value of μ.

- *Case 2:* $k \cdot s$ is NOT a multiple of N. Then, all the terms in μ point in different directions, and cancel out perfectly (in this perfect world where s divides N. In the real world, this would produce a small value of μ), producing exactly $\mu = 0$.

  We call this phenomenon "destructive interference".

16 Lecture 16: May 28th

Period-finding subroutine, Fourier Transform matrix as a unitary, Quantum Fourier Transform (QFT), the complete period-finding procedure.

16.1 Period-finding & Quantum Fourier Transform (cont'd)

Today we'll be talking more about the period-finding problem, and start moving towards how to solve this quantumly. But first, we recall the premise of the *period-finding* problem: given black-box access to $f : [M] \to [N]$ that is s-periodic. Find s. This quantum subroutine is crucial towards Shor's algorithm.

Further recall the first steps of period-finding quantumly:

1. Create $\frac{1}{\sqrt{M}} \sum_{x \in [M]} |x\rangle |f(x)\rangle$
2. Measure the second register. First register collapses to

$$|\psi\rangle = \frac{1}{\sqrt{L}}(|x\rangle + |x+s\rangle + \cdots + |x+(L-1)s\rangle)$$

In order to extract the pre-image for x, we discussed the *classical* Fourier Transform: F_M is a $M \times M$ matrix with entries $(F_M)_{jk} = \frac{1}{\sqrt{M}} \cdot \omega^{j \cdot k}$, where $\omega = e^{\frac{2\pi i}{M}}$.

Example 16.1. Let $v = \frac{1}{\sqrt{L}}[0 \cdots 0\ 1\ 0 \cdots 0]^T$, where $v_j = \frac{1}{\sqrt{L}}$ if and only if $j = x + \ell \cdot s$. We discussed last time how this is the exact vector representation of $|\psi\rangle$. When we multiply $F_M \cdot v$, we see that the magnitude of the k-th entry is not dependent on x at all, but rather only on k and s.

Furthermore, only the entries for which $k \cdot s$ is (close to) a multiple of M will carry a large magnitude, since $e^{\frac{2\pi i k}{L}} = 1$. Otherwise, we have small magnitude (exactly equal to 0 when s divides M).

Naturally now, we can ask if F_M really helps us compute the period "efficiently"? The truth is, even though it has a nice way of interacting with the period, computing $F_M \cdot v$ will take $O(M^2)$ time naively (optimized to be $O(M \log M)$). Since M is an m-bit number, this operation becomes exponential with respect to m. Ew.

In fact, why do we even bother doing this? If we just iterate through v and track the distance between two entries of '1', we can find the period just as efficiently as the classical Fourier Transform.

Well, it turns out this isn't entirely a waste of time, because we can apply FT quantumly. Let's dig in.

*****

Let $M = 2^m$. Let's talk about the big idea as to where the quantum speed-up occurs. First, notice F_M is unitary. So, if we think of F_M as a "quantum operation", it is a unitary transformation on m qubits (a very small number of qubits)!

Suppose then we had a m-qubit state $|\psi\rangle$ whose 2^m amplitudes encode a periodic data sequence. Further suppose we can implement this unitary F_M. Then $F_M|\psi\rangle$ is a state whose vector representation is the fourier transform of the vector representation of $|\psi\rangle$.

Importantly, we don't get to "read" the 2^m amplitudes, since we'd have to make measurements to extract them.

Conveniently, we only care about knowing what the entries of $F_M|\psi\rangle$ with large magnitude are. Fortunately, when we measure $F_M|\psi\rangle$ in the standard basis, we'd get an outcome whose probability depends on the magnitudes of the corresponding amplitude!

This means measuring in the std basis will likely yield one of the k-th entries such that $(F_M|\psi\rangle)_k$ has large magnitude.

So given all this, can we implement F_M efficiently? Fortunately, yes again! There is an implementation using only poly(m) gates. This is referred to as the **Quantum Fourier Transform (QFT)**.

Definition (Quantum Fourier Transform)**.** The m-qubit unitary F_M is referred to as the QFT. It acts as follows: for some standard basis vector $|j\rangle$ in $\mathbb{R}^{2^m}$,

$$|j\rangle \mapsto \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \omega^{k \cdot j} |k\rangle$$

There is a quantum circuit of size $O(m^2)$ implementing the QFT.

*****

Now, let's assume such a circuit exists (because it does). How do we solve the period-finding problem given this?

1. Prepare $|\psi\rangle = \frac{1}{\sqrt{L}}(|x\rangle + |x+s\rangle + \cdots + |x+(L-1)s\rangle)$ for some x
2. We apply QFT to $|\psi\rangle$, obtaining $F_M|\psi\rangle$.
3. Measure $F_M|\psi\rangle$ in the standard basis. We are likely to get k such that $k \cdot s \approx c \cdot M$ for some integer c ($k \cdot s$ is close to a multiple of M).

   In other words, $k \approx c \cdot \frac{M}{s}$. How do we figure out s?
4. Repeat steps 1 - 3 a few times. We get

$$k_1 \approx c_1 \cdot \frac{M}{s} \qquad k_2 \approx c_2 \cdot \frac{M}{s} \qquad \cdots$$

At this point, we know each k_i (the outcome of our measurements) and we know M (it's given). Assuming s divides M again, then $\frac{M}{s}$ is an integer, at which point we take the gcd of the k_i's, this will likely give us $\frac{M}{s}$, from which we simply solve for s.

But that's quite the assumption. What if s doesn't divide M? Now $\frac{M}{s}$ is no longer an integer. We now spawn in a random fun fact: for each repetition,

$$\Pr[\gcd(c_i, s) = 1] \geq \frac{1}{\log m}$$

We only need to repeat $O(\log m)$ times to have 0.99 probability of having at least one $k_i = c_i \cdot \frac{M}{s}$ with $\gcd(c_i, s) = 1$. When this is the case, simplifying the fraction $\frac{k_i}{M}$ will give us $\frac{c_i}{s}$, where s is just the denominator!

Remark. This is all contingent on the assumption that $k \cdot s$ is strictly equal to $c \cdot M$. This is not always the case. In which case we'd need *continued fraction expansion*.

17 Lecture 17: Jun. 2nd

Period-finding subroutine, Shor's Theorem, factoring to period-finding reduction, square-and-mult algorithm, the complete Shor's algorithm.

17.1 Period-finding and Shor's algorithm

Second last lecture!! Last week we discussed the *period-finding* problem, and today we'll see it applied in Shor's algorithm. Particularly, we'll see how to reduce the factoring problem to period-finding.

But first, recall the problem of *period-finding* once again: given black-box access to $f : [M] \to [N]$ that is s-periodic. Find s.

Further recall the *Fourier Transform*: F_M is the $M \times M$ matrix with entries $(F_M)_{jk} = \frac{1}{\sqrt{M}}\omega^{j \cdot k}$. The great "discovery" from last lecture was that this matrix F_M is unitary. If we let $M = 2^m$, we can think of F_M as a m-qubit unitary.

This is precisely the *Quantum Fourier Transform* (QFT), which is explicitly defined:

$$|j\rangle \longmapsto \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \omega^{k \cdot j} |k\rangle$$

We said that there is a $O(m^2)$ quantum circuit implementing the QFT_M.

Putting these pieces together, we came to our quantum algorithm for period-finding:

1. Create uniform superposition: $\frac{1}{\sqrt{M}} \sum_{x \in [M]} |x\rangle|f(x)\rangle$
2. Measure the second register. The first register will collapse to

   $$|\psi\rangle = \frac{1}{\sqrt{L}} \left(|x\rangle + |x+s\rangle + \cdots + |x+(L-1)s\rangle\right)$$

   Crucially, at this point, we do not know the value of x nor s, and we'd only know the image $f(x)$ that we measured.
3. Apply QFT and measure. Let k denote the outcome. With high probability, $k \cdot s \approx c \cdot N$ for some integer c.
4. Repeat steps 1-3 $O(\log n)$ times. Recover s from the k's.

The last part ("recover s from the k's") was a bit hand-wavy in the previous lecture, where we specifically explored the special case where s divides N (we did this through calculating the gcd).

But in general, s may not divide N. There's some scary math in this case (applying "continued fraction expansion"). We won't worry about it for the scope of this course, but trust that it works.

*****

For the rest of the lecture (and the course), we assume that we have an efficient black-box algorithm for period-finding. We now discuss *Shor's* period-finding.

Here, let $N = p \cdot q$ for some primes p and q. Our goal is to find the values of p and q. As usual, let n be the number of digits of N, that is, $N = 2^n$.

Theorem 17.1 (Shor's Theorem).

Factoring reduces to period-finding for the function $f : [M] \to [N]$ such that $f(x) = a^x \bmod N$.

Here, M is some sufficiently large number: $N^2 \leq M \leq 2N^2$. Further, a is coprime to N.

Not to worry, we'll eventually prove this reduction. But first, some intuition: why does the period of this exponential function matter?

Well, the period of f is the smallest s such that $f(x) = f(x+s)$ for all x. By definition of f, this is the smallest s such that $a^x = a^{x+s} \bmod N$. Since a and N are coprime, this is the smallest s such that $a^s = 1 \bmod N$.

We find the period for $f(x) = a^x \bmod N$. Period-finding algorithm assumes black-box access to f in the form of the unitary U_f. Before we apply our canonical period-finding algorithm as described above, we'd need to implement U_f first.

The main question is: is U_f efficiently implementable? We'll see that the answer is yes (obviously, otherwise we wouldn't be talking about this in class).

Naively, to compute $a^x \bmod N$, we could simply multiply a by itself x number of times. But x could potentially be $O(M)$, which then makes this exponential and therefore not efficient.

Instead, consider the following:

1. Compute $a, a^2, a^4, a^8, \ldots, a^{2^{m-1}} \bmod N$
2. Notice $a^x = a^{2^{m-1}x_1 + 2^{m-2}x_2 + \cdots + 2^0 \cdot x_m} = a^{2^{m-1} \cdot x_1} \cdot a^{2^{m-2} \cdot x_2} \cdots a^{2^0 \cdot x_m}$.

   Note that now, combined with step 1, we have quite a beautiful

$$a^{2^{m-i} \cdot x_i} = \begin{cases} a^{2^{m-i}} & \text{if } x_i = 1 \\ 1 & \text{if } x_i = 0 \end{cases}$$

   So, compute a^x by taking the product of a subset of numbers from step 1.

The total runtime of the above algorithm: $O(m \cdot n \cdot \log n) = O(n^2 \log n)$. Polynomial in n yay!

Okay great, now that we know how to compute $f(x)$ efficiently black-box, how do we go from period-finding to factoring? Recall $N = p \cdot q$.

1. Pick a number $a \in \{0, \ldots, N-1\}$ uniformly at random. Compute $\gcd(a, N)$. If $\gcd(a, N) > 1$, then we're done — a and N must share a factor being p or q lol. Most likely though, $\gcd(a, N) = 1$.
2. Pick M as a power of 2 such that $N^2 \leq M \leq 2N^2$. Use the period-finding subroutine to find s such that $a^s = 1 \bmod N$.
3. Suppose we're lucky and s is even. Then $\frac{s}{2}$ is an integer. We can then rewrite the above expression as

   $$a^s - 1 = 0 \bmod N \iff (a^{s/2} + 1)(a^{s/2} - 1) = 0 \bmod N$$

   This gives us the factors up to some constant c. If we're unlucky, one of $a^{s/2} + 1$ or $a^{s/2} - 1$ is already a multiple of N.

   Note that $a^{s/2} - 1$ cannot be a multiple of N on its own, because this would mean $a^{s/2} = 1 \bmod N$. But since s is the period (smallest s such that $a^s = 1 \bmod N$), this cannot be the case.

   To bring it back then, the only thing that could go wrong is if $a^{s/2} + 1$ is a multiple of N.
4. Suppose we're lucky (again), and $a^{s/2}+1$ is not a multiple of N. Since $(a^{s/2}+1)(a^{s/2}-1) = c \cdot N$, then $\gcd(a^{s/2} + 1, N) = p$ or q.

   Fact. One can show that we are lucky (twice) with probability $\geq 1/2$.

18 Lecture 18: Jun. 4th

Implementing QFT, interference, $O(n^2)$ gates. Hamiltonians, time evolution of a quantum system, Hermitian matrix, Schrodinger's equation, Hamiltonian simulation.

NOTE: Notes for this lecture was unfortunately not completed in time. It contains materials that are not found on the final, and I was too busy studying to fill in the gaps </3

18.1 Implementing QFT

Last lecture so sad... Today we'll talk about the last piece of the puzzle in Shor's algorithm, which is to actually implement the QFT (the F_N unitary).

Begin by letting $N = 2^n$. We will show that QFT_N can be implemented using only $O(n^2)$ gates. Recall that $\text{QFT}_N|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{k\cdot j}|k\rangle$. We make a crucial observation:

$$\text{QFT}_N|j\rangle = \bigotimes_{\ell=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + e^{\frac{2\pi i j}{2^\ell}}|1\rangle\right)$$

Proof.

$$\begin{aligned}
\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{k\cdot j}|k\rangle &= \frac{1}{\sqrt{N}} \sum_{k_1,\ldots,k_n\in\{0,1\}} e^{\frac{2\pi i}{2^n}\cdot(2^{n-1}k_1+2^{n-2}k_2+\cdots+2^0k_n)\cdot j}|k_1\rangle\cdots|k_n\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{k_1,\ldots,k_n\in\{0,1\}} e^{\frac{2\pi i}{2^n}2^{n-1}\cdot k_1\cdot j}\cdot e^{\frac{2\pi i}{2^n}2^{n-2}\cdot k_2\cdot j}\cdots e^{\frac{2\pi i}{2^n}\cdot k_n\cdot j}|k_1\rangle\cdots|k_n\rangle \\
&= \frac{1}{\sqrt{2^n}} \left(\sum_{k_1} e^{\frac{2\pi i}{2^n}\cdot k_1\cdot j}|k_1\rangle\right) \otimes \left(\sum_{k_2} e^{\frac{2\pi i}{2^n}\cdot k_1\cdot j}|k_2\rangle\right) \otimes \cdots \otimes \left(\sum_{k_n} e^{\frac{2\pi i}{2^n}\cdot k_n\cdot j}|k_n\rangle\right)
\end{aligned}$$

Notice something about the exponential terms:

$$e^{\frac{2\pi i}{2^n}\cdot 2^{n-\ell}\cdot k_\ell\cdot j} = \begin{cases} 1 & \text{if } k_\ell = 0 \\ e^{\frac{2\pi i}{2^\ell}\cdot k_\ell\cdot j} & \text{if } k_\ell = 1 \end{cases}$$

We can then use this trick to see that

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{k\cdot j}|k\rangle = \bigotimes_{\ell=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + e^{\frac{2\pi i j}{2^\ell}}|1\rangle\right)$$

□

...

This suggests the following way to implement QFT_N:

1. Start by creating $|+\rangle^{\otimes n}$

Let R_k be such that $R_k|0\rangle = |0\rangle$ and $R_k|1\rangle = e^{\frac{2\pi i}{2^k}}|1\rangle$

19 Afterwords

It's been a while since I wrote something like this for any fo my lecture notes, but I thought since I enjoyed this class very much, it would be worth it to take some time and free-write my thoughts.

Disclaimer: Almost every lecture (with the exception of the last one) has complete notes that's been vetted through at least once (when I was studying for the final). Also, everything beyond this point is my personal opinion / interpretation of the course materials.

I originally took this class because many of my friedns had suggested it, and considering my overall enthusiasm for theoretical cs, I thought it would be a fun course to take.

And overall, as stated above, I enjoyed this course very much. I especially found my calling during the latter half of the course, when we started discussing the various quantum computations and algorithms. But of course, we would not have gotten there without the (somewhat grueling) first 5 weeks of the course, where it was legitimately straight up linalg simulator.

In general, as someone who thinks of these concepts highly intuitively rather than math based, one thing I wish the course went over more was the motivation / intuition behind quantum mechanics. Now obviously, "intuition behind quantum mechanics" famously doesn't exist. But beyond the first lecture of the double-slit experiment, where we hand-waved the entire thing into a "qubit", there wasn't much more motivation behind it.

Like, especially during the latter half of the course, I didn't know what was physically realizable at this point in time, and what was all theoretical, and I wish there were more discussions that.

That said, the lecture's I did most enjoy were (1) the non-local games lectures, and most importantly, the philosophical results (such as "certifiable randomness" and the lhv discussion), (2) the Elitzur-Vaidman tester and QKC lecture, (3) the quantum period-finding lectures (that's when everything started clicking for me), and (4) the latter half of the final lecture with the Hamiltonian simulations (of which there are no notes on).

Anyway, let's dive into a very very quick course recap.

*****

We began the quarter with an introduction to what "quantum" means, through the double-slit experiment. It was cool, but also went over my head very quickly before we got into INFINITE lin alg and complex variables.

From there, we thoroughly explored the propreties of a single "qubit" through a mathematical lens: superpositions, evolutions, measurements, unitaries, and so much more. This eventually culminated in a lecture (and a half) on the application of single-qubit systems, where we explored the power of just a single quantum unit through the Elitzur-Vaidman tester and the quantum key distribution (again, one of my favorite lectures).

After that, we moved onto generalizing single-qubit systems through the discussion of multi-qubit states. Ngl, this part of the course was a bit hard for me to conceptualize at first, especially the idea of n-qubit states needing 2^n parameters to describe.

Similar to our discussions on single-qubit states, we also discussed the superpositions, evolutions, mesaurements, unitaries, and so much more regarding n-qubit states as well. This is a perfect set-up for quantum computation, which was to come.

But right before the midterm, we also took a detour and talked about non-local games, where stuff got quite *philosophical* — true randomness, lhv theory, etc etc. I enjoyed these discussions very much, and although it was a bit abstract, it did really show the impact that quantum mechanics had on the world.

After the midterm, we dove into qunatum computation. We began by talking about what "quantum computation" actually meant: preparing qubit states (much like binary input), applying 1 to 2-qubit gates (much like binary circuits), and measuring a subset of the qubits to get some result (much like reading binary output). This is also where I started to realize how probabilistic everything is to be computed quantumly.

From there, we started our speedrun of all the ground-breaking quantum algorithms, which were: (1) deutsch's (constant vs balanced), (2) simon's (period-finding of binary functions), (3) grover's *search*, (4) period-finding & shor's algorithm.

TBH, i feel like (very naively) all the quantum algorithms we've talked about so far is just all different rephrasings of

1. put ur input and/or aux in superposition
2. apply the function (with appropriate aux)
3. find a clever way to measure the subset of output we're interested in to extract information

and depending on the procedure (mostly step 3), it could provide us with potentially a quadratic (grovers) or exponential (simons) speed-up over classically going over all inputs one-by-one.

And unfortunately, the course ended on a somewhat abrupt note. I know there's so much more to talk about with quantum computaitons, and i'm sure andrea would have loved to talk about many things as well, which is why a lot of content was crammed into the last few weeks and the course feels like it ended on a cliffhanger. Fade the quarter system fr. Anyway, hopefully i'll enroll in graduate quantum computation (CSE 534) next fall, so stay tuned!