# CS395T NOTES: QUANTUM COMPLEXITY THEORY

ARUN DEBRAY
OCTOBER 10, 2016

These notes were taken in UT Austin's CS395T (Quantum Complexity Theory) class in Fall 2016, taught by Scott Aaronson. I live-TEXed them using `vim`, so there may be typos; please send questions, comments, complaints, and corrections to a.debray@math.utexas.edu.

## Contents

Lecture 1.

## Introduction to Quantum Mechanics: 8/29/16

*"The big secret of quantum mechanics is how simple it is once you take the physics out of it."*

The course website is http://www.scottaaronson.com/qct2016/, and the syllabus is at http://www.scottaaronson.com/qct2016/syllabus-qct2016.pdf. We'll be mostly following lecture notes found at http://www.scottaaronson.com/barbados-2016.pdf.

This lecture's goal is to acquaint the listener with the basic concepts and notation that we'll use in the rest of the course; it's not presented as review, but everything else in the course depends on it. For this material, there are many excellent references, some of which are listed in the syllabus.

Quantum mechanics has a very underserved reputation for being very complicated. Mysterious, yes; counterintuitive, yes; but complicated is a bit much. All sorts of interesting consequences follow from a single change to the laws of probability, crucial to physics at the subatomic level, but thought to apply to everything in the universe.

A probability of something happening is a real number $p \in [0,1]$: it makes no sense to ask what a probability of $-1/3$ is, much less $i/3$. But quantum mechanics assigns a more general number, an *amplitude* $\alpha \in \mathbb{C}$, to an event. The thesis of quantum mechanics is that any isolated physical system's state can be described by a vector of its amplitudes.

In particular, systems in quantum mechanics have a dimension; intuitively, if there are $N$ different things you can observe, the system is $N$-dimensional. The simplest quantum systems are two-dimensional, where there are two possinilities ) and 1. These systems have a special name: *qubits*.

In general, we think of the state of a quantum system as a unit vector $\psi \in \mathbb{C}^N$ of length 1. These vectors are denoted using a notation that Paul Dirac invented in the 1930s, the *Dirac ket notation*. The syntax looks a little jarring at first, but is convenient in a lot of ways. A *ket* is a vector $|v\rangle$: a qubit has two basis vectors $|0\rangle$, representing an outcome of 0 and $|1\rangle$, similarly an outcome of 1, so a general state is $|v\rangle = \alpha|0\rangle + \beta|1\rangle$, representing a linear combination, or *superposition*, of the two options: $\alpha, \beta \in \mathbb{C}$ are complex numbers, and must satisfy a *normalization rule*: $|\alpha|^2 + |\beta|^2 = 1$. In other words, $|v\rangle$ stands in for the column vector $\left(\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right)$. Usually, ket notation will only be for unit vectors, but sometimes we might use it more generally.

This feels a little schizophrenic. Is it both at the same time? Is it neither? In popular books, these are the only ontological categories the writer can imagine, but these really belong in a different conceptual framework altogether.

In addition to column vectors, we like row vectors too, denoted with a *bra* $\langle v|$. However, since we're in the land of complex vector spaces, taking the transpose comes along with complex conjugation, so $\langle v| = (\alpha^*\ \beta^*)$. Combining these two notations, $\langle\cdot|\cdot\rangle$ is the notation for the inner product. Thus, that $v$ is a unit vector is succinctly expressed in the condition $\langle v|v\rangle = 1$.

Explicitly, if $|\psi\rangle = \alpha_1|1\rangle + \cdots + \alpha_N|N\rangle$ and $|\varphi\rangle = \beta_1|1\rangle + \cdots + \beta_N|N\rangle$, their inner product is $\langle\psi|\varphi\rangle = \alpha_1^*\beta_1 + \cdots + \alpha_N^*\beta_N$. This measures how similar two vectors are: if the inner product is 1, they lie on the same line, and are related, but if it's 0, they're orthogonal, and thus very different.

Relatedly, there is an *outer product* $|\psi\rangle\langle\varphi|$, which is a rank-1 $N \times N$ matrix whose $ij^{\text{th}}$ term is $\alpha_i\beta_j^*$.

There are two things one can do to quantum systems.

(1) One option is a *unitary transformation*. These should be thought of as doing something smooth and well-behaved. They are continuous, reversible, and deterministic.
(2) The other choice is a *measurement*. These are useful, especially if you want to actually learn anything about system, but these are discontinuous and irreversible, and famously are probabilistic. Quantum mechanics tells you probabilities, not certainties.

Maybe you're wondering how two so very different systems can coexist in the same universe. This is the *measurement problem*, and people have been discussing it for a century. In some sense, unitary transformations arise from changes of basis, but if you follow that viewpoint far enough, it seems like all of quantum mechanics is a particular change of basis! Yet there are ways in which the choice of basis matters; unitary transformations are information-preserving, relating to the very general physical principle that information cannot be destroyed. A unitary transformation might horribly transform information, but it's still there.

The measurement problem and its metaphysics notwithstanding, we can at least write down the mathematical rules for these transformations. A unitary evolution is multiplication by a matrix: $|\psi\rangle \mapsto U|\psi\rangle$, but $U$ must be norm-preserving, so that all valid quantum states map to valid quantum states.[1] Since unitary transformations should be reversible, we'd like $U$ to be an invertible matrix.

**Exercise 1.1.** Show that the following are equivalent for a linear transformation $U : \mathbb{C}^n \to \mathbb{C}^n$:

(1) $U$ is norm-preserving and invertible.
(2) $U$ preserves inner products, i.e. $\langle U\psi|U\varphi\rangle = \langle\psi|\varphi\rangle$ for all $|\psi\rangle, |\varphi\rangle \in \mathbb{C}^N$.
(3) $U^\dagger U = I$ (here, $^\dagger$ denotes conjugate transpose).
(4) The rows of $U$ are an orthonormal basis for $\mathbb{C}^N$.
(5) The columns of $U$ are an orthonormal basis for $\mathbb{C}^N$.

Such a matrix is called a *unitary matrix*.

**Example 1.2** (Qubit)**.** The simplest example is a qubit, whose vector space is spanned by two basis vectors $|0\rangle$ and $|1\rangle$ (so it has two complex dimensions, or four real dimensions). Thus, the possible superpositions are $\alpha|0\rangle + \beta|1\rangle$ such that $|\alpha|^2 + |\beta|^2 = 1$. Often, but not always, $\alpha$ and $\beta$ will be real, making them easier to draw; in this case, we just need $\alpha^2 + \beta^2 = 1$, defining a circle.

The *plus state* is $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, and the *minus state* is $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. $(|+\rangle, |-\rangle)$ is also an orthonormal basis for this space.

What are some unitary transformations? We have the identity

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

as well as the *NOT gate*

$$\text{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

In general, a *gate* will refer to a unitary matrix that's applied to only one or a few qubits.

[1] This is what keeps the probabilities adding up to 1.

The identity and the NOT gate make sense for classical probability too, sending probability vectors to probability vectors. This is not true for the next matrix, called the *phase gate*:

$$\text{Phase} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

which sends $(\alpha, \beta) \mapsto (\alpha, -\beta)$. There are other phases, e.g. replacing $-1$ by another root of unity. Similarly, the *Hadamard matrix* is

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

Notice that $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$, but $H|+\rangle = |0\rangle$ and $H|-\rangle = |1\rangle$, so the Hadamard matrix switches the normal basis and the plus-minus basis. Thus, $H^2 = I$.

Finally, there are rotation matrices

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

This rotates counterclockwise by the angle $\theta$.

Every unitary transformation in two dimensions is a product of rotations and reflections. Notice that the Hadamard matrix is not a rotation: we can apply $R = R_{\pi/4}$, which sends $|0\rangle \mapsto |+\rangle$ and $|1\rangle \mapsto -|-\rangle$. In general, $|\psi\rangle$ and $-|\psi\rangle$, as well as $i|\psi\rangle$, produce the same physical behavior: there's no experiment that can tell them apart. The classical analogue would be to move the whole universe twenty feet to the left: does anything actually change?

We can calculate $R|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $R|1\rangle = (-|0\rangle + |1\rangle)/\sqrt{2}$. Evaluating on $|+\rangle$, we have to cancel out a $|0\rangle/2$ and a $-|0\rangle/2$:

$$R\frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{\frac{|0\rangle+|1\rangle}{\sqrt{2}} + \frac{-|0\rangle+|1\rangle}{\sqrt{2}}}{\sqrt{2}} = |1\rangle.$$

This is called *destructive interference*: the two ways to obtain $|0\rangle$ were in opposite amplitudes, so they descrutively interfered, and cancelled out to zero probability. Similarly, the outcomes leading to 1 displayed *constructive interference*. A lot of the weirdness of quantum mechanics comes out of interference phenomena, since they behave so differently from classical mechanics. The *double-slit experiment* is an example: if a photon passes through two slits in an opaque material, there are alternating zones of light and dark, which makes classical sense. But the part that makes no sense classically is that if you close off one of the slits, photons can appear where they hadn't before.

This wasn't just a violation of intuition; it was a violation of the axioms of probability: classically, one assumes that the probabilities $p_A$ and $p_B$ of getting to that point after passing through the two slits $A$ and $B$, respectively, should add to the total probability of a photon landing at that spot, but the experiment disproved that. This and its analogues in atomic nuclei, etc. are why quantum mechanics works with amplitudes instead of probabilities. In fact, the amplitude of this process is the sum of the amplitude occuring from slit $A$ and the amplitude occuring from slit $B$. Destructive interference explains why closing slit $B$ affects the answer.

**Measurements.** Given a qubit $\alpha|0\rangle + \beta|1\rangle$, we want to know whether it's 0 or 1. The rule is $\Pr[0] = |\alpha|^2$ and $\Pr[1] = |\beta|^2$. This is called *Born's rule*, after Max Born (who won his Nobel for work including this!). But the second, and very important, thing that happens is that the state "collapses" to whichever measurement you observed. This is much like some people one encounters: they're not certain about their opinion on a topic, but once they're asked about it, they pick an opinion and stick to it, at least until a unitary transformation is applied to them. This is why one says that measurement in quantum mechanics is an irreversible process.

In general, if we have a superposition of $N$ outcomes $\alpha_1|1\rangle + \cdots + \alpha_N|N\rangle$, then $\Pr[i] = |\alpha_i|^2$. This is why global phase is irrelevant: the only way you can learn anything about a quantum system is measurement. Many of the paradoxes or misunderstandings people make implicitly assume there's some other way to measure the system. This also shows why there's no way to tell apart $|\psi\rangle$ and $-|\psi\rangle$: no measurement can distinguish them. It also explains interference: two amplitudes may both be nonzero, but if they're opposite in sign, the norm-squared of their sum is zero or nearly zero.

Measurement is denoted with a sort of speedometer $\pitchfork$.[2] Precomposing with a unitary transformation allows one to measure with respect to a different basis, e.g. using the Hadamard matrix is measurement in the $\{|+\rangle, |-\rangle\}$-basis. This means we'll get the outcome $+$ with probability $|\langle\psi|+\rangle|^2$ and outcome $-$ with probability $|\langle\psi|-\rangle|^2$. This is really just a rotation.

A pure 0 state always evaluates to 0. A pure 1 state always evaluates to 1. An equal superposition gives 0 half the time, and 1 half the time. But evaluating with respect to the $\{|+\rangle, |-\rangle\}$ basis turns pure states into equal superpositions and vice versa. In other words, $\Pr[|v_i\rangle] = |\langle\psi|v_i\rangle|^2$.

**Example 1.3.** We can generalize to systems of multiple qubits, placing them beside each other. A qubit might correspond to an electron with two energy states, or two spin directions (up and down), or any physical system that can be in either of two discrete states: quantum mechanics says there can also be a superposition. The variety of these systems leads to the variety of proposals for the physical architectures of a quantum computer.

Suppose now we have two photons. We refer to the composite of these systems with a tensor product: $(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle)$. We can distribute out the $\otimes$: the two-qubit space is actually spanned by the four basis vectors $\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$.[3] The amplitudes are

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle.$$

This is a unit vector in $\mathbb{C}^4$.

Conversely, one might want to factor a state as a tensor product:

$$\frac{|00\rangle - |01\rangle - |10\rangle + |11\rangle}{2} = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

However, not every state can be factored, e.g. $(|00\rangle + |11\rangle)/\sqrt{2}$: if we expanded it out, too many terms would become 0, causing a contradiction. If a state can be written as a tensor product, it's called *separable*; a state that cannot be written in this way is *entangled*.[4] This is all that entanglement is, the quantum-mechanical version of correlation. Entanglement should not be changed by local unitary transformations, though it may be destroyed by measurement (see below). A global unitary transformation, involving both qubits, could entangle or unentangle qubits.

In general, separability arises when we have a state space $\mathbb{C}^{AB} = \mathbb{C}^A \otimes \mathbb{C}^B$. This can get more interesting in infinite-dimensional Hilbert spaces, but most of the spaces we consider in this class will be finite-dimensional, so just $\mathbb{C}^N$ for some $N$. Some quantum systems appearing in quantum optics arise not as tensor products, but as symmetric products, which can cause people to get tangled up talking about entanglement.

How do we measure in the two-qubit system? It's simple if you present a state as $a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$; the probability of $|00\rangle$ is $|a|^2$. Physically, though, this has surprising implications: the two qubits may be very far apart. If Alice has one and Bob has the other, then it's possible for Alice to only measure her qubit, which has state 0 with probability $\Pr[0] = |a|^2 + |b|^2$. If she observes 0, the two outcomes vanish: even before Bob can make a measurement, the state undergoes a *partial collapse* to $(1/\sqrt{|a|^2 + |b|^2})(a|0\rangle + b|1\rangle)$ (and something similar with $c$ and $d$, if Alice sees 1). This may be generalized to systems of any dimension.

These statements are true for both separable and entangled qubits, but for separable qubits, they reduce to trivialities.

If Alice and Bob's qubits are in the Bell pair state, and Alice measures a 0, then she know that whenever Bob measures it, he will measure a 0 (and similarly, if she measures a 1, so must he). Bob's qubit "updates" instantaneously as soon as Alice measures it, no matter how far away they are. This is what famously unsettled Einstein, since it violates the relativistic principle that information cannot exceed the speed of light; this is so-called "spooky action at a distance." This doesn't seem useful for creating a faster-than-light telephone, since Alice has no control over the bit she sends. When you pick up a newspaper, that determines

[2] This should be a semicircle with an arrow pointing to the upper right, but I don't know how to TeX that yet.

[3] In practice, to save effort, these are simplified: $|0\rangle \otimes |0\rangle$ is denoted $|0\rangle|0\rangle$ or even $|00\rangle$.

[4] There are ways to quantify the amount of entanglement; this state, sometimes called the *Bell pair*, *EPR pair*, or *singlet state*, happens to be maximally entangled.

the headline on every copy of that newspaper, but that's not as spooky: there are other variables which explain the correlations without FTL travel. A similar result for the two qubits would be called a *local hidden-variable theory*, postulating a shared secret between the two qubits that explains the correlation.

It took about thirty years for the question to be formulated in this way and then to be answered.

**Theorem 1.4** (No-communication theorem)**.** *It's not possible to use entangled states for faster-than-light communication.*

So quantum mechanics does not break relativity. However, there is no hidden-variable theory, either: quantum mechanics is an intermediate point between the hidden-variable theory and true FTL communication. Yet if you wanted to simulate quantum mechanics in a classical universe, the simulation would need FTL communication.

Bell conducted an experiment that led to this conclusion, which was really an early phenomenon of a familiar concept in theoretical computer science, the two-prover game. There are three actors: Alice, Bob, and a referee. Alice and Bob cannot communicate, but the referee can send challenges to Alice and Bob and collect their responses. Alice and Bob are trying to cooperate, trying to get the referee to accept with the largest probability. They may plan a strategy in advance, but cannot communicate during the experiment, just like in a separated police interrogation.

In the modern reformulation of Bell's theorem, this game is called the *CHSH game*. The referee sends a random bit $x \in \{0,1\}$ to Alice and an independent random bit $y \in \{0,1\}$ to Bob. Alice sends back a random bit $a = a(x, r_a)$ and Bob sends back a random bit $b = b(y, r_b)$ (here, $r_a$ and $r_b$ are the sources of randomness for Alice and Bob, respectively). Alice and Bob win the game if $a + b = xy \pmod 2$.

Clearly this is not a game many people play for fun. Classically, Alice and Bob can win 3/4 of the time by always responding 0 — and one can prove that, classically, there is no strategy that does better, a fact called *Bell's inequality*.

But if Alice and Bob shared a Bell pair of two qubits in advance, there is a way of correlating their measurements in this state such that their probability of winning is $\cos^2(\pi/8) \approx 0.85$. This is a lot more subtle than sending messages back and forth: by themselves, Alice and Bob don't notice anything special, since you need both of their answers. In this case, there can be no local hidden-variable theory, and entanglement is not just shared classical randomness.

The protocol takes a little time to explain, but Alice measures her qubit in a specific basis if she sees a 0, and in a different basis if she sees a 1, and Bob does something similar. One can show that the probability of winning is $\cos^2$ of the difference of their measurement angles, which can be as high as $\pi/8$. A second inequality, called *Tsirelson's inequality*, says that no matter how many qubits they share, Alice and Bob cannot do better.

Any theory with local hidden variables predicts that the success probability is at most 3/4, but quantum mechanics doesn't, so quantum mechanics is not a hidden-variable theory. Bell never imagined his experiment to be actually carried out, but in the 1980s, people actually did this, and the universe is consistent with the predictions given by quantum mechanics. Most physicists weren't surprised: this was not the first experiment testing quantum mechanics, and it's passed all of them, and wasn't the last.

A neat slogan is that, like everything else, Bell's theorem comes down to interference influencing correlation. Bell's theorem, rather than getting into metaphysical questions, uses quantum entanglement to solve problems, and in this way anticipates the field of quantum communication.

**Mixed states.** Suppose Alice and Bob have qubits in a Bell pair. What state does Alice see? Naïvely, one might expect Alice to end up with $|+\rangle$, but if this were the case, then if she measured it in the $\{|+\rangle, |-\rangle\}$ basis, she should always get $|+\rangle$. So what does it mean for her to apply a Hadamard gate $H$ to her qubit only? We take the tensor product $H \otimes I$: the Hadamard for Alice, and the identity for Bob. Often, the unitary operators we care about can be broken up into smaller components. One way of thinking about this: for all possible states of Bob's qubit ($|0\rangle$ and $|1\rangle$), Alice applies the Hadamard gate.

When Alice does this, the state looks like $(1/2)(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$: Alice observes $|0\rangle$ and $|1\rangle$ with equal probability. That's weird. The takeaway is that a rose (Bell pair) by any other name (basis) still smells as sweet (is still a Bell pair). Something new is happening: Alice's qubit is behaving more like a classical random bit than a quantum state.

To talk about a piece of an entangled system, one needs a more general description of states, called *mixed states*: these are probability distributions over different states. For example, as we just saw for the Bell pair, Alice's state is $1/2\ |0\rangle$ and $1/2\ |1\rangle$. To be clear: this is not a superposition, just a plain old random bit. This is a surprisingly classical form of uncertainly!

There's an important subtlety in mixed states, which is why people don't always think of them as probability distributions over pure states.[5] Specifically, there are different probability distributions that give rise to the same mixed state: Alice's mixed state is $1/2\ |0\rangle$ and $1/2\ |1\rangle$, but is indistinguishable from the mixed state $1/2\ |+\rangle$ and $1/2\ |-\rangle$: in any orthonormal basis, each of these produces each outcome half of the time. Writing out a mixed state as a distribution over pure states is redundant. Fortunately, there's a representation for mixed states that's not redundant, using what's called *density matrices*. These are a whole new (equivalent) way to view quantum mechanics itself, and is usually preferred by experimentalists.

Suppose I have a probability distribution of pure states $\{p_i, |\psi_i\rangle\}_{i=1}^n$; then, the corresponding density matrix is

$$\rho = \sum_{i=1}^n p_i |\psi_i\rangle\langle\psi_i|.$$

This is an $n \times n$ complex matrix.

For example, the density matrix for Alice's mixed state in the Bell pair is

$$\frac{1}{2}\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1 & 0\end{pmatrix} + \frac{1}{2}\begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}0 & 1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}.$$

You can compute that if we started with $(1/2, |+\rangle)$ and $(1/2, |-\rangle)$, we end up with the same matrix.

Generally, if we have an entangled system of the form $\sum \alpha_i |i\rangle|\psi_i\rangle$, then Bob's density matrix is

$$\rho_{\text{Bob}} = \sum_i |\alpha_i|^2 |\psi_i\rangle\langle\psi_i|.$$

This also eliminates global phase.

If $U$ is a unitary matrix, then it acts on $\rho$ by conjugation:

$$\rho \longmapsto \sum_i \rho_i U|\psi_i\rangle\langle\psi_i|U^\dagger = U\rho U^\dagger.$$

Another advantage of the density matrix is that measuring the mixed state in this basis just requires the diagonal entries: $\Pr[|i\rangle] = \rho_{ii}$.

That is, along the diagonal of a density matrix $\rho$, there's a probability distribution. Sometimes, that's all we have, and the density matrix is diagonal (including the Bell state). But density matrices may also have off-diagonal entries, e.g. the superposition $(1/\sqrt{2})(|0\rangle + |1\rangle)$. This is not the same, because in the $\{|+\rangle, |-\rangle\}$-basis, its outcome is always $|+\rangle$. Its density matrix is

$$\frac{1}{2}\begin{pmatrix}1 & 1\\1 & 1\end{pmatrix}.$$

Experimentalists regard off-diagonal entries as the signature of quantum behavior. In practice, the diagonal has the largest terms, but the bigger the off-diagonal terms are, the better the experiment was (according to the experimentalists).

The no-communication theorem says that quantum entanglement still preserves locality. More precisely, if Alice and Bob have entangled quantum systems, there is no combination of unitary transformations and measurements that Alice can make to her system that changes Bob's density matrix, unless we condition on Alice's measurement outcomes. This is very similar to how measurements affect classical correlation. Since Bob's density matrix can be used to calculate every possible outcome of every possible measurement Bob can make, this theorem encompasses anything Alice and Bob can do.

Density matrices don't provide us any new physics. Given a density matrix $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, there's an equivalent pure state

$$\sum_i \sqrt{p_i}|i\rangle \otimes |\psi_i\rangle,$$

and from the perspective of the second observer, these look the same.

[5] A pure state is a degenerate mixed state, which assigns probability 1 to a single state.

Lecture 2.

## The Quantum Toolbox: 9/12/16

Last time, we reviewed (or introduced) quantum mechanics, including state spaces, qubits, and measurements. We talked about entanglement, Bell's inequality, and the no-communication theorem, and what is and isn't counterintuitive about them. We also introduced density matrices, which provide an equivalent formulation for everything in quantum mechanics.

Today we'll cover a few remaining phenomena in quantum mechanics of one through three qubits, the distance between two quantum states, general notions of measure, and other ingredients that we'll need. Then, we'll introduce quantum circuits and build up to defining BQP, the complexity class bounded over polynomial-time quantum circuits.

These remaining phenomena will inform what we are and aren't allowed to do with qubits, which will come up again and again in quantum complexity theory. For example, we learned that measurement is destructive: it's modeled as an irreversible process. Wouldn't it be great if we could work around that? Reproducible measurements are a cornerstone of science, so it would be nice.

**The no-cloning theorem.** Specifically, what if we had a procedure that could copy states? We could start with a quantum state $|\psi\rangle$ and an *ancilla* (an extra qubit) $|0\rangle$, apply some unitary transformation and obtain $\{|\psi\rangle, |\psi\rangle\}$, so we could measure the first in one basis, and the second in another basis. However, this is not possible, thanks to the suggestively named no-cloning theorem.

Perhaps this is striking — in classical mechanics, it's very possible to copy information. This is one of the foundations of the Internet (as well as software and music piracy).

Suppose $|\psi\rangle = (\alpha|0\rangle + \beta|1\rangle)$, and we tensor that with the ancillary qubit $|0\rangle$. We want a unitary transformation

$$(\alpha|0\rangle + \beta|1\rangle)|0\rangle \longmapsto (\alpha|0\rangle + \beta|1\rangle)(\alpha|0\rangle + \beta|1\rangle) = \alpha^2|00\rangle + \alpha\beta|01\rangle + \beta\alpha|10\rangle + \beta^2|11\rangle.$$

So we want a $4 \times 4$ matrix sending $(\alpha, 0, \beta, 0) \mapsto (\alpha^2, \alpha\beta, \alpha\beta, \beta^2)$. This is not linear, so there's no such matrix. Thus, it's not possible to perfectly clone, and moreover this is a robust phenomenon: it's possible to prove theorems about approximate cloning, and only very weak approximations are allowed.

Another way to understand this is that unitary transformations preserve angles and inner products. Suppose $|\varphi\rangle$ and $|\psi\rangle$ are such that $|\langle\psi|\varphi\rangle| = c \in (0,1)$: they're neither parallel nor perpendicular. If we were to clone this, we'd obtain $|\langle\psi^{\otimes 2} \mid U \mid \varphi^{\otimes 2}\rangle| = c^2 < c$ after acting by the unitary matrix $U$, which means this isn't actually unitary. In general, there are some irreversible operations that increase the inner product, but none decrease it.

If $\varphi$ and $\psi$ are orthogonal or parallel, then we can clone: this is essentially a reduction to the classical case, where information can be duplicated.

**Monogamy of entanglement.** Suppose in the classical world we have three bits that are correlated: if you look at any two, you know the value of the third (e.g. knowing they xor to 1). This is called a *promiscuous entanglement* (really).

Alternatively, consider three qubits in the three-qubit analogue of the Bell pair, called the *GHZ state*:

$$\frac{|000\rangle + |111\rangle}{\sqrt{2}}.$$

Give the first qubit to Alice, the second to Bob, and the third to Charlie. One day, Charlie isn't answering his calls, so how does this affect Alice and Bob? Let's compute the density matrix for Alice and Bob's qubits. If Charlie's qubit is a 0, both Alice and Bob have 0, and similarly for 1. Thus, the density matrix is

$$\begin{pmatrix} 1/2 & & & \\ & 0 & & \\ & & 0 & \\ & & & 1/2 \end{pmatrix},$$

which is a classical-like system, a coin flip. The idea is that Charlie might have measured his qubit, and the system acts like he did, summed over all possible measurements. All three are entangled, but no two are, like the Borromean rings of topology (see Figure 1).

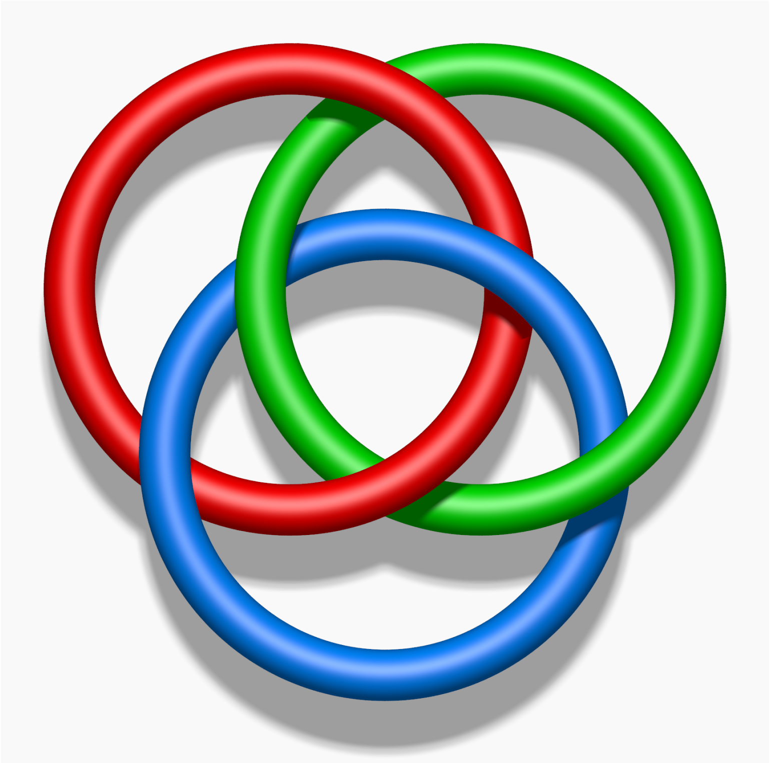

FIGURE 1. The Borromean rings, all three of which are linked, but no two of which are. Source: https://en.wikipedia.org/wiki/Borromean_rings.

This phenomenon is called *monogamy of entanglement*: if Alice and Bob are maximally entangled, Alice can't be maximally entangled with Charlie: quantum entanglement is "more jealous" than classical entanglement.[6]

Recall that a mixed state $\rho_{AB}$ is separable if it's possible to write it as a probability distribution over (tensor) product states, and that entanglement is the absence of separability. This isn't a very efficient definition of entanglement, which may worry the complexity theorists in this course, and indeed: Gurvits proved in 2003 that deciding whether a state is separable is an NP-hard problem, by embedding the subset-sum problem into it. So unless P = NP, there's going to be no clean formula to detect entanglement.

Most reductions to NP are *Karp reduction*, in which a "yes" solution to problem $A$ is translated to a "yes" solution to problem $B$ in polynomial time, and similarly for "no" solutions. But there are also *Cook reductions*, in which one has an oracle for problem $A$ and can make multiple queries to solve problem $B$. It was known for a while that there were some examples of NP problems which needed Cook reductions rather than Karp reductions, but Gurvits' proof was the first example for any real-world application.

Nobody knows the hardness of approximate separability (are these states almost separable?). There's some evidence that it's at least $O(n^{\sqrt{2}(\log n)})$, or else 3-SAT would be weirdly fast, but the problem is pretty wide open. So it's still quite hard to quickly tell whether states are entangled, and this frustrates a lot of experimentalists. There are some sufficient conditions, e.g. *entanglement witnesses* such as violating classical behavior such as the Bell inequality.

**Quantifying entanglement and distances.** The essential idea behind quantifying entanglement is called LOCC, for *local operations and classical communication*. This is a set of operations that don't increase entanglement: local unitary operators on Alice's or Bob's side, and classical communication. Any good measure of entanglement should be invariant under LOCCs.

[6] One often says that *maximal entanglement* means something that can be rotated into a Bell pair by a unitary matrix. This isn't so hard to define, but it's hard to come up with a good quantification of non-maximal entanglement. It gets more complicated for mixed states.

**Definition 2.1.** Suppose Alice and Bob have qubits in a state $\rho_{AB}$.

- The *entanglement of formation* $E_F(\rho_{AB})$ is the minimal number of Bell pairs needed to form $\rho_{AB}$.[7]
- The *distillable entanglement* $E_D(\rho_{AB})$ is the maximal number of Bell pairs that can be extracted from $\rho_{AB}$ by LOCCs.

As an analogy, a stock has a buy price and a sell price.

**Theorem 2.2.** *For pure states, these two measures coincide, and also coincide with the Shannon entropy of* $|\alpha_i|^2$ *in the density matrix* $\sum \alpha_i |v_i\rangle |w_i\rangle$.

It's possible to use these to quantify monogamy of entanglement. For example, given a mixed state on three regions $\rho_{ABC}$, it's possible to show that

$$E_D(\rho_{AB}) + E_D(\rho_{BC}) \le \log_2 \dim B.$$

For many things in this course, we'll need a way to measure distances. For pure states, the absolute value of the inner product is a good measurement, but in general, mixed states are more complicated. So we'll need to think about generalizations of probability distributions.

There are a few ways to measure distances of probability distributions, each of which has a quantum generalization. Probably the most useful measure is the *total variation distance*

$$\mathrm{TV}(\{p_i\}, \{q_i\}) = \frac{1}{2} \|\{p_i\} - \{q_i\}\|_{L^1} = \frac{1}{2} \sum_i |p_i - q_i|.$$

This is a really nice measure: it defines a metric, satisfying the triangle inequality and all that, but it also has a nice statistical interpretation: it's the greatest possible difference between two outcomes of the same event, and there is an experiment producing this difference. This generalizes to quantum states.

**Definition 2.3.** Let $\rho$ and $\sigma$ be two mixed states, and let $\{\lambda_1, \ldots, \lambda_n\}$ be the eigenvalues of $\rho - \sigma$. Then, their *trace distance* is

$$\|\rho - \sigma\|_{\mathrm{tr}} = \frac{1}{2} \sum |\lambda_i|.$$

Since (discrete) probability distributions are mixed states with diagonal matrices, this recovers the total variation distance.

Total variation distance 1 means that two distributions don't overlap, and are perfectly distinguishable. Similarly, two quantum states have trace distance 1 if they can be perfectly distinguished by a measurement: they live in orthogonal subspaces, in a sense.

**Exercise 2.4.** Show that the trace distance satisfies the triangle inequality, and therefore is actually a distance metric:

$$\|\sigma_1 - \sigma_3\|_{\mathrm{tr}} \le \|\sigma_1 - \sigma_2\|_{\mathrm{tr}} + \|\sigma_2 - \sigma_3\|_{\mathrm{tr}}.$$

Moreover, since conjugation by a unitary matrix doesn't change eigenvalues, the trace distance doesn't depend on basis:

$$\|\rho - \sigma\|_{\mathrm{tr}} = \|U\rho U^\dagger - U\sigma U^\dagger\|_{\mathrm{tr}}.$$

Finally, we'll make use of the fact that if some measurement accepts $\rho$ with probability $p$, it accepts $\sigma$ with a probability in $[p - \delta, p + \delta]$, where $\delta = \|\rho - \sigma\|_{\mathrm{tr}}$.

**Non-orthonormal bases.** So far, all of our measurements have been in orthonormal basis. According to our rules, we don't actually know how to do anything else, but this was like pure vs. mixed states: we started with just pure states, and had to eventually draw out mixed states.

The idea is that if you entangle a qubit or two with some ancillas, measurements might take on more possible values, rather than just two. So we want to understand what can be measured when we're allowed to entangle an unlimited number of ancillary qubits.

One introduces the formalism of POVMs, or *positive operator-valued measures*, for this. Given a set of Hermitian, positive definite matrices $E_i$ summing to the identity matrix, there is a procedure that returns $i$

[7] Technically, this is the limiting rate as we take more and more copies of this state, so isn't always an integer. The same is true for distillable entanglement.

with probability $\mathrm{Tr}(E_i\rho)$ for any state $\rho$, and every procedure defines a collection of matrices in this way. We won't prove this; see Nielsen-Chuang's book for more information.

For example, if $\rho$ is an ordinary measurement (matrix) in the basis $|\psi_1\rangle, \dots, |\psi_n\rangle$, then let $E_i = |\psi_i\rangle\langle\psi_i|$ (which, in this basis, has a 1 in position $(i, i)$ and 0s everywhere else, projecting onto $|\psi_i\rangle$). Then,

$$\mathrm{Tr}(E_i\rho) = \mathrm{Tr}(|\psi_i\rangle\langle\psi_i|\rho) = \langle\psi_i \mid \rho \mid \psi_i\rangle,$$

which is how we defined measurement last lecture.

The POVM formalism is incomplete in that it doesn't specify the post-measurement state, and depending on implementation, the result isn't determined.

**Superoperators.** These various formalisms can be unified into *superoperators* $\rho \to \$(\rho)$, which can be thought of as "allowed operations on operators." For example, we could take $\$(\rho) = U\rho U^\dagger$ for some unitary operator $U$, or we could let $\$(\rho)$ zero out the non-diagonal entries. Another example maps every state to a particular state. Superoperators are deterministic operators from mixed states to mixed states, though the image may be a non-pure state (and thus probabilistic).

To formalize this, start with a set $\{E_1, \dots, E_n\}$ of (not necessarily square) matrices such that

$$\sum_i E_i E_i^\dagger = I.$$

These define the superoperator

$$\$(\rho) = \sum_i E_i^\dagger \rho E_i.$$

One good exercise is to write down the matrices $E_i$ for the three superoperators mentioned above.

There's a theorem that all measurements involving ancillary systems can be expressed using superoperators, unifying several of the formalisms we've defined so far.

Let's combine these concepts into a lemma that we'll use many times. We've reiterated that measurement is a destructive process, but not all measurements are destructive; in a sense, measurements are only destructive when we don't have a basis vector near the state we're measuring. If you start with a state that's very close to $|0\rangle$, then almost all of the time, the state doesn't change by very much, so we haven't lost very much information. In quantum mechanics, this is sometimes referred to as the *information disturbance tradeoff*. The more randomness is generated, the more the state is disturbed.

**Lemma 2.5** (Almost as good as new lemma or gentle measurement lemma)**.** *Let $\rho$ be a mixed state and $M = \{E, I - E\}$ be a two-outcome POVM (so we accept $\rho$ with probability $\mathrm{Tr}(E\rho)$), and suppose that $\Pr[M(\rho) \text{ accepts}] \geq 1 - \varepsilon$. Then, it is possible to implement $M$ in such a way that the post-measurement state $\widetilde{\rho}$ satisfies $\|\widetilde{\rho} - \rho\|_{\mathrm{tr}} \leq \sqrt{\varepsilon}$.*

The intuition is that the trace distance is how much we've changed $\rho$ by, but since the probability is high, the damage is small.

*Proof.* The first observation is that we can reduce to pure states, because any mixed state is a convex combination of projectors onto pure states:

$$\rho = \sum_i \rho_i |\psi_i\rangle\langle\psi_i|,$$

and the square root function is convex, so the sum of the square roots of the individual errors is at most the square root of the total error.

The second is that we only have to look at a two-dimensional subspace, since we've reduced to states of the form $|\psi\rangle \otimes |0 \cdots 0\rangle$. In this case, we're conjugating by a unitary matrix with (at least) $1 - \varepsilon$ and (at most) $\varepsilon$ on the diagonals, so it has to look like

$$\begin{pmatrix} 1 - \varepsilon & \sqrt{\varepsilon(1-\varepsilon)} \\ \sqrt{\varepsilon(1-\varepsilon)} & \varepsilon \end{pmatrix}.$$

After measuring, we simply have

$$\begin{pmatrix} 1 - \varepsilon & 0 \\ 0 & \varepsilon \end{pmatrix},$$

and you can calculate the trace distance to be $\sqrt{\varepsilon(1-\varepsilon)}$. ☒

This is an important counterpart to the no-cloning theorem: measurement may destroy information, but if it accepts good values with near-certain probability, then the amount of damage done is very small. This is useful for quantum money, or copy-protected quantum software: you can prevent outright duplication, but observing that the certificate is valid doesn't destroy it.

In fact, multiple measurements are okay. The idea is that if there's at most a 1% chance that you'll fall off of cliffs and a 2% chance you'll be struck by lightning, you have at most a 3% chance of suffering both, no matter the correlations between the two events. The corresponding quantum version will be very useful.

**Theorem 2.6** (Quantum union bound)**.** *Suppose we apply POVMs $E_1, \dots, E_k$ in order to a mixed state $\rho$, and suppose that* $\operatorname{Tr}(E_i\rho) \geq 1-\varepsilon$*. Then,*

$$\Pr[E_1, \dots, E_k \text{ all accept}] \geq 1 - k\sqrt{\varepsilon}$$

*and $\|\widetilde{\rho} - \rho\|_{\text{tr}} \leq k\sqrt{\varepsilon}$, where $\widetilde{\rho}$ is the post-measurement state.*

This does not follow immediately from Lemma 2.5.

*Proof.* We need to use the linearity of quantum mechanics. The triangle inequality tells us that

$$\|\rho - E_k \circ \cdots \circ E_1(\rho)\|_{\text{tr}} \leq \|\rho - E_k(\rho)\|_{\text{tr}} + \|E_k(\rho) - E_k \circ E_{k-1}(\rho)\|_{\text{tr}} + \cdots + \|E_k \circ \cdots \circ E_2(\rho) - E_k \circ \cdots \circ E_1(\rho)\|_{\text{tr}}.$$

Applying a superoperator can't decrease the trace distance, so

$$\leq \|\rho - E_k(\rho)\|_{\text{tr}} + \|\rho - E_{k-1}(\rho)\|_{\text{tr}} + \cdots + \|\rho - E_1(\rho)\|_{\text{tr}}.$$

Applying Lemma 2.5,

$$\leq k\sqrt{\varepsilon}.$$ ☒

This isn't a tight bound, and the bound has been improved to $\sqrt{k\varepsilon}$ in a recent paper.

One technical point is that we should pin down how to tell that all of the $E_i$ accept, We can add ancillary qubits and entangle them to record whether a state accepts: if they all start with $|1_{E_i \text{ accepts}}\rangle$, they move by at most $\sqrt{\varepsilon}$ each, so we recover the bound we wanted.

∽ · ∼

Quantum computing acts on systems of $n$ qubits, where $n$ may be large. Such a state may be represented by

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

where we normalize

$$\sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1.$$

This is kind of incredible: it tells us that to simulate 1000 particles, you need $2^{1000} \approx 10^{300}$ pieces of information, more pieces than all the subatomic particles in the universe. This growth in complexity is why problems like the Schrödinger equation are so difficult, and people earn Nobels for "small" advances.

This was turned on its head at first to provide more efficient simulations of quantum systems in physics or chemistry, and now has found additional applications. But we still have a little way to go in our formalization of quantum computers: we could also have been talking about a very classical notion of probability distributions on $n$ bits, but nobody actually thinks of this as $2^n$ pieces of information. In quantum mechanics, though, these $2^n$ pieces of information are actually there in nature, thanks to interference. This suggests that creatively arranging interference can make for a computation that's a speedup from classical computation.

In some sense, this would all come from a $2^n \times 2^n$ unitary transformation, but not any unitary matrix (flip the $n^{\text{th}}$ qubits iff the first $n-1$ define a Turing machine that solves the halting problem). This problem arises in classical mechanics too: the halting problem is a Boolean function,[8] but is not accessible to us.

For this reason, we look for Boolean functions that can be implemented with relatively few AND, OR, and NOT gates A general Boolean function on $n$ bits can be implemented in $O(2^n/n))$ bits. This is due to

[8] A *Boolean function* is a function $f : \{0,1\}^n \to \{0,1\}$.

Shannon's counting argument: there are $2^{2^n}$ Boolean functions on $n$ bits. Since the NAND gate is universal, we can ask how many ways there are to arrange $T$ NAND gates. The first has $\binom{n}{2}$ possibilities to choose from, and the second can also ask for the output of the first, so it has $\binom{n+1}{2}$ possibilities. This can be approximated as

$$\binom{n}{2}\binom{n+1}{2}\cdots\binom{n+T-1}{2} \leq (n+T)^{2T}.$$

Thus, if we want to model all Boolean functions on $n$ bits using at most $T$ NAND gates, then $(n+T)^{2T} \geq 2^{2^n}$, i.e. $2T\log(n+T) \geq 2^n$, so $T = \Omega(2^n/n)$. This argument also shows that almost all functions require this many gates! But to know which ones do and don't at a large scale would provide a solution to whether $\mathrm{P} = \mathrm{NP}$: we know that only a small number of these functions are accessible or "nice."

Something very similar will happen in quantum computing: there will be a quantum circuit of a set of qubits, and an arrow of time. We can apply unitary operators as gates: applying the Hadamard operator $H$ at time $t$ to only bit 3 is akin to applying $I \otimes I \otimes H \otimes I$ to the whole system. This reflects the engineering behind all of this: quantum computers will be built out of small building blocks, so it's best to describe operations on those blocks.

We've already seen the Hadamard gate; let's discuss some others.

The *controlled NOT* gate is probably the next most important. If the first qubit is 1, it flips the second qubit; if the first qubit is 0, it doesn't. This can be written as

$$\mathrm{CNOT} : |x, y\rangle \longmapsto |x, y \oplus x\rangle.$$

In truth-table form, $|00\rangle \mapsto |00\rangle$, $|01\rangle \mapsto |01\rangle$, $|10\rangle \mapsto |11\rangle$, and $|11\rangle \mapsto |10\rangle$. Like the Hadamard gate, this is its own inverse; in matrix form, it's

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

We can use this to entangle two states: starting with $|00\rangle$ and applying $H \otimes I$, then CNOT, the result is the Bell state, so you can entangle with two gates. Another weird fact is that, after a change of basis, a controlled NOT from $x$ to $y$ is equivalent to one from $y$ to $x$.

This can shed light on some philosophy of quantum mechanics: if you want to measure a state $\alpha|0\rangle + \beta|1\rangle$, *decoherence theory* says we consider a system

$$(\alpha|0\rangle + \beta|1\rangle) \otimes |\text{Measurement}\rangle \otimes |\text{You}\rangle,$$

and there is some unitary transformation involving the atoms of the measuring device (and you!) with the outcome

$$(\alpha|0\rangle|\text{Measured } 0\rangle + \beta|1\rangle|\text{Measured } 1\rangle) \otimes |\text{You}\rangle.$$

When you look at it, this collapses to

$$\alpha|0\rangle|\text{Measured } 0\rangle|\text{You } 0\rangle + \beta|1\rangle|\text{Measured } 1\rangle|\text{You } 1\rangle.$$

This makes sense from a purely formal perspective, but actually thinking about what it *means*, or whether it means anything, is difficult and inconclusive. But as long as we talk about systems external to ourselves, this isn't as controversial.

Anyways, this provides an explanation for the CNOT gate: the target qubit is measuring the first qubit, just like entanglement. If you believe in the many-worlds perspective (which, well, is very metaphysical), then measurement is just a CNOT operation!

**Definition 2.7.** A gate set $G$ is *universal* if we can use gates from $G$ to implement any unitary operator on any number of qubits.

At some point, we'll worry about precision, but not yet. With no further conditions on the definition, these are necessarily infinite: there are uncountably many unitary operators, but a finite set can only generate countably many operators. Nonetheless, there is something nice to be said about these.

**Theorem 2.8.** *The set of* CNOT *and all* 1*-qubit gates is universal.*

We're not going to prove this.

Over a system of $n$ qubits, this can be approximated by $4^n$ gates, which follows from a dimension-counting argument: the dimension of the manifold $\mathrm{U}(2^n)$ is $4^n$. This means that we need systems of at least $T = 4^{n-1}$ gates to simulate all of these, since then $4T \geq 4^n$. Once again, the things that we can calculate efficiently are a vast subset of everything possible.

In real life, we cannot implement all of these perfectly: most are irrational (e.g. rotate by $1/e$), and some even are undecidable. For computational purposes, we only need to understand this to finite precision, thanks to the quantum union bound.

**Definition 2.9.** A gate set $G$ is *approximately universal* if we can use gates from $G$ to approximate any unitary transformation on any number of qubits to any desired precision.

Thanks to arbitrary precision, it doesn't matter what norm we use to evaluate this; they're all equivalent. This definition also allows finite gate sets to be approximately universal. We still don't completely understand approximately universal gate sets, but here are some nice facts.

**Proposition 2.10.** *For almost all 1-qubit gates $U$,*[9] CNOT $+U$ *is (approximately) universal.*

**Proposition 2.11.** *For almost all 2-qubit gates $U$,* CNOT $+U$ *is (approximately) universal.*

**Example 2.12.** The gate set

$$\left\{\mathrm{CNOT}, \begin{pmatrix} 3/5 & 4i/5 \\ 4/5 & -3i/5 \end{pmatrix}\right\}$$

is (approximately) universal.

The second matrix has to have $i$s somewhere, so it can see[10] the unitary group rather than just the real orthogonal group; the real part of the above matrix, along with the CNOT gate, densely generate $\mathrm{O}(n)$.

The third most important gate is the *Toffoli gate*, sometimes called "controlled controlled NOT." It flips the third qubit iff the first and second qubits are 1, akin to a reversible AND. One can write an 8 × 8 matrix or summarize it as $|x, y, z\rangle \mapsto x, y, z \oplus xy$.

**Proposition 2.13** (Shi, 2001)**.** *The gate set*

$$\left\{\text{Toffoli}, H, \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}\right\}$$

*is (approximately) universal.*

It's also good to know which gate sets are not universal. These include gate sets that only act classically, e.g. CNOT plus a Toffoli gate, or a Toffoli gate and a phase: these won't put anything into superposition that wasn't already. There are more interesting examples; this next one is particularly not obvious.

**Theorem 2.14** (Gottesmon-Knill)**.** *The gate set*

$$\left\{\mathrm{CNOT}, H, \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}\right\}$$

*is not only non-universal, but generates a discrete subgroup of* $\mathrm{U}(n)$*, and can be simulated by classical computation in polynomial time.*

This is surprising: these three gates alone do a lot of work, but to get a quantum speedup we need something else.

[9] This means if you pick a 1-qubit gate at random, the probability of it meeting this condition is 1.

[10] Pun intended?

Lecture 3.

## BQP: 9/19/16

Lecture 4.

## BQP in Complexity Theory: 9/26/16

Last time, we defined the complexity class BQP, which denotes the problems which can be solved by a quantum computer in polynomial time with small probability (say 1/3) of error. This fits into a hierarchy with other complexity classes.

- BPP is the classical analogue of BQP: the problems solvable by randomized polynomial-time algorithms with bounded error.
- #P is the counting problems associated to decision problems in NP: instead of asking "does an isomorphism of graphs exist," we ask "how many isomorphisms are there?"
- The *polynomial heirarchy* PH, defined with oracles from NP, etc.

There are some known relations between these classes: here $A \to B$ means $A$ is contained in $B$.

$$\begin{tikzcd}
\mathrm{P} \arrow[r] \arrow[dr] & \mathrm{NP} \arrow[r] & \mathrm{PH} \arrow[rr] & & \mathrm{P}^{\#\mathrm{P}} \arrow[r] & \mathrm{PSPACE} \\
& \mathrm{BPP} \arrow[r] & \mathrm{BQP} \arrow[r] & \mathrm{PP} \arrow[ur] & &
\end{tikzcd}$$

There are lots of things we don't know here. Possibly the most tantalizing is how effective quantum computers are.

**Question 4.1.** Is NP contained within BQP?

We think the answer is no, but don't expect to be able to prove it anytime soon. We seem far away from even proving conditional statements.

Zooming in, we have P and NP, with NP-complete appearing at the "top" (harder) of NP and P at the 'bottom." BQP is somewhere around the lower half, possibly containing the NP-complete problems, but probably not; in particular, it contains factoring and discrete-log, which are (probably) not in P.

**Theorem 4.2** (Lacher)**.** *If* P $\neq$ NP*, then there exist* NP-intermediate-*problems, i.e. problems that aren't* P *and aren't* NP*-complete.*

It's assumed that if BQP $\not\supset$ NP, then there should also be NP-intermediate problems not in BQP. One candidate problem is graph isomorphism, which was recently shown to be quasi-polynomial time, so isn't NP-complete, but not fast quantum algorithm is known. Lattice problems (especially finding the shortest nonzero vector in a given lattice) are NP, but not likely to be NP-complete, and no fast quantum algorithms are known. These lattice problems are the basis of some interesting and practical cryptographic algorithms, so are a focus of research in quantum computing.

Another important question:

**Question 4.3.** Is BQP contained in NP?

This relates to the problem of simulating a quantum circuit — it seems unlikely that one could do this quickly, but we don't know. So maybe it would be helpful to find a problem in BQP \ NP, but this generally requires something more general than languages.

**Definition 4.4.** A *promise problem* is a decision problem with two disjoint sets $\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}} \subset \{0,1\}^*$; given $x \in \Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$, decide whether $x \in \Pi_{\mathrm{YES}}$ or $x \in \Pi_{\mathrm{NO}}$.

Most problems in complexity theory fit into this paradigm, and arguably they'd be a better way to formulate complexity theory from the beginning. But they're particularly useful for quantum algorithms. For example, if you want to understand whether a given circuit $C$ accepts with probability greater than $1/2$, you could reformulate it as a promise problem, where you know it accepts with probability at least $1/2 + \varepsilon$ or $1/2 - \varepsilon$, and have to decide which. In fact, for $C$ a general quantum circuit, this problem is expected to be in BQP, but not NP.

We can generalize these even further.

**Definition 4.5.**

- A *relation problem* is one where there is a set of valid outputs, and the goal is to output any one of them.
- A *sampling problem* is one where the goal is to sample from a probability distribution, e.g. "sample from a satisfying state of this Boolean expression."

If we accept these more general paradigms, there's even stronger evidence that there are problems in BQP but not NP, but to say this we need to define what NP means for sampling problems, which is tricky. But this isn't evidence that BQP *sensu stricto* contains languages not in NP, but who cares? The point about the power of quantum computing doesn't seem completely model-dependent.

**Question 4.6.** Is BQP contained in the polynomial hierarchy?

This is an interesting contrast with the classical BPP: we don't know whether BPP is contained in NP or vice versa or neither or both, but the Sisper-Gács-Lautemann theorem shows that BPP $\subseteq \mathrm{NP}^{\mathrm{NP}}$, NP with an NP oracle, which is the second step in the polynomial hierarchy. This is because it's always possible to pull the randomness out of a random algorithm.

One thing we do have an example of is for sampling problems: there is a BQP-sampling problem $B$ that is not in the sampling class $\mathrm{BPP}^{\mathrm{PH}}$ — unless the polynomial hierarchy collapses, which is thought to be unlikely. There are a lot of complexity-theoretic statements which are predicated on the polynomial hierarchy not collapsing.

**Basic properties of BQP.** We defined BQP with regards to bounded error 1/3. Does this constant matter? In most, but not all,[11] cases, the precise number doesn't matter: you can just repeat the algorithm. For BQP, this is valid: the quantum computer can take multiple runs and collect the majority, just as in the classical case. This adjusting of the constant is called *amplification*.

The next question is oracle access. Oracles are a fundamental aspect of theoretical computer science, arising from Turing's thesis. You could think of them as fairies that magically solve certain problems, which sometimes led to ridicule, but there's another viewpoint: they're a way of capturing subroutines and understanding what part of a problem contains its difficulty. So, what does it mean for a quantum computer to make queries to an oracle? There are a few nuances.

We think of a black box as holding a function $f$, so if we feed it $x$, it outputs $f(x)$; alternatively, we could think of it holding a very long string, and we feed it an index $i$, and it returns the $i^{\text{th}}$ bit $x_i$. Both of these viewpoints are useful.

A quantum black box would take in quantum superpositions and map

$$\sum_i \alpha_i |i\rangle \longmapsto \sum_i \alpha_i |i\rangle |x_i\rangle. \tag{4.7}$$

But depending on what you mean by "oracle," this is tricky: you can't send a superposition over the Internet, so it would be very hard to use something like this in practice, but if you have access to the code for the black-box, you can feed it superpositions. If the black box is a file on your machine, the question is whether you have the still-theoretical technology of *quantum RAM* (QRAM). Thus, quantum oracles, useful in theory for breaking algorithms, aren't yet very worrisome in practice.

Since (4.7) is not unitary, we should more precisely write

$$\sum_{i,a} \alpha_{i,a} |i\rangle |a\rangle,$$

where $|a\rangle$ is an *answer register*. Then, we map

$$\sum_{i,a} \alpha_{i,a} |i\rangle |a\rangle \longmapsto \sum_{i,a} \alpha_{i,a} |i\rangle |\alpha \oplus x_i\rangle :$$

the black box writes the answer into the answer register. This is reversible, hence unitary. Sometimes, this is also generalized with *workspace qubits*, which pass though unchanged. Sometimes, it's helpful to also change the amplitudes in some cases, e.g. multiplying by $-1$. This helps for distinguishing all $x_i = 0$ from

[11] Example: if you want to detect an $\varepsilon$ bias on a loaded coin, you'll need $O(1/\varepsilon^2)$ flips.

all $x_i = 1$, since we square whenever we take measurements, and it would be good to be able to distinguish the 0 string from the 1 string! Thus, we could make a different type of query

$$\sum_{i,a,w} \alpha_{i,a,w} |i\rangle |a\rangle |w\rangle \longmapsto \sum_{i,a,w} \alpha_{i,a,w} (-1)^{x_i \cdot a} |i\rangle |a\rangle |w\rangle.$$

If the $x_i$ are boolean, these two types of queries are equivalent: going from the first type to the second, put the $a$ register in the $|-\rangle$ state, and in the other direction put $a$ in the $|+\rangle$ state. This allows us to deal with these two kinds of queries interchangeably for the rest of the course.

**Quantum algorithms.** Now that we know what a quantum query is, we can discuss some of the most famous quantum algorithms. These won't be the focus of this course, but one or two examples will be instructive.

**Example 4.8** (Deutsch-Jozsa algorithm)**.** This algorithm isn't the most amazing, but it was the first quantum algorithm! It computes the XOR of two bits $x_0$ and $x_1$ with one query, where classically you'd need two queries.

Start with two qubits in the zero state; apply a Hadamard gate to the first, take a (second-type) query measurement, and Hadamard the first qubit again and measure. This is a very common pattern.

In more depth, $|0\rangle \mapsto |+\rangle$ under the Hadamard gate, which is an equal superposition of $|0\rangle$ and $|1\rangle$. The phase query maps

$$|+\rangle \longmapsto \frac{(-1)^{x_0} + (-1)^{x_1}}{\sqrt{2}}.$$

If $x_0 = x_1 = 0$, this is $|+\rangle$, which the Hadamard gate maps to $|0\rangle$; if $x_0 = 0$ and $x_1 = 1$, we get $|-\rangle \mapsto |1\rangle$. If $x_0 = 1$ and $x_1 = 0$, we get $-|-\rangle \mapsto -|1\rangle$, which after measurement is indistinguishable from $|1\rangle$, and similarly if $x_0 = x_1 = 1$, we get $-|+\rangle \mapsto -|0\rangle$.

If we try to calculate the parity of $n$ bits, what kind of advantage do we get? Naïvely, you might expect to use a tree of gates and calculate the answer recursively, resulting in a logarithmic speedup, but this goes wrong: you end up with a little garbage left at the end, as when always happens when you feed the output of one quantum algorithm into another, and the way to fix this is to compute it twice. Thus, we obtain the recurrence relation $Q(n) \leq 2Q(n/2)$, which isn't very good. The top level doesn't need a second bit, so we obtain query complexity $n/2$. In fact, there's an easier way: split the bits into pairs, compute their parities in these pairs using Deutsch-Jozsa, and then classically XOR them together. Since we only made $n/2$ queries, this shows the quantum parity complexity of the parity problem on $n$ bits is at most $n/2$.[12]

Here's an esoteric-looking question.

**Question 4.9.** Is $\mathrm{BQP}^{\mathrm{BQP}} = \mathrm{BQP}$? In other words, if quantum problems can easily use quantum problems as subroutines, do we get any new problems?

By contrast, it's conjectured that $\mathrm{NP}^{\mathrm{NP}} \supsetneq \mathrm{NP}$, since you can encode two-quantifier statements in $\mathrm{NP}^{\mathrm{NP}}$. (We know $\mathrm{NP}^{\mathrm{NP}} \supset \mathrm{NP}$, but equality would imply the collapse of the polynomial hierarchy.) In general, a complexity class $C$ is *self-low* if $C^C = C$.[13] This is the first condition for a complexity class to be a model of physical reality.

For example, P is self-low and BPP is self-low. What about BQP? If $L \in \mathrm{BQP}$, we can make the query

$$\sum_x \alpha_x |x\rangle |a\rangle \longmapsto \sum_x \alpha_x |x\rangle |a \oplus L(x)\rangle.$$

The extra debris can be dealt with by uncomputing, but what about the error? You might worry that it stacks up, but thanks to amplification, this is not a problem; to be precise, you have to use the gentle measurement lemma and the quantum union bound. In particular:

**Theorem 4.10** (Bennett-Bernstein-Brassard-Vazirani (1997))**.** $\mathrm{BQP}^{\mathrm{BQP}} = \mathrm{BQP}$.

Everything works as long as you remember to amplify and uncompute.

You can't really use this to make progress on P vs. NP.

[12] In fact, the parity complexity is exactly $n/2$, even for algorithms with unbounded error!

[13] This is an iffy endeavor, not the least because quantifying over all complexity classes is a bit weird, but in the cases we consider, oracles will make sense.

**Theorem 4.11** (Baker-Gill-Solovay (1975))**.** P *and* NP admit contradictory relativizations*; that is, there are classes A and B such that* $\mathrm{P}^A = \mathrm{NP}^A$*, but* $\mathrm{P}^B \neq \mathrm{NP}^B$.

This is a little weird, but is very helpful: any proof which is to make progress on P vs. NP must depend on whether an oracle is present. All results borrowed from computability theory and logic aren't sensitive to the present of an oracle, so this was the first result to indicate that we needed fundamentally different techniques.

Note that $\mathrm{P} = \mathrm{NP}$ would not imply that $\mathrm{P}^A = \mathrm{NP}^A$ for all $A$: equipping P with an $A$-oracle is different than equipping NP with an $A$-oracle. Oracles act on *definitions* of complexity classes — even if P and NP were equal, they have different definitions.

For example:

- Currently, it's clear that Barack Obama is President of the United States.
- Therefore, Barack Obama = President of the United States.
- If Mitt Romney had won the election of 2012, he would've been the President of the United States.
- Hence, if Romney won in 2012, he would have been Barack Obama.

The issue is what your definition of "is" is.[14] However, in the polynomial hierarchy, it's possible to do some algebraic cancellations.

It's hard to make strong statements about the "spaces" of problems, but there are some results; it's possible to define random oracles, hence a probability distribution on the space of oracles.

**Theorem 4.12** (Bennett-Gill (1981))**.** $\Pr_A[\mathrm{P}^A \neq \mathrm{NP}^A] = 1$*, where A is a random oracle.*

This led to the *random oracle hypothesis*: that if a random oracle made two complexity classes distinct with probability 1, then they're equal. But this was quickly shown to be false: IP is the complexity class of problems solvable by an interactive proof in polynomial time. It's contained in PSPACE, but relative to a random oracle, with probability 1, $\mathrm{coNP}^A \neq \mathrm{IP}^A$. However, Shamir showed in 1990 that IP = PSPACE!

Let's talk about another quantum algorithm.

**Example 4.13** (Bernstein-Vazirani problem)**.** Suppose we're given a black box $f(x) = s \cdot x \bmod 2$, where $s \cdot x = s_1x_1 + \cdots + s_nx_n$ is the inner product over $\mathbb{F}_2$, and $s$ is some secret $n$-bit string. Our goal is to learn $s$ by querying $f$ with as few queries as possible. This is an example of a promise problem; we're promised that $f$ has this particular form. For promise problems, we'll see that the gap between classical and quantum complexity can be $n : 1$ for $n$ as large as you'd like.

Classically, you can use the basis vectors $e_1 = 100\cdots0$, $e_2 = 010\cdots0, \ldots, e_n = 0\cdots01$. Then, $f(e_i) = s_i$, so we've found $s$ with $n$ queries; since we need to find $n$ bits, this is also a lower bound.

However, there's a quantum algorithm that does this in one query! We make the phase query over all $n$-bit strings:

$$\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{s\cdot x}|x_1\cdots x_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{s_1x_1}|x_1\rangle(-1)^{s_2x_2}|x_2\rangle\cdots(-1)^{s_nx_n}|x_n\rangle.$$

After evaluating this, we get a $|+\rangle$ in place $i$ if $s_i = 1$, and $|-\rangle$ if $s_i = 0$, so after a Hadamard transformation we've determined $s$.

But depending on how you look at this, it's less impressive: we need $n$ qubits and make $n$ computations anyways. So what would be nice is to find a classical problem which needs more than polynomial queries, but which admits a quantum algorithm with only polynomial queries, or $n^{o(1)}$ vs. $n^{\omega(1)}$.

**Example 4.14** (Recursive Fourier sampling)**.** Similarly to the Bernstein-Vazirani problem, suppose we're given $f(x) = s \cdot x \bmod 2$ and we want to learn $h(s)$, where $h$ is pretty much anything except an inner-product function, e.g.

$$h(s) = \begin{cases} 1, & \text{if } |s| \equiv 0 \bmod 3 \\ 0, & \text{otherwise.} \end{cases}$$

Here, $|s|$ is the Hamming weight of $s$.

[14] This issue was first explicated by a different president.

But to make this more interesting, we don't provide $f(x)$ directly: instead, we require (in our best Dr. Evil voices) $f(x) = h(s_x)$, where $f_x(y) = s_x \cdot y \bmod 2$, and these could be predicated on further recursive levels.

If there are $d$ levels, the usual classical solution requires $\Theta(n^d)$ queries, and this is essentially optimal, as long as $h$ isn't degenerate. However, for the quantum algorithm, we get $O(2^d)$, because you have to uncompute garbage at all levels of recursion except for the first: the number of queries at level $k \in \{1, \dots, d\}$ is twice that of the previous one.

If $d = \log_2 n$, then $2^d = n$ and $n^d = n^{\log n}$; this was the first example where quantum algorithms provided a polynomial query complexity to a super-polynomial classical algorithm. This is nice, but still very contrived: nobody actually cares about this problem outside of complexity theory. (That said, it's also conjectured to not be in the polynomial hierarchy.) This was also used to show that there's an oracle $A$ such that $\mathrm{BPP}^A \neq \mathrm{BQP}^A$.[15]

This can also show that $\mathrm{BQP}^A \not\subset \mathrm{NP}^A$, and even $\mathrm{BQP}^A \not\subset \mathrm{MA}^A$.

**Definition 4.15.** The *Merlin-Arthur* complexity class MA is the class of languages $L$ on $\{0,1\}^*$ such that there is a polynomial-time verifier $M$ such that there's a witness $w$ $\Pr_r(M(x, w, r) \text{ accepts}) \geq 2/3$ if $x \in L$, and if $x \neq L$, for all witnesses $w$, $\Pr_r(M(x, w, r) \text{ accepts}) \leq 1/3$.

MA fits into the polynomial hierarchy $\mathrm{NP} \subset \mathrm{MA} \subset \mathrm{AM} \subset \Pi_2 = \mathrm{coNP}^{\mathrm{NP}}$; the last class is the second step in the polynomial hierarchy, and the third is the class of *Arthur-Merlin* problems, where the verifier is allowed randomness. There's a big conjecture in complexity theory that NP = MA = AM, even though they're quite different with respect to oracles.

It's also open whether there exists an oracle $A$ with $\mathrm{BQP}^A \not\subset \mathrm{AM}^A$, though not for a lack of trying. There are candidates, and if this is broken, it (probably/hopefully) captures much of the difficulty of the entire polynomial hierarchy.

**Example 4.16** (Simon's problem (1991))**.** Simon's algorithm is another oracle separation, but inspired Shor's algorithm, and contains the distilled essence of it without the number theory.

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a black-box Boolean function, and suppose we're promised that there exists a secret string $s \neq 0^n$ such that for all $x, y \in \{0,1\}^n$, $f(x) = f(y)$ iff $x \oplus y = s$ (here, $\oplus$ is bitwise XOR). This says a lot about $f$, e.g. that it's 2 : 1, and the witnesses differ by the *secret shift* $s$. The goal is to output $s$.

A decision-problem variant is to distinguish whether $f$ has this form, or is a one-to-one function.

One way to solve this is to randomly pick $x_1, x_2, x_3, \dots$ until $x_i$ and $x_j$ are found such that $f(x_i) = f(x_j)$, and then output $s = x_i \oplus x_j$. Because of the birthday paradox, the expected number of queries this algorithm makes is $2^{n/2}$. This is akin to hash-breaking attacks in cryptanalysis which exploit this fact, which are sometimes known as *birthday attacks*.

This algorithm is optimal: if $f$ is picked uniformly at random from the functions satisfying this constraint, any classical algorithm behaves like the one we just described on most $f$. This is an example of *Yau's minimax principle*: in game theory, it doesn't matter whether your opponent's strategy is deterministic or random; you may as well provide a strategy that works against all deterministic strategies. Similarly, if you want to prove a lower bound on a randomized algorithm, prove it for all deterministic algorithms that the randomized algorithm could model, and then model the distribution. That is, to prove a lower bound for a randomized algorithm, it suffices to pick a hard distribution over inputs and give a deterministic lower bound.

We can choose a nonzero string $s$ uniformly at random and then choose a random $f$ that hides $s$. Each query is made one by one, $x_1, \dots, x_k$ yielding $f(x_1), \dots, f(x_k)$. Suppose we haven't found any collisions; then, we can rule out $x_i \oplus x_j$ for all $i, j \in \{1, \dots, k\}$. If we're Bayesians, the posterior distribution over the remaining values of $s$ is still uniform, so to isolate $s$, we need $\binom{k}{2} \geq 2^n - 1$, or $k = \Omega(2^{n/2})$.

*Simon's algorithm* is a quantum algorithm for solving Simon's problem. The key is that making a query only is useful for what happens to the input register, irrespective of what we threw into the input register. So we want to compute the state

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

[15] Then again, any PSPACE-complete problem $B$ defines an oracle such that $\mathrm{BPP}^B = \mathrm{BQP}^B = \mathrm{PSPACE}$.

even though we'll actually discard $f(x)$. In the end, we end up with a state $(|x\rangle + |y\rangle)/\sqrt{2}$, where $x \oplus y = s$. In other words, we'd need to measure twice, and we could recover $s$... but that's not allowed in quantum mechanics.

Instead, we'll use one of our other tricks, and measure in the Hadamard basis: if I Hadamard all the qubits of $x$, I obtain

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

Why is this? We saw that this was the result of $H^{\otimes n}$ according to Bernstein-Varizani, and $H^{\otimes n}$ is its own inverse. Thus, if we apply $H^{\otimes n}$ to the Bell state in $x$ and $y$, we obtain

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

We'll only see this if there's nonzero amplitude, i.e. when $(-1)^{x\cdot z} = (-1)^{y\cdot z}$, i.e. $x \cdot z \equiv y \cdot z \bmod 2$ and $x \cdot z + y \cdot z \equiv y \cdot z + y \cdot z \equiv 0 \bmod 2$. Thus, we only see states $z$ such that $s \cdot z \equiv 0 \bmod 2$.

Next, we do it again. And again, and again, resulting in a system of randomly chosen linear equations $s \cdot z_1 \equiv 0$, $s \equiv z_2 \equiv 0, \ldots, s \equiv z_m = 0$, and Gaussian elimination is a polynomial-time algorithm to solve this once $m \approx n$.

Thus, we have a black-box problem where the classical query complexity is $\Theta(2^{n/2})$ and the quantum complexity is $O(n)$ (and one can prove this is optimal). This was the first example of bona fide exponential separation — and because it led to Shor's factoring algorithm and other results in cryptography, it's more important than just its formulation. And generalizing it to the hidden subgroup problem leads to discrete log, but almost graph isomorphism and lattice problems.

Lecture 5.

## The Hidden Subgroup Framework: 10/3/16

Last time, we discussed some quantum algorithms and what they had to say about quantum complexity theory. For example, we discussed recursive Fourier sampling as introduced by Bernstein-Vazirani and Simon's problem, which is exponential for classical algorithms but $\Theta(n)$ for a certain quantum algorithm. Each of these implies the existence of an oracle $A$ such that $\mathrm{BPP}^A \neq \mathrm{BQP}^A$.

These results, announced in the early 1990s, were the first concrete evidence that quantum computers are faster than classical computers. However, these problems have little immediate practical application: they're black-box algorithms. But if the hidden function $f$ in Simon's problem is useful...

For Simon's algorithm in particular, we want a function $f$ such that $f(x) = f(y)$ iff $x$ and $y$ differ by a specified phase. For example, we could let $f(x) = Ax$ where $A \in \mathbb{F}_2^{n\times n}$ is of rank $n-1$ — but if you know your function looked like this, you don't need a quantum algorithm to identify it efficiently. So the problem is not just about the oracle: there's something interior to the problem that makes it interesting. But for a real advantage, we'd want some sort of real-world problem for which quantum algorithms have a significant speedup, but for which there's no additional structure that makes it possible to explain this away.

This is where Shor's algorithm (1994) comes in. There's a general framework, called the *hidden subgroup framework*, for which Simon's and Shor's problems are two very specific cases, and creates a very wide paradigm for possible quantum algorithms we might hope to invent. These kinds of problems have been huge in quantum algorithms research in the past two decades — some might say too huge, but it's undeniable that hidden subgroup problems (HSPs) are important.

Let $G$ be a group, which will often be large but finite, and let $f : G \to \mathbb{N}$ be a function (more generally, we map to strings over an alphabet, finite graphs, or something else that can be indexed by $\mathbb{N}$ anyways). Suppose we have access to $f$ as a black box, and are promised that there exists a *hidden subgroup* $H \le G$ such that for all $x, y \in G$, $f(x) = f(y)$ iff $x$ and $y$ are in the same left coset of $H$, i.e. $xy^{-1} \in H$. We allow $H = 1$, which places the constraint that $f$ is injective, or $H = G$, so $f$ is constant, or something in between. The basic problem is to find $H$, e.g. producing a list of generators.

Some weaker but related problems are still important, e.g. deciding whether $H$ is trivial or not. This is the promise problem where $f$ is either one-to-one or many-to-one, and if the latter, its collisions have a specific structure.

This is a kind of weird problem to study, and nobody studied it before quantum computing, but different interesting problems can be fit into this format.

**Example 5.1.** Simon's problem (Example 4.16) is a hidden subgroup problem where $G = \mathbb{Z}/2^n$, which we may think of as strings of length $n$, where the multiplication is bitwise XOR. Specifically, the problem is to determine whether $|H| = 1$ or $|H| = 2$: if the latter, $H = \{0, s\}$, and the goal is to find $s$, which generates $H$.

In this case, $G$ is abelian, but not the only abelian group. Thinking like Peter Shor, is there an analogue of Simon's problem for $\mathbb{Z}/N$, or even for $\mathbb{Z}$? Subgroups of $\mathbb{Z}$ are multiples of some integer, and cosets are infinite arithmetic progressions: for $f : \mathbb{Z} \to \mathbb{N}$ to satisfy the hidden subgroup promise means that $f$ is periodic: there should exist an integer $s > 0$ such that for all $x, y \in \mathbb{Z}$, $f(x) = f(y)$ iff $s \mid (x - y)$; in particular, these are the only collisions.

**Example 5.2** (Period finding)**.** Period finding is a different special case of the hidden subgroup problem: given black-box access to an $f : \mathbb{Z} \to \mathbb{N}$ periodic with period $s$ (and not periodic with any period less than $s$), we want to find $s$.

There are problems of great interest contained in this algorithm: factoring and discrete-log reduce purely classically to period-finding. This was implicitly known, but nobody needed it or bothered to write it down before Peter Shor. The second interesting fact is that this problem is in $\mathrm{BQP}^f$, polynomial (or poly-log) in the number of bits in $s$. This makes constructing a quantum computer a rather more practical idea.[16]

We won't discuss Shor's algorithm in detail; it's covered in more detailed in, e.g. Vazirani's lecture notes. The point is to give enough of a sketch proof to convince the reader that factoring is in BQP. The reason that public-key cryptography is based in factoring uses number-theoretic structure in factoring that makes it interesting. This structure was known to Euler — it's not too fancy — but the same structure allows Shor's algorithm to work.

Why does factoring reduce to period finding? Suppose $N = pq$, and let $f_x(r) = x^r \bmod N$, where $x$ is coprime to $N$.[17] Then, $f_x$ is periodic, but its periodicity is governed by the multiplicative group $(\mathbb{Z}/N)^\times$ of things which are relatively prime to $N$.

**Theorem 5.3** (Euler)**.** *If $N = pq$, where $p$ and $q$ are prime, then $|(\mathbb{Z}/N)^\times| = (p-1)(q-1)$.*

For example, $(\mathbb{Z}/15)^\times = \{1, 2, 4, 7, 8, 11, 13, 14\}$, and this is indeed 8 elements.

In particular, if $\varphi(N) = (p-1)(q-1)$, then $x^{r+\varphi(N)} \bmod N \equiv x^r \bmod N$, since $x^{\varphi(N)} = 1 \in (\mathbb{Z}/N)^\times$ by Lagrange's theorem. That is, the period of $f$ divides $\varphi(N)$.

The next step is to try this with random $x$; after a few tries, their least common multiple is $\varphi(N)$ with overwhelming probability. This produces the factors: $\varphi(N) = N - p - q + 1$, so we know $p + q$ and $pq$, and therefore can solve for $p$ and $q$. This completes the reduction from factoring to period finding.

Lastly, why is period finding in BQP? The idea is the same as for Simon's problem, but the argument is a little hairier.

First, of all, it's not good to compute $x^r$ directly. It's much better to use *repeated squaring*, writing it as a product of $x^{2^k}$-terms, e.g.

$$x^{30} = (((x^2)^2)^2)^2((x^2)^2)^2(x^2)^2x^2,$$

and then calculate these. We want to learn things about the black-box function $f$, but just as for Simon's algorithm the fastest algorithm is a birthday attack: we check whether $f(x_i) = f(x_j)$ and then output $x_i - x_j$. This outputs a multiple of the true period, but doing this a few times, then computing the greatest common divisor (which is polynomial time, known at least a thousand years ago) after a few runs spits out the true period with high probability.

[16] Another fun fact is that if an adversary can corrupt a small number of bits per period, then Shor's algorithm still works very closely, thanks to the unitarity of quantum mechanics: if the original difference in states is off by $\varepsilon$, then we lose $O(\varepsilon)$ from the probability of success. This was studied in more generality by Hales and Hollgren.

[17] If $x$ isn't coprime to $N$, Euclid's algorithm allows us to find a common factor, which is therefore one of $p$ or $q$.

There are factoring algorithms that are better than brute-force., The *number field sieve* is an algorithm that has runtime about $O(\exp(n^{1/3}))$ where $n$ is the number of digits (assuming a smoothness conjecture in number theory). This is much faster than the naïve collision-finding algorithm, reducing factoring to a problem of elliptic curves.

But in the quantum world, we can imitate Simon's algorithm: let $Q$ be some number, at least exponential in $n$, and take the superposition

$$\frac{1}{\sqrt{Q}}\sum_{r=1}^{Q}|r\rangle|f(r)\rangle.$$

Because $f$ is computable in polynomial time, even classically, we can create a superposition like this (well, up to error correction, which is a separate problem).

Following Simon's algorithm, we measure the second register $|f(r)\rangle$ and then throw it away, caring only about its value on the first register, which will be a superposition of different preimages of the observed $f$, i.e. a superposition of $r$, $r+s$, $r+2s$, etc., with the appropriate normalization.

But since we didn't know what $r$ was, $|r+js\rangle$ doesn't tell us very much; we have to measure it in a different basis. For Simon's algorithm, the Hadamard gate sufficed, but not here; we want something which respects the structure of the group better, and we will use something called the quantum Fourier transform. In fact, the Hadamard base change in Simon's algorithm is an instance of this Fourier transform for $G = \mathbb{Z}/2^n$.

Imagine each signal is a wavefunction; we consider it in the basis with respect to time, where we measure what the signal is at each point. But we could also consider the frequency basis, asking what the different periodic components of the signal are. The Fourier transform is a unitary transform from the time domain to the frequency domain.

Specifically, with $Q$ as before and $\zeta = e^{2\pi i/Q}$ a primitive $Q^{\text{th}}$ root of unity, the *quantum Fourier transform* is the unitary map extending the assignment

$$|r\rangle \longmapsto \frac{1}{\sqrt{Q}}\sum_{j=0}^{q-1}\zeta^{rj}|j\rangle.$$

For example, if $Q = 5$, the QFT has the matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \zeta & \zeta^2 & \zeta^3 & \zeta^4 \\ 1 & \zeta^2 & \zeta^4 & \zeta & \zeta^3 \\ 1 & \zeta^3 & \zeta & \zeta^4 & \zeta^2 \\ 1 & \zeta^4 & \zeta^3 & \zeta^2 & \zeta \end{pmatrix}.$$

There's a nontrivial argument for showing that the quantum Fourier transform can be implemented with a small family of circuits with $n\log n$ gates; this is akin to the algorithm used in the (classical) fast Fourier transform, and has a huge speedup, but the catch is that in the quantum case, we get a quantum state rather than an explicit representation of the Fourier transform in memory.

Now, we measure in the standard basis, as in Simon's algorithm. What have we learned? There's some error analysis going on here, but there is a way to assemble the results of $O(1)$ measurements and determine the period with high probability. This uses another polynomial-time classical algorithm called the *continued-fraction algorithm*.

Discrete-log is similar, but more complicated.

⌣ · ⌢

Once again, Shor's algorithm applies the hidden subgroup problem to specific abelian group. This led people to wonder whether it's possible to solve the hidden subgroup problem in a similar way, at least for finite groups. Work of Alexei Kitaev (!) and others showed that if $G$ is a finite abelian group, then the hidden subgroup problem for $G$ is in $\mathrm{BQP}^f$. Perhaps this isn't a surprise, because all finite abelian groups are direct products of cyclic groups, but elliptic curve groups are finite abelian groups, which means generalizations of Shor's algorithm provide efficient solutions to discrete-log on elliptic curves as well, and therefore could break elliptic-curve cryptosystems as well. This was explicitly pointed out by Dan Boneh.

Worse (better?) yet, there are generalizations of Shor's algorithm for other infinite abelian groups, which can be used to break other cryptosystems, e.g. the Buckman-William cryptosystem.

People turned to nonabelian groups after this, with their eyes on the graph isomorphism problem: given two graphs $G$ and $H$, are they isomorphic? This is a well-known problem in NP $\cap$ coAM, meaning under a suitable derandomization hypothesis, it's in NP $\cap$ coNP. This means that, unless the polynomial hierarchy collapses, it's in NP but not NP-complete. Factoring, phrased as a decision problem, is also in NP $\cap$ coNP (because primality testing is in P). Like factoring, graph isomorphism appears to be a good target for quantum algorithms precisely because it's likely to not be NP-complete.

This year, there was a breakthrough in the graph isomorphism problem: Babai showed it's quasi-polynomial for a classical, deterministic algorithm (specifically, $n$ to the poly-log of $n$). This somewhat trivially tells us that it's in BQP.

There were some signs of this beforehand: it's very hard to come up with examples for which the graph isomorphism problem is difficult; in particular cases that come up in practice, there are nice approximations, and for random graphs, there are a bunch of invariants (e.g. spectrum, degree sequence) that distinguish them. There are a bunch of algorithms that *almost* always work.

It turns out there's a reduction from graph isomorphism to the hidden subgroup problem for the symmetric group $S_n$. Consider the question of determining the automorphism group of a graph (e.g. a graph is *rigid* if its automorphism garoup is trivial) — this is stronger than the graph isomorphism problem, because given two graphs $G$ and $H$, we can form the disjoint union $G \amalg H$ and ask if it has any automorphisms that switch $G$ and $H$!

Let $G$ be a fixed graph, $\sigma \in S_n$, and $f(\sigma)$ be the graph $\sigma(G)$, which reorders the indices of $G$. We can order the set of graphs to be indexed by $\mathbb{N}$, so $f : S_n \to \mathbb{N}$ is a function. This has $\mathrm{Aut}(G)$ as its hidden subgroup, because the permutations which don't change $G$ are exactly those in $\mathrm{Aut}(G)$.

There are some very deep principles in mathematics, such as the union bound or Markov's inequality. One of the others is that a finite group $G$ is generated by at most $\log_2|G|$ elements: this is because adding a generator to a finite group at least doubles its size: assuming $g_1 \neq e$, $|\langle g_1 \rangle| \geq 2$, and if $H = \langle g_1 \rangle \leq \langle g_1, g_2 \rangle = G$, then Lagrange's theorem claims that $|H|$ divides $|G|$; if they aren't exactly the same, the quotient must be at least 2. Thus, any subgroup is specified by polynomially many generators, making this problem well-posed.

Another important class of problems in NP is lattice problems based on the (presumed) hardness of the shortest vector problem; this is instrumental in post-quantum cryptography. Regev (2004) noticed that the shortest vector problem is approximately reducible (in a sense that we'll make precise) to the hidden subgroup problem for $D_{2n}$, the symmetries of the $n$-gon (rotations and reflections, which define an only slightly nonabelian group of order $2n$). This suggests that extending Simon's algorithm to nonabelian groups will be hard. Similar reductions exist for other lattice problems, albeit with different approximation parameters. There are even negative results that certain things that do work in the abelian case don't work in the nonabelian case, or even results showing that to do this for $D_{2n}$ requires either a completely novel algorithm or the subset-sum problem.

There are certain nonabelian groups for which the hidden subgroup problems are efficiently solvable, in all cases very close to abelian groups in ways admitting approximation. One family of examples is the *Heisenberg groups*

$$\left\{ \begin{pmatrix} 1 & a & b \\ 0 & 1 & c \\ 0 & 0 & 1 \end{pmatrix}, a, b, c \in \mathbb{F}_p \right\}$$

for some finite field $\mathbb{F}_p$.

There is a striking observation, however, which can be interpreted as saying quantum computers "almost" solve the hidden subgroup problem for nonabelian $G$.

**Theorem 5.4** (Ettinger-Høyer-Knill 1997). *For all finite $G$, there exists a quantum algorithm to solve the hidden subgroup problem with group $G$ using only* polylog $\cdot|G|$ *queries to the oracle $f$.*

There are stronger statements for specfic groups, including $D_{2n}$: there are efficient measurements that provide classical information describing the answer.

The catch: this is very computationally inefficient, and provides an example where query complexity doesn't tell the entire story.

*Proof.* Let $f : G \to \mathbb{N}$ be the function with the hidden subgroup, and as usual form the state

$$\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |f(x)\rangle.$$

Then, measure the second register, which produces a random coset of $H$:

$$|Hg\rangle = \frac{1}{\sqrt{H}} \sum_{h \in H} |hg\rangle.$$

That is, one obtains a mixed state

$$\rho_H = \mathbb{E}_{g \in G}[|Hg\rangle\langle Hg|].$$

If $\rho_H$ doesn't determine $H$, then we can repeat this several times, forming $\rho_H^{\otimes k}$, where $k$ is polylog$|G|$.

The key claim is that $\rho_H^{\otimes k}$ information-theoretically determines $H$. That is, there's some measurement, maybe not efficient, but exists. This is called the *coset sampling approach*.

One ingredient in this is that if $G$ is finite, it has at most $|G|^{\log|G|}$ subgroups, because every subgroup of $G$ can be specified by at most $\log_2|G|$ generators (this bound is sharp when $G = \mathbb{Z}/2^n$). Thus, we can order the subgroups $H_1, \dots, H_r$ of $G$ in decreasing order of size; for each subgroup, we will measure $\rho_H^{\otimes k}$ and see if it's what we've found. This is extremely wasteful, but it suffices!

For each subgroup $H_i$, we apply the measurement $M_i$ which projects onto the subspace span$\{|gH_i\rangle : g \in G\}$. If $H = H_i$, then the measurement accepts (produces the same state as $\rho_H$), as $\rho_H$ is an equal mixture of the given basis for this subspace. In fact, it will accept any $H$ containing $H_i$. But if $H \not\supseteq H_i$, then with overwhelming probability, this measurement rejects: in this case, $M_i^{\otimes k}$ accepts with probability at most $1/2^k$, because $M_i$ accepts $\rho_{H_i}$ with probability at most $1/2$, ultimately because $|\langle H_i|H\rangle| \leq 1/\sqrt{2}$, and therefore squaring it won't get us over $1/2$.

Finally, how large do we need to choose $k$? The gentle measurement lemma tells us that each time the state rejects, $\rho_H^{\otimes k}$ is changed by at most $1/2^k$ in the trace distance. Thus, by the quantum union bound, it's okay to repeat this $O(\sqrt{2^k})$ times. Thus, we just need to set $\sqrt{2^k} = |G|^{\log|G|}$, i.e. $k = \log^2|G|$, which is polylogarithmic as desired. ☒

Shor's algorithm and its generalizations to abelian and non-abelian groups are among the most famous quantum algorithms. They provide huge speedups to the classical case of the hidden subgroup problem, but demonstrate that query complexity isn't everything.

The second tremendously important family of quantum algorithms is Grover's algorithm and its family. Though this isn't a quantum algorithms course, algorithms are important as an obstruction to quantum impossibility proofs.

Grover's algorithm can be thought of as solving a different black-box problem (that this and the hidden subgroup problem arise as black-box problems are probably why quantum researchers like black-box problems), and obtains a polynomial speedup (which cna be proven to be optimal) that's useful in a wide variety of applications.

Let $f : \{1, \dots, N\} \to \{0, 1\}$ be a function, which we'll take as a black box.

- Suppose we're promised there's an $i$ such that $f(i) = 1$ (a *marked item* or *marked element*). Find such an $i$.
- A decision-version variant: decide whether there's an $i$ such that $f(i) = 1$.

These problems are mutually reducible: the decision problem can be used to binary-search $\{0, \dots, N\}$, and one can check if the search problem succeeds to power the decision problem.

You can think of this as a problem of looking things up in a database with some physical representation, e.g. in a quantum computer in some kind of superposition. Alternatively, $f$ could be a checking function for some combinatorial problem, e.g. an optimization problem (binary-search over all possible cutoffs, and let $f(x) = 1$ if the cost of $x$ is less than the cutoff) or an NP-complete problem.

The classical query complexity of the Grover problem is $\Theta(N)$: there is no catch. This is still true for randomized algorithms.

Grover produces a quantum algorithm with query complexity $O(\sqrt{N})$, and unlike with Ettinger-Høyer-Knill's theorem, the number of quantum gates is $O(\sqrt{N}\log N)$, which is reasonable, and uses only $O(\log N)$ qubits.

Just as Shor's algorithm was an instance of the hidden subgroup problem, Grover's algorithm is an instance of the amplitude amplification problem. Suppose $|v\rangle$ and $|w\rangle$ aren't quite orthogonal, in that $\langle v|w\rangle \geq \varepsilon$. Suppose also we're given two unitary transformations as black boxes: $U_v = I - 2|v\rangle\langle v|$, which reflects across the line $\mathrm{span}\{v\}$; and $U_w = I - 2|w\rangle\langle w|$, reflection about $w$.

**Theorem 5.5.** *It is possible to convert $|v\rangle$ into a state arbitrarily close to $|w\rangle$ with only $1/\varepsilon$ oracle calls to $U_v$ and $U_w$.*

We think of $w$ as unknown; in most of the cases we care about, $v$ is known, but even if it were unknown, this would still work.

The claim is that repeating $U_vU_wU_vU_w\cdots$ for $1/\varepsilon$ times works. Drawing a picture helps: all the action in this algorithm is confined to the two-dimensional subpace $\mathrm{span}\{|v\rangle, |w\rangle\}$: $U_wU_v|v\rangle$ triples the angle between $|v\rangle$ and $|w\rangle^\perp$; the next time we get five times the original angle, and so on, until $\Theta(1/\theta) = \Theta(1/\varepsilon)$ iterations, when the state is as close as it can be to $|w\rangle$.

This algorithm has an unusual property not found in most classical algorithms: it gets closer after iterations, then farther again, then closer, then farther, as it oscillates around the origin. Like a soufflé, you have to take it out of the oven at the right time; you might not exactly know $\langle v|w\rangle$, in which case you have to try different numbers of trials to approximate the inner product, and hence approximate $w$. There are other modifications, e.g. fixed-point Grover search (Tulis-Grover-Patel, 2005), where you also take measurements at each step, which allows the algorithm to converge monotonically to $w$.

The Grover problem is a specific instance of this in which

$$\begin{aligned}|v\rangle &= \frac{1}{\sqrt{N}}(|1\rangle + \cdots + |N\rangle)\\ |w\rangle &= |j\rangle.\end{aligned}$$

Thus, $|\langle v|w\rangle| = 1/\sqrt{N}$. We have access to $f = U_{|j\rangle}$ as a black box. The algorithm's circuit is iterations of Hadamards and alternatively using $U_0$ and $f$. The success probability follows the sinusoidal problem, and therefore could be quite dramatic.

Lecture 6.

## Quantum Query Complexity and Lower Bounds: 10/10/16

*"With enough effort, any of us can do math."*

Today, we'll discuss quantum query complexity, and some methods for establishing lower bounds. Last time, we discussed Shor's algorithm in the context of the hidden subgroup problem for $\mathbb{Z}$. For finite nonabelian groups, the answer is much harder: there's a tantalizing result of Ettinger-Høyer-Knill that says it's always possible to almost solve the hidden subgroup problem in polynomial time, but for general groups very little is known. Finally, we discussed Grover's algorithm.

Recall that Grover's algorithm heuristically told us that we can approximate a hidden state if we know how to reflect across it and across something nearly orthogonal to it. The complexity is $O(\sqrt{N})$, and the square root comes from the 2-norm that quantum mechanics leans on — in the classical case, you might think about 1-norms of probability vectors, but for quantum computing, we think about amplitudes. Grover's algorithm is a sequence of queries and unitary transformations: the amplitude $t/\sqrt{n}$ on the marked item corresponds to the probability $t^2/n$: because quantum mechanics preserves the 2-norm, it opens up the possibility of finding the marked item grows quadratically in the number of queries, and Grover's algorithm realizes that possibility.

There are many variations on Grover's algorithm known today, a whole paradigm of repeatedly iterating in order to concentrate more and more amplitude on the correct answer. This applies to lots of tasks, but doesn't always give as much as a $\sqrt{N}$ speedup.

This is pretty nice. But is Grover's algorithm optimal? Just how efficient can a search be? The mathematical model of query complexity is a good place to prove optimality. This question was asked very early on, and the discovered answer is that, yes, Grover's algorithm is optimal for searching an unordered list.

**Definition 6.1.** Let $f$ be a function. Then, $Q(f)$, the *bounded-error quantum querty complexity* of $f$, the minimal number of queries made by a quantum algorithm $A$ for $f$ such that for all inputs $x$,

$$|\Pr[A(x) \text{ accepts}] - f(x)| \leq \frac{1}{3}.$$

**Theorem 6.2** (BBBV 1994)**.** *Any quantum searching algorithm for an unordered list requires $\Omega(\sqrt{n})$ steps. Specifically, if $\mathrm{OR}_n$ denotes the n-bit OR function, then $Q(\mathrm{OR}_n) = \Omega(\sqrt{n})$.*

For a fun historical note, this bound was established several years before Grover's algorithm was discovered! It was surprising that the bound was sharp. In this language, Grover showed $Q(\mathrm{OR}_n) = \Theta(n)$.

**Corollary 6.3.** *There exists an oracle $A$ such that $\mathrm{NP}^A \not\subset \mathrm{BQP}^A$.*

*Proof.* Let $A$ encode an oracle which encodes an exponential search problem, and consider the problem of finding an $x \in \{0,1\}^n$ such that $A(x) = 1$. Then, $A$ has an NP witness, but any quantum algorithm must solve an unordered search, hence by BBBV must take time $\Omega(\sqrt{2^n}) = \Omega(2^{n/2})$. ☒

Theorem 6.2 was the first known limitation on quantum algorithms. There are several known proofs, each of which comes with a different useful insight, but BBBV proved it in about the most direct way possible: the central fact they exploit is that quantum mechanics is linear.

*Proof of Theorem 6.2.* Consider a Grover search algorithm on $n$ inputs $x_1 \cdots x_n$, where we want to find an $i$ such that $x_i = 1$, and suppose we have a $T$-query quantum algorithm for it. First, we make a test run, setting all $x_i$ to 0; let

$$|\psi_t\rangle = \sum_{i,z} \alpha_{i,z,t} |i, z\rangle$$

denote the state after running this algorithm. Define

$$\alpha_{i,t} = \sqrt{\sum_z |\alpha_{i,z,t}|^2},$$

and define the *query magnitude of i* to be

$$m_i = \sum_{t=1}^{T} |\alpha_{i,t}|^2.$$

This represents the total magnitude with which index $i$ gets queried. This implies that

$$\sum_{i=1}^{n} m_i = \sum_{i=1}^{n} \sum_{t=1}^{T} |\alpha_{i,t}|^2 = \sum_{i=1}^{T} 1 = T.$$

Thus, there's a $j \in [n]$ such that[18] $m_j \leq T/n$.[19]

This allows us to fool the algorithm: if we flip the $j^{\text{th}}$ bit, it can't notice that much. Since

$$\sum_{t=1}^{T} |\alpha_{j,t}|^2 \leq \frac{T}{n},$$

then we can use the Cauchy-Schwarz inequality to conclude that

$$\sum_{t=1}^{T} |\alpha_{j,t}| \leq \frac{T}{\sqrt{n}}.$$

Now, let's modify the test run by flipping $x_j$ from 0 to 1 in the final query (number $T$). The final quantum state of the algorithm may differ by at most $2\alpha_{j,T}$, because before this change, everything else was absolutely the same.

[18] Here and everywhere, $[n]$ denotes the set $\{1, \ldots, n\}$.

[19] This is sometimes called the *Lake Wobegon principle*, in honor of the fictional Midwest town where all the children are above average!

Now, we can change the penultimate result of the $j^{\text{th}}$ bit, so the final state is changed by at most $2(\alpha_{j,T} + \alpha_{j,T-1})$, then the third state, and so on, until we've changed all observations of state $j$. Since errors have to add up linearly, this sum is at most

$$2\sum_{t=1}^{T}|\alpha_{j,t}| \leq \frac{2T}{\sqrt{n}}.$$

We conclude that if $T = o(\sqrt{n})$, then the distance between the unmarked item and the marked item is $o(1)$, so there's no way to distinguish them with $2/3$ probability of success. Indeed, a more refied algorithm shows that after $T$ queries, the probability of finding the marked item is $O(T^2/n)$, so Grover's algorithm is indeed optimal. ☒

This was a worst-case analysis, but may be bootstrapped to an average-case analysis: a Markov-like bound (the "strong Lake Wobegon principle") may be used to show that most indices aren't very far above average.

Those of us who like searching algorithms know that OR isn't the only function that's worth studying. For example, we can ask for the *parity* or *Hamming weight* $|x|$ of a string $x$, the sum of the bits mod 2; this is our next test case. This has more complicated behavior, and its quantum complexity is $Q(P_n) \leq n/2$ by the Deutsch-Josza algorithm (Example 4.8); moreover, the proof of Theorem 6.2 provides a lower bound of $\Omega(\sqrt{n})$, because to distinguish all 0s from all 0s and a 1 is a subcase of computing parity.

There is still a quadratic gap between these bounds, and we'll be able to translate it using a strikingly different method of proof which transforms problems into questions about polynomials. This is an old idea, going back at least to Minsky-Papert's *Perceptrons*. The conceptual part used polynomial methods to place bounds on the effectiveness of neural networks, and quantum computation researchers repurposed it to place bounds on the effectiveness of quantum algorithms. We'll be able to provide a sharp bound for parity as well as another proof for the effectiveness of Grover's algorith.

Though Razborov and Smolensky used these methods in circuit complexity, the first use of polynomial methods in quantum computation was by Beals-Buhrman-Cleve-Mosca-de Wolf (1998).[20]

**Lemma 6.4** (Beals et. al.). *Let $A$ be a quantum algorithm that makes $T$ queries to $x_1, \ldots, x_n \in \{0,1\}^n$. Then,* $\Pr[A(x) \text{ accepts}]$ *is a polynomial in $x_1, \ldots, x_n$ of degree at most $2T$.*

*Proof.* We'll show that if $A$ has made $t$ queries, each amplitude $\alpha_{i,z,t}$ is a polynomial in $X_1, \ldots, X_n$ of degree at most $t$. This suffices, because to calculate the probability that $A$ accepts, we sum the squares of the magnitudes of these amplitudes, hence obtain polynomials of degree at most $2t$. This requires getting rid of the absolute value signs, but there's a technical lemma that shows by adding a constant number of gates and no queries, all magnitudes may be made real (this is an encoding trick using an embedding $\mathrm{U}(n) \hookrightarrow \mathrm{O}(N)$); alternatively, we can show it for the real parts and the imaginary parts.

To prove the claim, we induct on the number of queries. At $T = 0$, the amplitudes are polynomials of degree 0, since they're just numbers. What if we make a query? The state looks like

$$\sum_{i,z}\alpha_{i,z}|i,z\rangle,$$

where by the inductive assumption $\alpha_{i,z}$ is a degree $T$ polynomial in $x_1, \ldots, x_n$. If this is in phase, the next query multiplies by $(-1)^{x_i} = (1 - 2x_i)$, so we obtain $(1 - 2x_i)\alpha_{i,z}$, which is a degree $(t+1)$-degree polynomial. After this, there may be a unitary transformation, but this doesn't depend on $x$, and therefore is a linear function of the magnitudes; thus, it sends polynomials to polynomials. ☒

This is an extremely useful fact. This still works if the algorithm does some classical post-processing, thanks to the deferred measurement lemma.

Another way to think of this is through the approximate degree of a function.

**Definition 6.5.** If $f$ is a Boolean function, its *approximate degree* $\widetilde{\deg}(f)$ is the minimum degree of a polynomial $p : \mathbb{R}^n \to \mathbb{R}$ such that $|p(x) - f(x)| \leq 1/3$.

[20] See https://arxiv.org/abs/quant-ph/9802049 for the paper.

Then, Lemma 6.4 says that $\widetilde{\deg}(f) \le 2Q(f)$. Alternatively, $Q(f) \ge \widetilde{\deg}(f)/2$. The right-hand side is a purely mathematical animal, and tells us that to find a lower bound on quantum query complexity, we just need to know how to well approximate a function by a low-degree polynomial.

*Remark.* The 2-norm shows up a lot in quantum computation, and ultimately depends on the fact that quantum mechanics itself is based on the 2-norm. Why is this? Experiments in physics bear it out, of course, but can we derive a theoretical or computational answer?

What would go wrong if we tried to formulate a theory conserving the 3-norm or the 4-norm? Ultimately the Pythagorean theorem picks out the 2-norm as fundamental, but even if you search for exotic geometry there are other options (though none have an inner product). For anything resembling quantum mechanics, we want a large amount of linear operations preserving the norm. For example, *stochastic matrices* are those that preserve the 1-norm (on nonnegative vectors), and in this sense the 1-norm underlies classical statistics. The 2-norm is preserved by orthogonal matrices, and underlies quantum mechanics.

**Theorem 6.6.** *For the* 3*-norm, or indeed for any $p \neq 0, 2$, the only linear transformations (on a finite-dimensional space) that preserve the $p$-norm are products of permutations and reflections.*

So there's not really enough to do anything resembling quantum mechanics.

A heuristic way to think about this theorem is that conserving the 2-norm is an $O(n^2)$-dimensional constraint, but conserving the 4-norm is $O(n^4)$-dimensional, which is overdetermined. This isn't a rigorous proof, but it makes the result less surprising.

Theorem 6.6 is sort of a statement about physics; in quantum field theory, people consider functions on spaces of fields, but really they consider functions that can be well-approximated by Fourier series, and therefore there's a countable basis where the theorem is more applicable. Similarly, in other kinds of physics, the Hilbert space of states can be reduced to countably infinite dimension, and if you believe the Beckenstein bound in quantum gravity, maybe even finite-dimensional.

So we can't speak for all norms on all vector spaces, but among the $p$-norms, we know why the 2-norm is special.

∽ · ∼

Returning to quantum computation, we've reduced to the still somewhat hairy problem of determining $\widetilde{\deg}(f)$ for a function $f$. This is about polynomials in $n$ variables, but can we reduce to polynomials in 1 variable?

**Lemma 6.7** (Symmetrization lemma., Minsky-Papert 1968)**.** *Let $p$ be a multilinear polynomial*[21] *in $n$ variables and let $q : \mathbb{N} \to \mathbb{R}$ be defined by*

$$q(k) = \mathbb{E}_{|X|=k}[p(X)].$$

*Then, $q$ is a polynomial in $k$ of degree at most* $\deg(p)$.

*Proof.* We can write

$$p(X) = \sum_{|S| \le \deg(p)} \alpha_S \prod_{i \in S} x_i.$$

Now, we'll use one of the great mystical facts about the universe, linearity of expectation:

$$q(k) = \mathbb{E}_{|X|=k}[p(X)] = \sum_{|S| \le k} \alpha_S \underbrace{\mathbb{E}_{|X|=k}\left[\prod_{i \in S} x_i\right]}_{p_S(k)}. \tag{6.8}$$

This is a purely combinatorial problem: over all inputs of Hamming weight $k$, how many inputs are 1 on all of $S$? There are $\binom{n}{k}$ vectors with Hamming weight $k$, and $\binom{n-|S|}{k-|S|}$ ways to make the product equal to 1

[21] A polynomial is *multilinear* if no term has any variable with a power higher than 1. This is an appropriate assumption for our situation, since we're working with Boolean functions, $0^2 = 0$, and $1^2 = 1$.

(the entries indexed by $S$ have to be 1, and the rest can be anything). Now, we compute:

$$\begin{aligned}
p_S(k) = \frac{\binom{n-|S|}{k-|S|}}{\binom{n}{k}} &= \frac{\dfrac{(n-|S|)!}{(n-k)!(k-|S|)!}}{\dfrac{n!}{k!(n-k)!}} \\
&= \frac{(n-|S|)!}{n!}\frac{k!}{(k-|S|)!} \\
&= k(k-1)(k-2)\cdots(k-|S|+1).
\end{aligned}$$

Thus, $p_S(k)$ is a polynomial of degree at most $d$ in $k$, and therefore by (6.8), $q(k)$ is too. ☒

We apply this first to the parity problem $P_n$; suppose $q$ is the univariate polynomial representing the expected value of the multivariate problem of approximating $P_n$. Since the parity of the zero string is 0, $q(0) \leq 1/3$, but the parity of the 1-string is expected to be 1, $q(1) \geq 2/3$; similarly, $q(2) \leq 1/3$, $q(4) \geq 2/3$, and so on. Thus, $q$ switches from increasing to decreasing at least $n$ times, so $q'(x) = 0$ at least $n$ times, so $\widetilde{\deg}(P_n) \geq \deg(q) \geq \deg(q') \geq n$, and therefore $Q(P_n) \geq n/2$ and Deutsch-Josza is optimal.

The polynomial method, like any quantum lower-bound method worth its salt, can provide a lower bound for Grover's algorithm, meaning a lower bound on the approximate degree of the $n$-bit OR function equal to $O(\sqrt{n})$.

This is already a more sophisticated use of the polynomial method: we know $q(0) \leq 1/3$ and $q(k) \geq 2/3$ when $k \geq 1$. This isn't a lot of information, but the key is a shallow fact: $q(k)$ is a probability for $k \in [n]$, hence is always between $[0,1]$ on those points. Markov investigated polynomials with these constraints in 1887, after a request from Dmitri Mendeleev. The polynomials that have the largest derivative (at a point) while fitting in a $1 \times 1$ box are the *Chebyshev polynomials*

$$T_d(x) = \cos(d \arccos x).$$

**Exercise 6.9.** Show that the Chebyshev polynomials are actually polynomials in $x$. Hint: use $e^{ix} = \cos x + i \sin x$.

These polynomials oscillate a lot, with their oscillations increasing near the boundary of the cube. The Chebyshev polynomial of degree $d$ has maximal first derivative $d^2$, and this is optimal.

We restate this theorem in an equivalent way that's more useful for us.

**Theorem 6.10** (Markov's inequality)**.** *Let $p$ be a polynomial such that on an interval $[x_0, x_0 + w]$, $p(x)$ is bounded in a box of height $h$, and suppose there is an $x \in [x_0, x_0 + w]$ such that $p'(x) = \Delta$; then,*

$$\deg(p) \geq \sqrt{\frac{w\Delta}{h}}.$$

The proof isn't hard, but is outside the scope of the course; the result we use it for was known before Beals et. al.'s work.

**Theorem 6.11** (Nisan-Szegedy 1994)**.** $\widetilde{\deg}(\mathrm{OR}_n) = \Omega(\sqrt{n})$.

*Proof.* We know $q(0) \leq 1/3$ and $q(1) \geq 2/3$, so by the mean value theorem, there's an $x \in [0,1]$ such that $q'(x) \geq 1/3$. Thus, we have $\Delta = 1/3$ and $w = n$. To calculate $h$, we need to be clever: we've so far only constrained $h$ on integers from 0 to $n$, as it is only interpreted as a probability on these integers. However, the larger the maximum is, the larger the derivative has to be, and these constraints will cancel each other out.

Suppose the minimal bounding box has height $h$; then, since $q$ is at most 1 on the integers, the mean value theorem tells us that $\Delta \geq 2(h/2 - 1) = h - 2$. Thus, by Theorem 6.10,

$$\deg(q) \geq \sqrt{\frac{n \max\{1/3, h-2\}}{h}}.$$

This is $\Omega(\sqrt{n})$, as desired. ☒

Symmetrization is useful for symmetric boolean functions, but things are harder in general. Consider the problem of determining whether an $n \times n$ bitmap has a row of all 1s (ANDs followed by an OR). One solution is to recursively call Grover's algorithm; there are a few subtleties to making this work (why is the error small?), but ultimately amplification of the inner search will push the error probability down to $1/n$, which takes $\log n$ repetitions. Thus, this algorithm has query complexity $O(\sqrt{n} \log n)$, which is an upper bound for the problem. There are finer upper bounds, in particular hacking away at the $\log n$ by cleverly redistributing the amplitude — de Wolf obtained the bound $O(\sqrt{n} \log^* n)$; the $\log^*$ function is the number of repeated applications of log that get a number down to 1; then, another result brought it down to $O(\sqrt{n})$.

There's an $n^{1/4}$ lower bound coming from a single row's AND (or from the top level). One might want to multiply them together, but that's not how it works. It is true, however, that the approximate degree of this problem is $\Theta(\sqrt{n})$ (Aaronson-Shi-Ambainis got $\Omega(n^{1/3})$, and Sherstov got $\Omega(n^{2/5})$ and eventually $\Omega(\sqrt{n})$). Some of these proofs involve fancier reductions from the multivariate to the univariate case, e.g. using linear programming and the dual problem. Generalizations of this are well-understood, thanks to an understanding about the quantum query complexity of a composition of functions.

In 2000, Andris Ambainis invented a third way to obtain quantum lower bounds, now called the *quantum adversary method*. This is another important tool in this subject, and is more broadly applicable, even in a sense universal, than the polynomial method. It's a kind of grown-up version of the BBBV lower bound. This method was first used on the OR-ANDs problem we just discussed, where it provides a sharp bound, and has since been employed on hundreds of other lower bounds.

The first idea is that any algorithm that works on one input must also work on a superposition of inputs, and will return a superposition of answers. This means there was a lot of entanglement between the input registers and the workspace for the function, and this is an entanglement of a specific kind, of the values going to 0 with 0, and those going to 1 with 1.

One can look at the density matrix of the inputs: each entanglement lowers the values of the off-diagonal entries, and by the end, they should be close to zero. The key idea: how much can a single query decrease these off-diagonal etnries by? This allows one to prove that if an algorithm needs to decrease this by a large amount, it must make lots of queries.

**Theorem 6.12** (Ambainis)**.** *Let $f : \{0,1\}^n \to \{0,1\}$, $X$ be the subset of 0-inputs, and $Y$ be the set of 1-inputs (i.e. $x \in X$ if $f(x) = 0$, and $x \in Y$ if $f(x) = 1$). Let $R \subset X \times Y$ be a relation such that*

(1) *for all $x \in X$, there are at least $m$ neighbors $y \in Y$ such that $(x, y) \in R$;*
(2) *for all $y \in Y$, there are at least $m'$ neighbors $x \in X$ such that $(x, y) \in R$;*
(3) *for all $x \in X$ and $i \in [n]$, there are at most $s$ $y \in Y$ such that $(x, y) \in R$ and $x_i \neq y_i$; and*
(4) *for all $y \in Y$ and $i \in [n]$, there are at most $s'$ $x \in X$ such that $(x, y)$ and $x_i \neq y_i$.*

*Then,*

$$Q(f) = \Omega\left(\sqrt{\frac{mm'}{ss'}}\right).$$

How do we use this? Well, let $X$ be the set of 0-inputs and $Y$ be the set of 1-inputs, and define $(x, y) \in R$ if $x$ and $y$ have edit distance 1. Then, $m = n$, $m' = 1$, $s = 1$, and $s' = 1$, so the query complexity is $\Omega(\sqrt{n})$.