18.435: Quantum Computing

Jakin Ng

Fall 2023

Contents

1 Introduction

1.1 Course Information

This class is taught by **Aram Harrow**, and these notes are taken by **Jakin Ng**. The grade breakdown is 40% weekly problem sets, 20% midterm on October 27, and 40% final. There will be office hours on Monday/Wednesday from 2-3 and on Thursday 9:30 - 11.

Sources relevant to the class include

- Harrow's notes
- Shor's notes
- The book QCQI
- Preskill's notes

This class is listed as 2.111/6.6410/8.370/18.435 and will draw from math, including probability and linear algebra, physics, including quantum mechanics, and CS, including algorithms, discrete math, and complexity theory.

The plan for the course will be to cover the basics of QI, algorithms, information theory, and error correction.

1.2 Introduction to Quantum Computation

In classical computing, bits are either 0 or 1, and we can use the gates AND, OR, NOT, and FANOUT to compute all functions that we call **computable**. It turns out that most functions take an exponential time to compute.

1.3 Computational Complexity

Intuitively, there are simply a lot of functions, and so most of them will take a long time to compute.

Question. *How many functions are there from* $\{0,1\}^n \longrightarrow \{0,1\}$*?*

Answer. *There are* 2^{2^n} *possible functions, which comes from writing out the function. There are* 2^n *inputs in* $\{0,1\}^n$*, and for each input, there are 2 choices, 0 or 1, of what the function returns.*

Computational complexity refers to the amount of time it takes to compute a problem, as a function of the input size.

Example 1.1
What is the computational complexity of multiplication of n-bit numbers, say a and b? This clearly depends on which algorithm or method one uses to compute the product! For example, using the algorithm which is "add a to itself b times" will take a very long time. Another algorithm would be , which would be $O(n^2)$ time.

In general, if the computation time runs in polynomial time relative to the input, we will consider this to be quick. Most generic random functions will be exponential time, but thankfully most functions we care about have some structure to them, and thus will be polynomial time, or **tractable**.

Proposition 1.2 (Church-Turing Thesis)
Changing the computational model or computer architectures used to compute an algorithm changes the computational complexity by at most a polynomial factor. This means that "all computational models" have essentially the same computational power, so the specific computer or computer architecture used is not super important.

More specifically, we might ask *how* the computational complexity changes with each computational model – after all, computers have changed significantly since 30 years ago. The Church-Turing thesis has survived the following challenges:

- random-access memory vs. a tape

- parallelism
- reversible computing
- analog computing – theoretically, this allows you to go beyond Turing machines, but practically, due to noise, circuitry can only support a discrete, finite set of voltages
- relativity & black holes – this changes the speed of time
- randomness – this is an open question and not actually known, but most people believe that it does not overturn the Church-Turing thesis

The first model of computing that truly challenges the Church-Turing thesis is **quantum computing**.

2 Qubits

The equivalent textbook sections are **Nielsen & Chuang, 1.2 + 2.1.1 - 2.1.4**.

The Church-Turing thesis survived many changes in computation, even (mostly) randomized computation. The first model of computing that truly challenged the Church-Turing thesis was **quantum computing**.

In randomized computation, a bit in $\{0, 1\}$ is upgraded to a probability distribution (p_0, p_1), with $p_0, p_1 \geq 0$ and $p_0 + p_1 = 1$, where p_0 is the probability that the bit is 0 and p_1 is the probability that the bit is 1.

For n bits, we upgrade $\{0, 1\}^n$ to a probability distribution $(p_0, \cdots, p_{2^n}) \in \mathbb{R}^{2^n}$, where \$$p_x \geq 0$ and $\sum_x p_x = 1$. This has a probability associated to each of the 2^n potential assignments of bits in $\{0, 1\}^n$.

2.1 Qubits

Now, we want to abstract away from the physical substrate. Let's apply this through quantum mechanics. Classical mechanics is based on two perfectly distinguishable states, say a voltage level of 0 and a voltage level of 5. A quantum 2-level system, which we call a "qubit," also has two perfectly distinguishable states. What makes it quantum is that the state might be some *superposition*, or linear combination, of the two states, with an amplitude of each state. We consider not only two possible states, but also superpositions of those two states.

Example 2.1 (Photon Polarization)
An example might be photon polarization, where it could be either vertically or horizontally polarized. However, a superposition might be if it were diagonally polarized.

Example 2.2 (Electron Spin)
Another example would be an electron with spin 1/2, where it could either be *up* or *down*. A superposition is harder to describe, but

Example 2.3 (Atomic Orbitals)
Atomic orbitals also are quantum, where there are 1s orbitals and 2p orbitals, which are perfectly distinguishable. We also have hybrid orbitals, which have an amplitude of each of the states.

We have "ket 0" and "ket 1", which we write as $|0\rangle$ and $|1\rangle$. In general, we would write $|\text{state}\rangle$.

What is interesting is combining two states using superposition. We have some complex amplitude of one state, added to some complex amplitude of the other state, which we write as

$$\alpha_0|0\rangle + \alpha_1|\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \in \mathbb{C}^2.$$

In this notation,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

If we have n qubits, we end up with the linear combination

$$\alpha_{000}|000\rangle + \alpha_{001}|001\rangle + \cdots + \alpha_{111}|111\rangle \in \mathbb{C}^{2n}.$$

In terms of dimensions, quantum computing uses exponentially large vectors. This is similar to randomized computing, but it turns out that amplitudes are more powerful than probabilities, which we will get into.

2.2 Comparison of Models of Computation

For every model of computation, there are **states** and **operations** on these states. Let's compare states and operations for deterministic computing, randomized computing, and quantum computing.

- **Deterministic Computation.** In deterministic computation, our states are in $\{0, 1\}^n$. We look at functions $f : \{0, 1\}^n \longrightarrow \{0, 1\}^m$.

Example 2.4

For example, in 1-bit computing, where $n = m = 1$, there are only 4 possible functions. There is the IDentity function, the NOT function, and the CONSTant 0 and 1 functions.

With multiple bits, we get to AND, OR, FAN-OUT, and so on, but there are always only finitely many possible functions.

- **Randomized Computation.** In randomized computing, the system is in an unknown state, but the distribution is known. Then we would say that the state is a probability distribution over the possible states. The operations in randomized computing include all the deterministic operations, but also can introduce randomness. Thus, we could say "we flip a bit with probability γ," which would represent error or noise.

Example 2.5

For example, in a binary symmetric channel, which comes from information theory, we would transmit a bit 0 or 1, and with probability $1 - \gamma$, the correct bit would be transmitted, and with probability γ, the bit would be flipped to the other one.

Then, we would say that the function can be described with a matrix, where

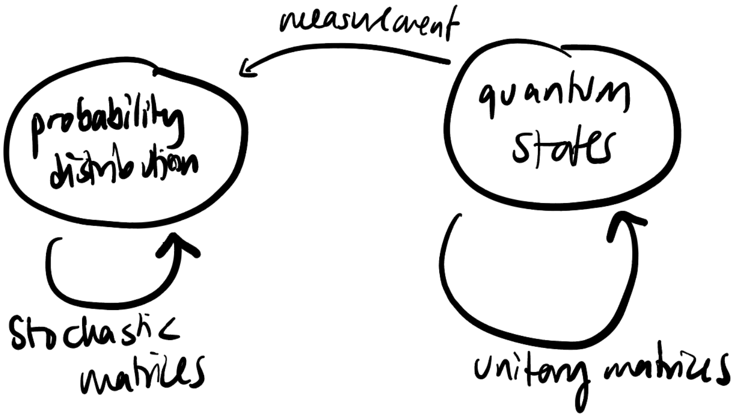
$$\begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = \begin{pmatrix} 1-\gamma & \gamma \\ \gamma & 1-\gamma \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}.$$

For $\gamma = 0$, this would represent the IDentity function, and for $\gamma = 1$, this would represent the NOT function. In a more interesting case, for $\gamma = 1/2$, we end up with $\begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$, which produces a completely random output (regardless if the input bit is 0 or 1, it will randomly be sent to 0 or 1), and ends up with $\begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$.

In general, where $N = 2^n$ for n-bit strings, we want to describe a probabilistic function where we write down for each input distribution, the output distribution. For $p \in \mathbb{R}^N$, an operation can be represented by $p \longrightarrow Sp$ for S an $N \times N$ matrix. Let's suppose that $x \to y$ with probability S_{yx}. For every possible transition from an input x to an output y, we write down the probability as a grid of numbers S. Then, $P(\text{output } = y) = \sum_x p_x S_{yx} = (Sp)_y$ Thus, the probabilistic function is described by matrix multiplication using the matrix that describes each of the transition probabilities.

What are the constraints on S? For all x, $\sum_y S_{yx} = 1$. Since all entries are valid probabilities, $S_{yx} \geq 0$. Matrices obeying such constraints (every column is a valid probability distribution) are called "stochastic" matrices, which take in a probabilistic state and return a different probabilistic state.

- **Quantum Computation.** For quantum states, because there are also probability distributions involved, there are two concepts to introduce. First, we want to work out the operations which take in a quantum state and return a quantum state, which turn out to be unitary matrices. Secondly, we also want to work out how to "measure" quantum states and return a probabilistic state. Relating quantum states back to observation is what makes quantum computation useful.



Definition 2.6

Consider a quantum state

$$|\psi\rangle = \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix} = \psi_0|0\rangle + \psi_1|1\rangle.$$

The **standard measurement** states that

$$\Pr(0) = |\psi_0|^2 \text{ and } \Pr(1) = |\psi_1|^2.$$

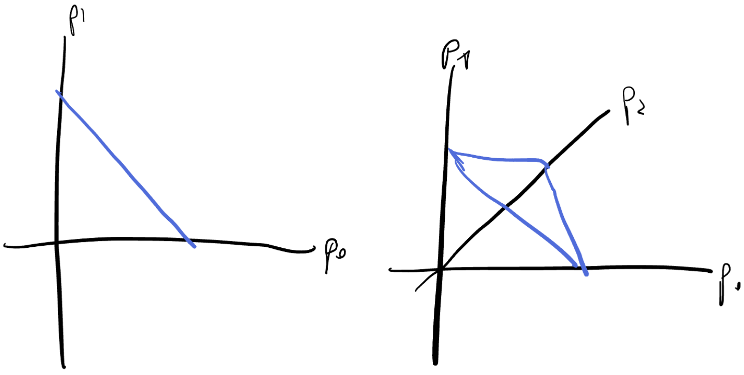
We can take this as an axiom, which is experimentally consistent with physical observations.

In particular, the standard measurement implies that $|\psi_0|^2 + |\psi_1|^2 = 1$, and more generally, for $|\psi\rangle \in \mathbb{C}^d$, where $d = 2^n$ for n qubits,

$$\sum_x |\psi_x|^2 = 1.$$

It turns out that in the quantum computing setting, this is basically the *only* constraint necessary.

Let's compare this with the probability condition: $|p_0|^1 + |p_1|^1 = 1$. In the $p_0 - -p_1$ plane, the allowed states form a line. For a "trit," with three states, there would be a triangle of allowed states, and in higher dimensions the allowed states form the "probability simplex."



For quantum states, we can think of $\mathbb{C}^{2^n}$ as instead $\mathbb{R}^{2 \cdot 2^n}$, where we think of each complex number as two real numbers, which makes our condition from . This ends up with a *sphere* of allowed quantum states. Very importantly, spheres are round and thus rotationally invariant, but triangles/simplices are not. In probability, there are "preferred states" at the corners, which are the deterministic states, which are special. For quantum states, there are *no* preferred states. We could write states as a superposition of many choices of two states, rather than simply 0 and 1, and our theory will work out just as well. That is, we can change basis with no issue, and without losing physical meaning.

3 Operations on Qubits

3.1 Review

Last time, we stated that a quantum state can be described as $|\psi\rangle = \psi_0|0\rangle + \psi_1|1\rangle$, where (ψ_0, ψ_1) is a unit vector. Then, the state is 0 with probability $|\psi_0|^2$ and 1 with probability $|\psi_1|$.

Much of quantum computation is obtained by taking quantum mechanics as axiomatic. The axioms we will take are:

- Measurement is defined as the state being x with probability $|\psi_x|^2$.
- Transformations taking one state to another are linear.
- Quantum states are a unit vector in a vector space.
- The composite of two state spaces is a tensor product of the two spaces.*

3.2 Vector Spaces and Transformations

Our goal is to study functions that transform one quantum state to another. We want to show that if we have $|\psi\rangle \longrightarrow |\psi'\rangle$, then we can write $|\psi'\rangle = U|\psi\rangle$, as a matrix-vector product, where U is unitary. The fact that U must be unitary is not shocking, as we are mapping unit vectors to unit vectors, but let's go through it.

We want to figure out what choices of U preserve normalization. Recall that for vectors $v = (v_0, v_1), w = (w_0, w_1)$, we can define a Hermitian inner product

$$\langle v, w\rangle = v_0^* w_0 + v_1^* w_1,$$

where the $*$ means complex conjugate (for $z = a + bi = re^{i\theta}$, $z^* = a - bi = re^{-i\theta}$). Note that these brackets are not the same as the brackets used in writing down quantum states. In general, we have

$$\langle v, w\rangle = \sum v_x^* w_x,$$

and thus

$$\langle v, v\rangle = \sum |v_x|^2 = ||v||^2.$$

Definition 3.1
In quantum mechanics, we have the notation

$$|v\rangle = \begin{pmatrix} v_0 \\ v_1 \end{pmatrix},$$

which is called a **ket**, and

$$\langle v| = \begin{pmatrix} v_0^* & v_1^* \end{pmatrix} = |v\rangle^\dagger = |v\rangle^{T*},$$

which is called a **bra**, and is the complex conjugate transpose of the ket.

Using this notation, we can write the matrix product

$$\langle v||w\rangle = \begin{pmatrix} v_0^* & v_1^* \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} = v_0^* w_0 + v_1^* w_1 = \langle v, w\rangle = \langle v|w\rangle,$$

which turns out to be the same as the inner product. We will commonly write the inner product as $\langle v|w\rangle$.

Taking the basis vectors $0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, we end up with

$$\langle 0|0\rangle = 1, \langle 0|1\rangle = 0, \langle 1|0\rangle = 0, \langle 1|1\rangle = 1.$$

We can also take inner products of elements of the vector space spanned by 0 and 1; for example,

$$\langle 0|\frac{1}{2}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}.$$

*We may or may not cover this in the future.

We can also take

$$|0\rangle\langle 1| = |0 \bigotimes 1| = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

This combines two states and produces a matrix, and we will see that these matrices are "operators," in quantum mechanics language. In general, this is called the outer product.

Definition 3.2
The **outer product** of v and w is

$$|v\rangle\langle w|,$$

which is a matrix.

3.3 Unitaries to Computation

Now, let's discuss operators. Recall that we took linearity of transformations as an axiom.

Guiding Question
Which linear matrices or operations preserve norm?

Let's think about a linear operator $U : \psi \longrightarrow \psi'$, where $|\psi| = |\psi'|$.

We have

$$\sum_x |\psi_x|^2 = \langle \psi|\psi\rangle,$$

where $|\psi\rangle = \psi_0|0\rangle + \psi_1|1\rangle$. Let $|\psi'\rangle = U|\psi\rangle$. Then we have

$$\begin{aligned} \langle \psi'|\psi'\rangle &= |\psi'\rangle^\dagger|\psi'\rangle \\ &= (U|\psi\rangle)^\dagger U|\psi\rangle \\ &= \langle \psi|U^\dagger U|\psi\rangle, \end{aligned}$$

and we want this to be equal to $\langle \psi|\psi\rangle$.

We must therefore have

$$\langle \psi|U^\dagger U|\psi\rangle - \langle \psi|\psi\rangle = 0$$
$$\langle \psi|(U^\dagger U - I)|\psi\rangle * = 0,$$

which implies that $U^\dagger U = I$.

Definition 3.3
A **unitary matrix** satisfies $U^\dagger U = I$, or $U^{-1} = U^\dagger$. Equivalently, a unitary matrix preserves norm, as we just showed.

Let's see some examples of unitary matrices.

Example 3.4 (Rotation)
Rotations form a large class of unitary matrices. In fact, large class of gates we perform on qubits are rotations in a very large space.

Example 3.5 (Phase Shift)
Phase shift matrices are of the form

$$\begin{pmatrix} e^{i\phi_0} & 0 \\ 0 & e^{-i\phi_0} \end{pmatrix},$$

which turns out to be a rotation around the z-axis in a vector space we will see later.

Guiding Question
How are unitary matrices different from randomized computation?

Recall the binary symmetric channel, which takes $1 \longrightarrow 1$ and $0 \longrightarrow 0$ with probability $1-p$ and $0 \longrightarrow 1$ and $1 \longrightarrow 0$ with probability p. For a binary symmetric channel with $p = 1/2$, applying the binary symmetric channel once or twice will give a completely random output.

Example 3.6 (Hadamard Gate)
Consider the matrix $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. We compute $H^\dagger = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H$ and $H^2 = \frac{1}{2}\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = I$, so H is unitary. In fact, H is Hermitian.

We compute

$$H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle,$$

so there is a probability of $1/2$ to be 0 and a probability of $1/2$ to be 1, with an input of 0. Also,

$$H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle,$$

so we also have a probability of $1/2$ to be 0 and a probability of $1/2$ to be 1, with an input of 0. In fact, we can look at H as an input-output table, and read off the columns. This looks the same as the binary symmetric gate with one application of it.

However, looking at H^2, we see that applying H twice to $|0\rangle$ gives back $|0\rangle$, and applying H twice to $|1\rangle$ gives back $|1\rangle$.

This is called the **Hadamard gate**. This looks random after one application, but is deterministic after two applications. The Hadamard gate is one of the first gates that looks different from classical computation models.

3.4 Interference and Information

This leads us to our last discussion of the day.

Guiding Question
What types of information can be transmitted with complex gates

Let's start with a 0 state, which is a qubit, and send it through the Hadamard gate, and then send it through a phase change $\begin{pmatrix} e^{i\phi_0} & 0 \\ 0 & e^{-i\phi_0} \end{pmatrix}$, then send it through the Hadamard gate again.

$$|0\rangle \xrightarrow{H} |\alpha_1\rangle \xrightarrow{\text{phase shift}\begin{pmatrix} e^{i\phi_0} & 0 \\ 0 & e^{-i\phi_0} \end{pmatrix}} |\alpha_2\rangle \xrightarrow{H} |\alpha_3\rangle.$$

We have

$$|\alpha_1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle,$$

$$|\alpha_2\rangle = \frac{e^{i\phi_0}}{\sqrt{2}}|0\rangle + \frac{e^{-i\phi_0}}{\sqrt{2}}|1\rangle,$$

and

$$\begin{aligned}
|\alpha_3\rangle &= H|\alpha_2\rangle \\
&= \frac{e^{i\phi_0}}{\sqrt{2}}(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) + \frac{e^{-i\phi_0}}{\sqrt{2}}(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle) \\
&= \frac{e^{i\phi_0} + e^{-i\phi_0}}{2}|0\rangle + \frac{e^{i\phi_0} - e^{-i\phi_0}}{2}|0\rangle \\
&= \cos\phi_0|0\rangle + i\sin\phi_0|1\rangle.
\end{aligned}$$

Thus, for α_3,

$$P(0) = \cos^2(\phi_0), P(1) = \sin^2(\phi_0).$$

If we do the same with a binary symmetric channel, this behavior of interference between the two H's is not exhibited.

Interference is a hallmark of quantum behavior. We will learn that this is the interference of two computational pathways, which can be complicated evaluations of functions, will give the quantum Fourier transform. This is the real beauty and secret behind quantum computation.

4 Generalized Measurement

4.1 Review

Let's do a conceptual overview of what we talked about at the end of last time: interference. Recall last time that one difference between doing a series of stochastic matrices (random operations) and doing a series of unitary matrices, is interference. A series of stochastic matrices, is transitioning from one state to another with different probabilities, and summing up all the paths to get from different states will give a sum of a product of probabilities. In this case, we are adding up nonnegative numbers, so adding more possibilities will only increase the probability.

In the quantum setting, it's similar in that a unitary matrix sends one state to another state that is a superposition of basis states, with complex amplitudes, and so on. However, adding complex numbers rather than nonnegative numbers will conceptually be very different. This can be called constructive or destructive interference, based on whether phases cancel out or not, and if the numbers are out of phase, this is called incoherent. In physics, adding complex numbers is called interference, and it is very relevant to wave mechanics. Thus, there will be interference through different paths of computation.

4.2 Generalized Measurement

So far, we defined the standard measurement as an axiom, but we can use some linear algebra to talk about a more general concept of measurement. A measurement comes from taking some quantum state, and doing an experiment given that state; the state "collapses" to an outcome if the experiment results in that outcome. A measurement can be described by specifying the probability of each possible outcome, although a single experiment will only collapse the state to a specific outcome.

Definition 4.1
For the **standard measurement**, where $|\psi\rangle = \begin{pmatrix}\psi_0\\ \psi_1\end{pmatrix} = \psi_0 |0\rangle + \psi_1 |1\rangle$, the state will collapse to $|0\rangle$ when measured with probability $|\psi_0|^2$ and will collapse to $|1\rangle$ with probability $|\psi_1|^2$.

Today, we will not discuss the full range of measurements, but we will move in that direction. In fact, $|0\rangle$ and $|1\rangle$ are not necessarily special, and there are other forms of measurement as well.

4.2.1 Plus and Minus States

Let's define two new states.

Definition 4.2
The quantum **plus state** is

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

and the **minus state** is

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

Example 4.3 (Measuring $|+\rangle$ and $|-\rangle$)
We might think that we won't be able to measure $|+\rangle$ and $|-\rangle$ states, since if we keep using the standard measurement, we will keep getting $|0\rangle$ and $|1\rangle$ with probability $1/2$ for both the plus and minus states. However, if we first apply a gate, the Hadamard gate from last time, we will get

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

and

$$H|-\rangle = |1\rangle.$$

This makes us think that we should do a measurement that is "apply the Hadamard gate and use the standard measurement." Generally, if we start with

$$|\psi\rangle = \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix},$$

we will get

$$H|\psi\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} \psi_0 + \psi_1 \\ \psi_0 - \psi_1 \end{pmatrix}.$$

Measuring $H|\psi\rangle$, we get

$$P(0) = \frac{1}{2}|\psi_0 + \psi_1|^2.$$

The fact that there are different kinds of measurement that are incompatible is an example of "complementarity" or "uncertainty." There is no measurement that dominates all other measurements completely.

4.2.2 Higher Dimensions: Qudits

Qubits are 2-level systems in $\mathbb{C}^2$ and qudits are d-level systems in $\mathbb{C}^d$.

Definition 4.4
A **qudit** is a d-dimensional vector

$$|\psi\rangle = \begin{pmatrix} \psi_0 \\ \vdots \\ \psi_{d-1} \end{pmatrix} = \psi|0\rangle + \cdots + \psi_{d-1}|d-1\rangle.$$

We use $|0\rangle, \cdots, |d-1\rangle$ as the basis vectors, and some may use $|1\rangle, \cdots, |d\rangle$. There is no standard indexing. If we have n qubits, we will be in $d = 2^n$, since there are 2^n on-off configurations for each qubit. However, we can actually use any d, not just powers of two.

Example 4.5 (qu3its)
For example, in $\mathbb{C}^3 = \left\{ \begin{pmatrix} \psi_0 \\ \psi_1 \\ \psi_2 \end{pmatrix} : \psi_0, \psi_1, \psi_2 \in \mathbb{C} \right\}$, we can write any vector as $\psi_0|0\rangle + \psi_1|1\rangle + \psi_2|2\rangle$.

4.2.3 Bases

In three dimensions, and also in general, we can define a basis.*

Definition 4.6 (Basis)
We say that $\{|v_0\rangle, |v_1\rangle, |v_2\rangle\}$ is a basis if any vector in $\mathbb{C}^3$ can be written uniquely as a $\mathbb{C}$-linear combination of $|v_0\rangle, |v_1\rangle$, and $|v_2\rangle$.

Thus, for an n-dimensional basis, there are n degrees of freedom.

Example 4.7
Different bases for $\mathbb{C}^2$ include:

- $|0\rangle, |1\rangle$
- $|+\rangle, |-\rangle$
- $|\pm i\rangle$
- $17|0\rangle, |1\rangle$
- $|0\rangle, |+\rangle$

*Recall from linear algebra the definitions of a complex vector space, the span of vectors, linear combinations, and so on.

In order to deal with some complications from coefficients, we should talk about orthogonal bases and orthogonal vectors.

Definition 4.8

Two vectors v and w are orthogonal if their inner product is zero. That is, if $\langle v|w\rangle = 0$.[a]

[a] This can be seen using the cosine formula, where we see that $\langle v|w\rangle = ||v||||w|| \cos\theta$, so if the inner product is zero, then the angle θ between them is $\pi/2$.

Definition 4.9

We say that a set of vectors, or in particular a basis, is **orthonormal** if

$$\langle v_i|v_j\rangle = \delta_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} .$$

Proposition 4.10

Let $|v\rangle_1, \cdots, |v\rangle_d$ be an orthonormal basis, and consider $|\psi\rangle$. Then

$$\langle\psi|\psi\rangle = \sum_{i=1}^{d} |a_i|^2,$$

where the a_i are the amplitudes of ψ in terms of the basis.

Proof. We can calculate in bra-ket notation that

$$\langle\psi| = \sum_{i=1} da_i^* \langle v_i| .$$

Then,

$$\langle\psi|\psi\rangle = \sum_{i=1}^{d} a_i^* \langle v_i| \sum_{j=1}^{d} a_j|v_j\rangle = \sum_{i,j=1}^{d} a_i^* a_j \langle v_i|v_j\rangle = \sum_{i=1}^{d} |a_i|^2.$$

Here the notation * refers to the complex conjugate. □

4.2.4 Generalized Measurement

Now, we can define a generalized measurement rule.

Proposition 4.11

We can measure in any orthonormal basis $|v_0\rangle, \cdots, |v_{d-1}\rangle$. If $|\psi\rangle = \sum_{i=0}^{d-1} a_i |v\rangle_i$, then

$$P(i) = |a_i|^2,$$

and its post measurement state is $|v_i\rangle$.

5 Generalized Measurement

Today, we will cover:

- General measurements using unitaries
- Uncertainty
- Multipartite systems, tensor products, and entanglement

5.1 Measurement

Guiding Question
Given the standard measurement, how can we define measurement on an arbitrary orthonormal basis?

Consider an arbitrary orthonormal basis $|v_0\rangle, \cdots, |v_{d-1}\rangle \in \mathbb{C}^d$. Given a state $|\psi\rangle$, which is a linear combination of the basis vectors, we want to measure the state and figure out the amplitudes of each of the basis vectors. By scientific hypothesis, we can use the inner product to calculate the probability of being in state i:

$$P[i] = |\langle v_i|\psi\rangle|^2,$$

and in this case, we say that the post-measurement state is $|v_i\rangle$. A measurement takes in a quantum state and spits out a classical answer in terms of probabilities. Measuring may or may not destroy information, depending on whether the probability is equal to 1 or less than 1.

Given $\psi_i = \langle i|\psi\rangle$, the ith component of ψ, we can decompose $|\psi\rangle$ as

$$|\psi\rangle = \psi_0|0\rangle + \cdots + \psi_{d-1}|d-1\rangle.$$

Last time, recall that we took measurements in the $|+\rangle$ and $|-\rangle$ basis by applying the Hadamard gate, then using the standard measurement. We can define an analogous measurement for general orthonormal bases.

Definition 5.1
For an orthonormal basis $|v_0\rangle, \cdots, |v_{d-1}\rangle$, let the **measurement operator** $U = \sum_{i=0}^{d-1} |i\rangle \langle v|_i$ be the matrix with $|v_i\rangle$ as the ith row.

Proposition 5.2
The measurement operator U is unitary if the basis is orthonormal.

Proof. We can check that U is unitary:

$$UU^\dagger = \sum_i |i\rangle \langle v_i| \sum_j |v_j\rangle \langle j| = \sum_{i,j} |i\rangle \langle j| \langle v_i| |v_j\rangle = \sum_i |i\rangle \langle i| = I.^{*}.$$

□

We can verify that this corresponds to the same matrix that we used last time for $|+\rangle$ and $|-\rangle$.

[*] Here, we take in three dimensions $|0\rangle = \begin{pmatrix}1\\0\\0\end{pmatrix}$, $|1\rangle = \begin{pmatrix}0\\1\\0\end{pmatrix}$, and $|2\rangle = \begin{pmatrix}0\\0\\2\end{pmatrix}$

Example 5.3
Let's say we want to take measurements in the $|+\rangle, |-\rangle$ basis. Last time, we said that we should do the Hadamard gate, then measure in the standard basis. Let's see what our new formulation gives us. Let $v_0 = |+\rangle$ and $v_1 = |-\rangle$. In the standard basis,

$$|v_0\rangle = |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

and

$$|v_1\rangle = |+\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Thus, we end up with

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

which is precisely the Hadamard gate, and so taking U as we defined above matches with the Hadamard gate from last time.

Thus, our formulation is that to take the amplitude of $|\psi\rangle$ in a basis v_i, we define the measurement operator $U = \sum |i\rangle \langle v_i|$, apply U to $|\psi\rangle$, and then take the standard measurement. We can take probabilities of being in state $|i\rangle$ by taking the norm squared of the amplitude of vector i:

$$\begin{aligned} P[i] &= |\langle i|U|\psi\rangle|^2 \\ &= \left| \langle i| \sum_j | \, |j\rangle \langle U|_j \, |\psi\rangle \right|^2 \\ &= |\langle v_i|\psi\rangle|^2. \end{aligned}$$

Essentially, we take measurements by rotating into our new basis, then taking the measurement. Unfortunately, our new measurement is with respect to the standard basis, not our arbitrary orthonormal basis. To convert back, we can simply rotate back by multiplying by U^{-1}, or equivalently as U is unitary, $U^\dagger$.

5.2 Uncertainty

Consider two different bases, $\{|0\rangle, |1\rangle\}$ and $\{|+\rangle, |-\rangle\}$. There cannot be a definite outcome in both. Assume we have two bases $|v_0\rangle, \cdots, |v_{d-1}\rangle$, and $|w_0\rangle, \cdots, |w_{d-1}\rangle$. What is the probability

$$P[\text{get } |w_j\rangle \text{ given state } |v_i\rangle] = |\langle w_j|v_i\rangle|^2?$$

The only case when no information is lost is when the bases are permutations of each other.

Question. *What do you mean by losing information?*

Answer. *By losing information, suppose that someone sends a message in the v basis, and they send the ith basis vector. The receiver measures in the w basis. That corresponds to their outcome. They won't deterministically get the outcome. If you didn't measure and instead sent the vector, then there is no information lost since you can multiply by th einverse matrix, but once you measure, then you can't get the information back.*

Example 5.4
Here are some examples of multiple bases.

Basis	Polarization	Spin−1/2
$\|0\rangle, \|1\rangle$	vertical and horizontal	up and down
$\|+\rangle, \|-\rangle$	diagonal and the opposite diagonal	right and left
$\|\pm i\rangle = \frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$	clockwise and counterclockwise	into the board and out of the board

Uncertainty means that if I know the y-direction of the spin, that obscures my knowledge of the z-direction of the spin. This is related to position and momentum, but we will talk about this another day.

5.3 Multiple Systems

- **Deterministic.** With one die, we have 6 possibilities: $[6] = \{1, 2, 3, 4, 5, 6\}$. With two dice, we have 36 possibilities, described by the set $[6] \times [6] = \{(x, y) : x, y \in [6]\}$. In general, for n bits, there are 2^n states. The benefit of binary over unary is that the size of the possible number of states grows exponentially with the size of the system.

- **Probabilistic Computing.** In one system, we have $p_A = \begin{pmatrix} p_A(1) \\ \vdots \\ p_A(m) \end{pmatrix} \in \mathbb{R}^m$. If we add a second system $p_B = \begin{pmatrix} p_B(1) \\ \vdots \\ p_B(n) \end{pmatrix} \in \mathbb{R}^n$, the joint system is a vector $p_{AB} \in \mathbb{R}^{mn}$, with a probability for each pair of states. Like in deterministic computing, we have exponential growth, but in the *dimension* of the vector space containing the vectors, rather than the *number* of possible states. However, the joint system may have all sorts of correlations.

  In the pleasant case where the probability distributions are independent, then

  $$Pr[a, b] = P_A(a)P_B(b).$$

  The product distribution is written A vector would be written as

  $$p_{AB} = \begin{pmatrix} p_A(1)p_B(1) \\ p_A(1)p_B(2) \\ \vdots \\ p_A(a)p_B(b) \\ \vdots p_A(m)p_B(n) \end{pmatrix},$$

  which can be denoted as

  $$p_{AB} = p_A \otimes p_B,$$

  the tensor product of p_A and p_B.

  Definition 5.5
  The **tensor product** of two vectors, $x \otimes y$, is the vector of all products of entries of x and entries of y.[a]

  [a] What order should these products be listed in? It is natural for it to inherit a lexicographical order from the underlying systems. What is important is to be consistent.

- **Quantum Computing.**

  For example, if one qubit in $\mathbb{C}^2$, and two qubits are in $\mathbb{C}^4$, with basis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, then in general n qubits will be in $\mathbb{C}^{2n} = |00\cdots0\rangle, \cdots, |11\cdots1\rangle$.

  Consider two qubits, one in state $|\alpha\rangle = \alpha_0 |0\rangle + \alpha |1\rangle$ and $|\beta\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle$. An "axiom" of quantum mechanics, which is actually somewhat forced by consistency, is that the joint state of the two qubits is

  $$|\alpha\rangle \otimes |\beta\rangle = \begin{pmatrix} \alpha_0 |\beta\rangle \\ \alpha_1 |\beta\rangle \end{pmatrix} = \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix} = \sum_{i,j\in\{0,1\}} \alpha_i\beta_i |ij\rangle.^{\dagger}$$

  However, for this joint state, α and β are not correlated in any interesting ways. We could simply have considered the two separately. This is the quantum equivalent of independent probability distributions. These states are "not entangled."

[†] Maybe we would want to write this as a 2-dimensional matrix, or in general an n-dimensional tensor, but we choose to write it as a long vec

In fact, just like how some probability distributions are not independent, some qubits will be correlated, or entangled. For example,

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \neq |\alpha\rangle \otimes \beta.$$

States that can be written as products are called **product states**, and states that cannot are called **entangled states**. They are analogous to non-independent random variables, but are quantum states with amplitudes, so behave differently from probability distributions in certain ways.

In general, most states are entangled, and product states do not account for all states in the system.

6 Tensor Products and Entanglement

Today, we will cover:

- Tensor product
- Entanglement
- Two-qubit gates
- No-cloning theorem
- Partial measurement

6.1 Review

Recall from last time that given two quantum states $|\alpha\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$ and $|\beta\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle$, the product state is

$$|\alpha\rangle \otimes |\beta\rangle = \alpha_0\beta_0 |00\rangle + \alpha_0\beta 1 |01\rangle + \alpha_1\beta_0 |10\rangle + \alpha_1\beta_1 |11\rangle .$$

This is the most general product state, but not all two-qubit states can be written in this form. For example, $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ is not a product state, and we say it is **entangled**.

A bit of notation: we will write

$$|00\rangle = |0\rangle \otimes |0\rangle , |01\rangle = |0\rangle \otimes |1\rangle ,$$

and so on.

Definition 6.1
We say a two-qubit state is **entangled** if it cannot be written as a product state.

Example 6.2 (Product States)
Some examples of product states include:

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

Example 6.3 (Entangled State)
The state

$$\frac{|00\rangle + |01\rangle + |10\rangle - |11\rangle}{\sqrt{2}}$$

is entangled, and can be written as

$$\begin{pmatrix} 1 & 1 & 1 & -1 \end{pmatrix} \textit{ or } \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} .$$

It *cannot* be written as a product state of any two qubits.

The full theory of how to think about entangled states and so on is covered in 8.371.

Question. *Should we write out the amplitudes/coefficients of the product state in a matrix or a vector?*

Answer. *It depends on the context. In the case of determining whether a 2-qubit state is entangled or a product state, it can be useful to write it out as a matrix. We write a 2-qubit state as* $\begin{pmatrix} \psi_{00} & \psi_{01} \\ \psi_{10} & \psi_{11} \end{pmatrix}$. *If the 2-qubit state is a product state, the matrix of coefficients will have rank 1, and if it is entangled, it will have rank greater than 1. This is true in general for n-qubit states.*

6.2 Tensor Products of Matrices

Entanglement is often seen as a mysterious phenomenon, where experimentalists or theorists may come up with Bell inequality violations, and other complicated ideas. It's common to hear about how difficult it is to do an experiment with entanglement, and it requires a lot of work. However, this may seem in tension with the fact that *most states are entangled*, since most matrices are full rank. For example, n qubits has 2^n degrees of freedom, as there it is a superposition of 2^n basis vectors[*], but $|\alpha_1\rangle \otimes \cdots \otimes |\alpha_n\rangle$ has only $2n$ degrees of freedom, since there are 2 degrees of freedom to choose coefficients for each $|\alpha_i\rangle$.

Definition 6.4
The **tensor product** of $U \otimes V$ is a block matrix

$$\begin{pmatrix} U_{00}V & U_{01}V \\ U_{10}V & U_{11}V \end{pmatrix}.$$

If U and V are 2×2, then their tensor product will be 4×4. If $U = \sum_{ij} U_i j \, |i\rangle \langle j|$, then

$$U \otimes V = \sum_{ijk\ell} U_{ij} V_{k\ell} \, |i,j\rangle \langle j,\ell| \, ,$$

where $|i,k\rangle = |i\rangle \otimes |k\rangle$.

The basic rule is arranging all the possible ways of multiplying one entry of U with one entry of V, and arranging them in one giant matrix.

Example 6.5 (Tensor product of identity)

Let $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ be the reverse matrix. The tensor product of $I \otimes X$ is $\begin{pmatrix} X & 0 \\ 0 & X \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$, which is X's in the shape of an I. Instead of indexing the rows and columns by 1, 2, 3, and 4, we can index them by $00, 01, 10$, and 11, which are the basis vectors for 2-qubit states.

On the other hand, $X \otimes I$ is $\begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$, which is I's in the shape of an X. The indexes corresponding to each column and row are $11, 01, 10$, and 11. We can read off the matrix to see that $(I \otimes X)\,|11\rangle = |10\rangle$, since all the coefficients are zero except for $|10\rangle$, which has a coefficient of 1.

We have

$$(U \otimes V)(|\alpha\rangle \otimes |\beta\rangle) = U\,|\alpha\rangle \otimes V\,|\beta\rangle \, .$$

We definitely want this to be true, since α and β may not be next to each other. This means that α is undergoing U and β is undergoing V, and physically α and β should not be influencing each other. We can almost think of the tensor product as a "comma," where we say the first part of the tensor product is acting on the first system, and the second part of the tensor product is acting on the second system. This breaks down a little when we think about quantum field theory, but we don't have to think about this.

Proposition 6.6
A more general rule for tensor products is that

$$(A \otimes B)(C \otimes D) = AC \otimes BD.$$

Also,

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C.$$

[*] For example, with 3 qubits we would have $|000\rangle, |001\rangle, \cdots, |111\rangle$.

Again, think of the tensor product as a "comma": A and C are acting on the first system, and B and D are acting on the second system.

Not all operations can be written as tensor products, and in this case it will cause entanglement, potentially producing entangled states from non-entangled states.

Example 6.7 (Non-product unitary matrix)
One example of a non-product unitary matrix, which cannot be written as a tensor product, is the controlled-not gate, CNOT $= |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. We can write this as

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

Thus, 00 maps to itself, 01 maps to itself, and 10 and 01 swap.

We can calculate that

$$CNOT(|+\rangle \otimes |0\rangle) = CNOT\left(\frac{|00\rangle + |10\rangle}{\sqrt{2}}\right) = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

Another example of a 2-qubit gate is $SWAP(|\alpha\rangle \otimes |\beta\rangle) = \beta \otimes \alpha$.

6.3 No-Cloning Theorem

Mathematically, the tensor product is important because it is a bilinear operation: it is linear in either of its inputs, working by linear combinations.

- $(|\alpha\rangle + |\beta\rangle)(|\gamma\rangle + |\delta\rangle) = |\alpha\rangle \otimes |\gamma\rangle + |\alpha\rangle \otimes |\delta\rangle + |\beta\rangle \otimes |\gamma\rangle + |\beta\rangle \otimes |\delta\rangle$

In some sense, the tensor product is the *most general* bilinear map. Any other bilinear map can be expressed by doing the tensor product, then applying a linear map.

Proposition 6.8
If $f(|\alpha\rangle, |\beta\rangle)$ is bilinear, then $f = T|\alpha\rangle \otimes |\beta\rangle$, where T is some linear operation.

This characteristic of the tensor product is because the tensor product contains all the information of ways to combine $|\alpha\rangle$ and $|\beta\rangle$. We will not directly use this fact a lot, but it's nice to know the intuition for the tensor product. The inner product is a bilinear map (or sesquilinear) map taking in two vectors and outputting a complex number. Tnesor product is similar: a bilinear map taking in two matrices and outputting another, larger matrix.

A very important feature of classical computing is that given information, it's easy to make a copy of it. The only restrictions are legal, from copyright, but in quantum computing, there are physical difficulties in copying information.

Guiding Question
Can we build a quantum copying machine?

A quantum copying machine should be a unitary matrix, $U_{copy} = U_C$ such that for any state $|\psi\rangle$, $U_C|\psi\rangle \otimes |0\rangle = \psi \otimes \psi$. This is always possible for a *particular* $|\psi\rangle$, but in order to be interesting, we need U_C to be universal, and be able to copy *any* $|\psi\rangle$.

Theorem 6.9 (No-Cloning Theorem)
There does not exist any quantum copying machine.

Proof. We want U_C to satisfy, for all $|\psi\rangle$,

$$U_C |\psi\rangle \otimes |0\rangle = \psi \otimes \psi.$$

Mathematically, this cannot hold because the left hand side is linear, while the right hand side is quadratic. That is, taking 2 as the factor, because U_c is linear and the tensor product is bilinear,

$$U_c(2|\psi\rangle \otimes |0\rangle) = 2(U_c |\psi\rangle \otimes |0\rangle) = 2(\psi \otimes \psi) = 2\psi \otimes \psi,$$

which is *not* the same as $(2\psi) \otimes (2\psi)$. Thus, this equation cannot possibly hold for all ψ. This equation can hold for a particular ψ, but not for all ψ. □

As an application of this, there is a very important useful application of quantum computing. One of the limitations for classical cryptography is if a message is intercepted, the intended recipient *as well as* any eavesdropper can make a copy of that message. The no-cloning theorem states that this can be different in the quantum case. It's not necessary to assume that the message is being copied by an eavesdropper, although it's still important to think about the eavesdropper.

6.4 Partial Measurement

Bell's inequality comes from Bayesian probability, and we will see next time how it relates to quantum computing.

7 Measuring Entangled States

We have stochastic matrices mapping probability distributions to probability distributions. We also have unitary matrices mapping quantum states to quantum states. Measurement consumes a quantum state and produces a probability distribution, while leaving a quantum state left over, after it has collapsed.

Example 7.1 (Conditional Probability)

Consider $p_{AB} = \begin{pmatrix} 1/2 \\ 1/6 \\ 1/6 \\ 1/6 \end{pmatrix}$, where the first entry is the probability of $|00\rangle$, and correspondingly the other entries are for $|01\rangle, |10\rangle$, and $|11\rangle$. If we only look at the first bit, we have $p_A(0) = 2/3$ and $p_A(1) = 1/3$. Moreover, using conditional probability, we have $p_{B|A=0} = \begin{pmatrix} 1/2 \\ 1/6 \end{pmatrix} \cdot \frac{1}{P_A(0)} = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix}$, which is the conditional probability distribution.

Now, consider the quantum analogue.

Example 7.2 (Post-Measurement States)

If we had $|\psi_{AB}\rangle = \begin{pmatrix} 1/\sqrt{2} \\ i/\sqrt{6} \\ -1/\sqrt{6} \\ \frac{1}{\sqrt{6}} \end{pmatrix} = \frac{1}{\sqrt{2}}|00\rangle + \frac{i}{\sqrt{6}}|01\rangle - \frac{1}{\sqrt{6}}|10\rangle + \frac{1}{\sqrt{6}}|11\rangle$. After measuring A, the probability of the first qubit being [0] is $2/3 = (1/\sqrt{2})^2 + (i/\sqrt{6})^2 = 2/3$. Then, the post-measurement state, given the outcome 0, is

$$|0\rangle \otimes \frac{1}{\sqrt{Pr(0)}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{6}}|1\rangle\right) = |0\rangle \otimes \left(\sqrt{\frac{3}{4}}|0\rangle + i\sqrt{\frac{1}{4}}|1\rangle\right) = \begin{pmatrix} 1/\sqrt{2} \\ i/\sqrt{6} \end{pmatrix} / \sqrt{2/3}.$$

We can define this in general.

Definition 7.3 (Partial Measurement)
Consider $|\psi\rangle \in \mathbb{C}^{d_A} \otimes \mathbb{C}d_B := \text{Span}\{|v\rangle \otimes |w\rangle : |v\rangle \in \mathbb{C}^{d_A}, |w\rangle \in \mathbb{C}^{d_B}\}$, where we split $\mathbb{C}^{d_A d_B}$ into two systems A and B. Then, we can measure the A system in an orthonormal basis $|v_1\rangle, \cdots, |v_{d_A}\rangle$. We define subnormalized[a] states $|\varphi_i\rangle_B = (\langle v_i| \,| \otimes I_{d_B})\,|\psi\rangle$.[b] Then, we have $p_A(i) = \langle\varphi_i|\varphi_i\rangle$, and the post-measurement state is

$$\frac{|\varphi\rangle \otimes |v_i\rangle}{\sqrt{p_A(i)}}.$$[c]

Next, measuring the B system in $|w_1\rangle, \cdots, |w_{d_B}\rangle$ has $Pr[j|i] = |\langle w_j|\varphi_i\rangle|^2/p_A(i)$.

[a] The norm is ≤ 1.

[b] Here, $|\psi\rangle$ is of shape $d_A d_B \times 1$. Also, I_{d_B} has dimension $d_B \times d_B$, and $\langle v_i|$ has shape $1 \times d_A$. Their tensor product then has dimensions $1 \cdot d_B \times d_A \cdot d_B$.

[c] We can check that this is normalized because $\sum p_A(i) = \langle\psi| \sum_i |v_i\rangle\langle v_i| \otimes I|\psi\rangle$.

We have

$$Pr(i \text{ for A measurement}, j \text{ for B measurement}) = p_A(i)Pr[j|i] = |\langle w_j|\varphi_i\rangle|^2 = |(\langle v_i| \otimes \langle w_j|)\,|\psi\rangle|^2.$$

We can check that this corresponds to our original notion if we measure the whole combination of systems.

Example 7.4
Consider $|\alpha\rangle \otimes |\beta\rangle$. If we measure, we get

$$P[i,j] = |(\langle v_i| \otimes \langle w_j|)(|\alpha\rangle \otimes |\beta\rangle)|^2 = |\langle v_i|\alpha\rangle|^2 |\langle w_j|\beta\rangle|^2.$$

We can also gain some new insights for entangled states.

Example 7.5

Let $|\psi\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$. If Alice measures in

$$\begin{aligned} |v_0\rangle &= \cos\alpha\,|0\rangle + e^{i\phi}\sin\alpha\,|1\rangle \\ |v_1\rangle &= -\sin\alpha\,|0\rangle + e^{i\phi}\cos\alpha\,|1\rangle\,, \end{aligned}$$

then we have

$$\begin{aligned} \langle\varphi_0| &= (\langle v_0| \otimes I_2)\,|\psi\rangle \\ &= ((\cos\alpha\,\langle 0| + e^{-i\phi}\sin\alpha\,\langle 1|) \otimes I)\frac{|00\rangle+|11\rangle}{\sqrt{2}} \\ &= \frac{\cos\alpha}{\sqrt{2}}\,|0\rangle + \frac{e^{-i\phi}\sin\alpha}{\sqrt{2}}\,|1\rangle\,. \end{aligned}$$

Then, $p_A(0) = \langle\varphi_0|\varphi_0\rangle = 1/2$. Then, the post-measurement state is $|v_0\rangle \otimes (\cos\alpha\,|0\rangle + e^{-i\phi}\sin\alpha\,|1\rangle)$. If we set $\phi = 0$, and Bob measures in $\cos\beta\,|0\rangle + \sin\beta\,|1\rangle$ and $-\sin\beta\,|0\rangle + \cos\beta\,|1\rangle$, we can calculate the probability to be $\pi/2 + \beta$.

Also, $Pr(\text{Bob gets } 0|\text{Alice gets } 0) = \cos^2(\beta - \alpha)$.

8 Entangled States, Nonlocality, and EPR pairs

For measuring entangled states, a lot is analogous to measuring non-independent probability distributions. However, "non-locality" leads to very different behavior from probability distributions. This is called Bell's Theorem.

8.1 Entangled States and Non-Locality

Let's take a look at an interesting example that highlights the distinctive features of quantum computing. "Local" refers to acting on only one qubit or system at a time.

Example 8.1 (EPR Pair)
Consider the state, called an "EPR pair" or "Bell state[a],"

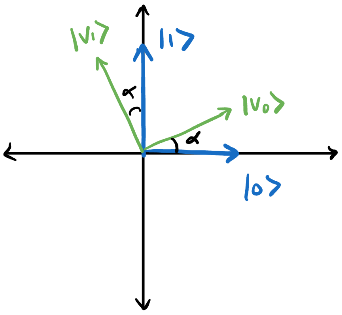
$$|\Phi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

[a] This is not the only pair of states called an EPR pair or Bell state, as there are other maximally entangled states that can be obtained.

Let's measure in the basis

$$|v_0\rangle = \cos\alpha\,|0\rangle + \sin\alpha\,|1\rangle$$
$$|v_1\rangle = -\sin\alpha\,|0\rangle + \cos\alpha\,|1\rangle\,.$$

Geometrically, $|v_0\rangle$ makes an angle of α with the $|0\rangle$ state, and $|v_1\rangle$ makes an angle of α with the $|1\rangle$ state.



For Alice measuring only the first bit, $P(0) = P(1) = 1/2$*. We say this outcome is $x \in \{0, 1\}$.

We can let Bob measure in

$$|w_0\rangle = \cos\beta\,|0\rangle + \sin\beta\,|1\rangle$$
$$|v_1\rangle = -\sin\beta\,|0\rangle + \cos\beta\,|1\rangle\,.$$

For Bob measuring only the second bit, $P(0) = P(1) = 1/2$, for $y \in \{0, 1\}$. However, these are only the marginal distributions, not the joint distribution.

We can write

$$P(x = y) = P(00) + P(11).$$

We calculate

$$\begin{aligned} P(00) &= |(\langle v_0| \otimes \langle w_0|)\,|\Phi\rangle\,|^2 \\ &= (\cos\alpha\,|0\rangle + \sin\alpha\,|1\rangle) \otimes |(\cos\beta\,|0\rangle + \sin\beta\,|1\rangle)\frac{|00\rangle + |11\rangle}{\sqrt{2}}|^2 \\ &= \frac{(\cos\alpha\cos\beta + \sin\alpha\sin\beta)^2}{2} \\ &= \frac{\cos^2(\alpha - \beta)}{2} \end{aligned}$$

*We calculated this last time!

To calculate $P(11)$, we can plug in $\pi/2+\alpha$ and $\pi/2+\beta$ instead, which yields the same answer in the end, so $P(11)=\frac{\cos^2(\alpha-\beta)}{2}$ as well. Thus,

$$P(x=y)=\cos^2(\alpha-\beta), P(x\neq y)=\sin^2(\alpha-\beta).$$

At the extremes of $\alpha,\beta=0$ and $\alpha=0,\beta=\pi/2$, the probabilities are $0,1$ and $1,0$.

Also, for $\alpha=\beta=\pi/4$, we are measuring in the $|+\rangle,|-\rangle$ basis. Then,

$$|\Phi\rangle=\frac{|+\rangle\otimes|+\rangle+|-\rangle\otimes|-\rangle}{\sqrt{2}}.$$

This is unsettling because we saw "uncertainty" before: if we are certain of a measurement in the $|0\rangle,|1\rangle$ basis, we cannot be certain of the measurement in the $|+\rangle,|-\rangle$ basis. However, we *can* have *correlation* in both bases simultaneously.

8.2 Bell's Experiment

Bell showed that there is an experiment that can confirm the theories of quantum mechanics, and distinguishes between the behavior of random bits and qubits.

We consider the CHSH game, which is a similar experiment. Consider Alice and Bob, who share either randomness (random bits), or one copy of $|\Phi\rangle$. These two possibilities can in fact be distinguished.

- **Randomized bits.** Consider $a,b\in\{0,1\}$ to be random bits. Alice and Bob then need to both output a bit, x from Alice and y from Bob. Their goal is that $x\oplus y=ab$[†] Alice and Bob can win with probability $3/4$, by setting x and y to be the same. For any deterministic strategy, 75% is the best possible winning probability, which shows that probabilistic strategies cannot achieve higher.
- **Entangled bits.** The EPR pair allows them to beat $3/4$. Depending on the inputs a,b, Alice and Bob can choose their measurement angles, measure their states, and let x and y be their outputs. For Alice, if $a=0$, she chooses $\alpha=0$, and if $a=0$, she chooses $\alpha=\pi/4$. For Bob, if $a=0,\beta=\pi/8$, and if $b=1,\beta=-\pi/8$. Now,
$$P[x=y]=\cos^2(\alpha-\beta).$$
If the input is $(0,0),(0,1),(1,0)$, then $P[x=y]=\cos^2(\pi/8)$. If the input is $(1,1)$, then $P[x=y]=\cos^2(3\pi/8)=\sin^2(\pi/8)$. Then $\cos^2(\pi/8)\approx 0.857$. Somehow, this is better than the randomized strategy.

Rather than "shared randomness," the original paper called it "local hidden variables," which indicates that there were correlated, hidden variables that we simply could not access. However, this rules out that random variables that are controlled by some eavesdropper Eve. This is useful for quantum cryptography.

8.3 Classical Gates and Computing

Classical computing can be expressed through various models:

- Von Neumann: There is a computer with CPU, RAM, I/O, which is what our computers are like, but it can be more complicated or confusing to reason about.
- Turing machine: There is a finite state machine and a tape that it moves along. The Turing machine can compute anything that a Von Neumann machine can.
- Cellular automata: cells communicate based on their neighbors, and this is also equivalent to other models.
- Circuits: circuits use gates and wires, with AND and OR gates. They are easy to program but also easy to reason about.

[†] Here $\oplus$ is the XOR operation, where if $a,b=(1,1)$ then $x\neq y$; otherwise $x=y$.

9 Gates, Universality, and Reversible Computing

9.1 Universality in Classical Computing

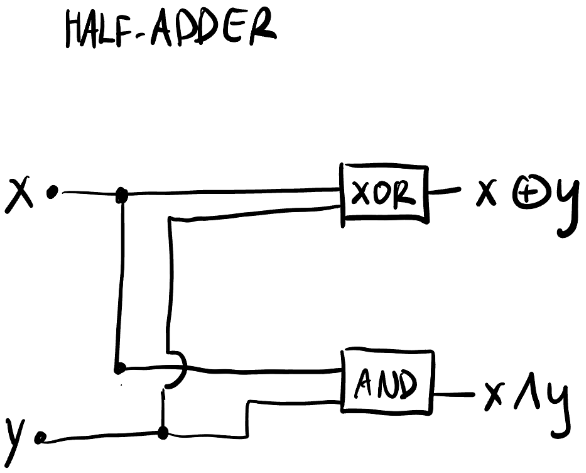
Consider these gates in the circuit model.

input bits	output bits	examples
1	1	I, NOT
2	1	AND, OR, XOR, NAND, NOR
1	2	FANOUT
2	2	CNOT, SWAP
3	3	CCNOT, CSWAP

What gates are reversible? How much information is created or destroyed from a particular gate? Clearly, if the number of output bits is less than the number of input bits, the gate cannot be reversible. However, FANOUT, which has more output bits than input bits, clearly preserves the amount of information.

In the past, these were thought of before quantum computing. By not creating or destroying any information, they don't produce any waste heat. In a normal computer, bits are being erased, and when they are stored in RAM, heat is continually being dissipated. Any fluctuation from a stable configuration is driven back to the stable configuration. However, the very niche "physics of computing" field showed that there is no lower bound on the amount of energy that a computer needs to dissipate.

Consider the following gate.



Example 9.1 (HALF-ADDER gate)
The HALF-ADDER gate adds the two input bits. Given inputs (x, y), it has output

x + y	
0 + 0	00
0 + 1	01
1 + 1	10

In binary, the output is the sum of the bits x and y.

Guiding Question
Which functions can we compute using gates and circuits?

In fact, not only can HALF-ADDER be constructed, but any function can be constructed using circuits and gates.

Theorem 9.2 (Universality Theorem)
Any function can be made from AND, OR, and NOT, assuming FANOUT and ERASE are free.

Proof. Consider $f : \{0,1\}^n \longrightarrow \{0,1\}^m$. Without loss of generality, let $m = 1$. We can make an input-output table with all of the 3-bit strings and their outputs. This specifies the function f.

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	1	1

Then, we have $f(x) = 1$ if x matches any of the rows that has 1 as its output. For example, in this case, we would check that $x = (0,0,0)$ OR $(1,1,1)$ OR any of the rows that has 1 as its output.[*] Next, in order to check if $x = (0,0,0)$, for example, we can take $(\text{NOT}x_1)$ AND $(\text{NOT}x_2)$ AND $(\text{NOT}x_3)$. In general, we can take an AND of x_i if we want to check if the ith bit is 1 and NOTx_i if we want to check if the ith bit is 0.

Overall, for any f, we can write

$$f(x) = \bigvee_{y \in f^{-1}(1)} \bigwedge_{i=1}^{n} \begin{cases} x_i & \text{if } y_i = 1 \\ \neg x_i & \text{if } y_i = 0 \end{cases},$$

which will yield f in terms of AND, OR, and NOT.[†] □

In fact, we can do better.

Corollary 9.3
Any function can be made from OR and NOT, or AND and NOT.[a]

[a] We might ask, "can we omit any more gates?" In fact, if we only have OR and AND, we have a class of circuits called "monotone circuits." That is, if we flip a bit from 0 to 1, the output will never flip from 1 to 0. Actually, $NAND(x,y) = \neg(x \wedge y)$ can produce AND and NOT, so NAND suffices to compute any function.

Proof. We can write AND as $x \vee y = \neg(\neg x \wedge \neg y)$. □

9.2 Measuring Complexity in Classical Computing

However, being able to compute a function is not the end of the story.

Guiding Question
How efficiently can we compute a particular function? What is a measure of efficiency?

Some ways of measuring complexity include:

- Size: The number of gates, or "the total electricity bill"
- Depth: The number of layers, or the "wall-clock time"
- Width: The maximum number of bits used at any point, or the amount of memory needed

Often, there is a tradeoff between these different measures. Sometimes, by increasing the depth, the size can be greatly reduced.

Up to now, we have been considering $f : \{0,1\}^n \longrightarrow \{0,1\}$, which is analogous to multiplying 32-bit numbers. However, we may want consider the asymptotic complexity of a function with inputs of any size, such as

[*] Here, we think of 0 as FALSE and 1 as TRUE.

[†] This is $O(n \cdot 2^n)$ elementary gates, but we are not concerned with complexity, only computability. Using a counting argument, we can show that the best is $O(2^n)$, which can be achieved using a slightly different construction.

"multiplication" in general. We write $\{0,1\}^* = \bigcup_{n\geq 0}\{0,1\}^n$, which consists of strings of bits of any size. For asymptotic complexity, we want to instead consider functions

$$f : \{0,1\}^* \longrightarrow \{0,1\},$$

and consider how the complexity grows as the number of input bits grows.

Definition 9.4 (Big-O Notation)
A function $F(n)$ is $O(G(n))$ if there exists some c, n_0 such that for all $n \geq n$,

$$F(n) \leq cG(n).$$

We write $F(n)$ is $\Omega(G(n))$ if G is $O(F(n))$, and we write $F(n)$ is $\Theta(G(n))$ if both O and Ω are true.[a]

[a] In math and physics, we typically write O to mean what computer scientists write as Ω, which means that F grows as $G(n)$. In computer science, which is the convention we adopt for this class, for F to be $O(G(n))$, we simply require that F is upper bounded as $G(n)$, and F may grow much slower than G.

Common asymptotics include $O(n), O(n^2)$, or $O(2^n)$. This notation abstracts away unnecessary constants, and ignores the behavior when n is small.

Question. *What if we use different circuits for different inputs?*

Answer. *Lurking in the background of this is that we might use a different circuit for 32 and 33 bit numbers, so we need to consider the complexity of "preparing" the circuit. The way we abstract this out is by saying there is a little Turing machine in the background that takes in a number, say 33, and prepares a circuit, say to multiply 33-bit numbers, and this Turing machine should not take too long. This is an interesting question/theory, but it will not come up in our discussion of circuits, since this is not a complexity theory class.*

Example 9.5 (Addition)
To add n-bit numbers, we need 2 HALF-ADDERs (we need two to compute the carry over bit), which has c gates for some constant c for each digit, so adding can be roughly $O(2n) = O(n)$ complexity in terms of the number of gates, in this formulation. That is, the size is $O(n)$.

Example 9.6 (Multiplication)
We can reduce multiplying n bit numbers to n addition problems, which yields a runtime of $O(n^2)$. In fact, we can use a smarter method to get a runtime of $O(n \log n)$.

Overall, we can write $n^{O(1)}$, which is n to some constant, as $POLY(n)$. At the crudest level, we would want to distinguish between polynomial time, and exponential time, and times in between.

10 Reversible Gates and Quantum Gates

Quantum mechanics is reversible, so we will study quantum circuits made from reversible quantum gates. To understand these reversible quantum gates, we can first study reversible classical gates. Consider gates such as $x_1, x_2 \mapsto f(x_1, x_2)$ or $x_1, x_2, x_3 \mapsto f(x_1, x_2, x_3)$.

Definition 10.1
A gate $f : \{0,1\}^n \longrightarrow \{0,1\}^n$ is **reversible** if f^{-1} exists.

Example 10.2
Reversible gates include I, NOT, and CNOT[a].

[a] Recall that $CNOT(x, y)$ maps x to x and y to $\neg y$ if $x = 0$.

To construct a quantum gate, note that

$$\sum_x |f(x)\rangle \langle x|$$

is always stochastic, and in fact is unitary if f^{-1} exists (Fredkin).

Definition 10.3
The Toffoli gate, or the CCNOT gate, has 3-bit inputs and outputs. If the first two bits are both 1, it inverts the third bit, and otherwise all the bits stay the same. That is, $CCNOT(x, y, z) = (x, y, z \oplus (x \wedge y)$, where $\oplus$ is binary addition or XOR.

The truth table of the CCNOT gate is:

(x, y, z)	$f(x, y, z)$
000	000
010	010
100	100
110	111
001	001
011	011
101	101
111	110

Claim 10.4. *The NOT and CCNOT gates are universal.*

Proof. We can generate AND from CCNOT by taking $CCNOT(x, y, 0) = (x, y, x \wedge y)$, and we can generate FANOUT from $CCNOT(1, x, 0) = (1, x, x)$. From last time, NOT, AND, and $FANOUT$ suffice for universality. □

This does not violate the no-cloning theorem because

$$|0\rangle \longrightarrow |00\rangle, |1\rangle \longrightarrow |11\rangle,$$

but

$$|+\rangle \longrightarrow \frac{|00\rangle + |11\rangle}{\sqrt{2}} \neq |+\rangle \otimes |+\rangle.$$

10.1 Cleaning up garbage bits

Unfortunately, there is a lot of "trash" or "garbage" left over in this paradigm, since we keep track of a lot of past bits.

Suppose $x \longrightarrow f(x)$ is possible with irreversible gates. With reversible gates $x \longrightarrow x, f(x), g(x)$, we can do controlled maps, such as CNOT. Erasing is not trivial, and cannot be done for free, due to entropy and energy considerations.

Let us have a register of x, and several 0s. With reversible gates, we can first compute $x \longrightarrow x, f(x), g(x), 0$. Then, taking $CNOT$ of $f(x), g(x)$, and 0, we will have $x, f(x), g(x), f(x)$. Now, we can undo this transformation, to get $x, f(x)$. This only works with $x, f(x)$, and $g(x)$, and may not work with just $g(x)$. We are not writing the padding with zeroes.

If f is reversible, we might be able to do $x \mapsto x_1, f(x)$ efficiently, but not $x \mapsto f(x)$.

If we have an algorithm to compute $f(x)$ using AND, OR, and NOT, we can compute $x, f(x)$ as well using reversible gates.

10.2 NOT and CNOT are not universal

However, $\{NOT, CNOT\}$ is not universal. Consider

$$\{0,1\} \longrightarrow \mathbb{F}_2.^{*}$$

Then, we can consider $\{0,1\}^n \longrightarrow \mathbb{F}_2^n$, where $\mathbb{F}_2^n$ is a vector space. From the field perspective, $NOT(x) = x + 1$. Similarly,

$$CNOT \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ x+y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Another example is $NOT_2 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$

Thus, any sequence of NOT and CNOT is an **affine transformation**

$$x \in \mathbb{F}_2^n \mapsto Ax + b.$$

In particular, there is no way to obtain AND, since AND is the product of two bits. Using a counting argument, there are $n^2 + n$ bits describing A and b, but there are 2^n bits describing functions. That is why we need CCNOT, as CNOT is not sufficient.

The reason CCNOT does not have the same limitation is because CCNOT takes $(x, y, z) \mapsto (x, y, z + xy)$, and so we can obtain polynomials, not just affine transformations.

10.3 Quantum Gates

In the world of quantum gates, we use unitary matrices. For qubits,

$$U(2) = \{2 \times 2 \text{ unitary matrices}\}.$$

For qudits,

$$U(d) = \{d \times d \text{ unitary matrices}\},$$

which are the operators that act as quantum gates.

Note that if we have $|\psi\rangle$, $e^{i\theta}|\psi\rangle$ will have the same behavior when measured. Thus, the "phase" degree of freedom is not relevant.

Example 10.5 (Pauli Gates)
The NOT gate, which does a bit flip, is denoted

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \sigma_x = \sigma_1.$$

The Y gate is denoted

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \sigma_y = \sigma_2.$$

The phase flip gate is denoted

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \sigma_z = \sigma_3.$$

We can think of the Y gate as doing a bit flip, then a phase flip, up to a phase of i.

*The field with two elements, where addition and multiplication are considered modulo 2.

We might try to produce other gates, as in the classical case, with these Pauli gates as "building blocks." Unfortunately, this does not work.

Claim 10.6. *The set $\{\sigma_x, \sigma_y, \sigma_z\}$ of Pauli gates are* not *universal.*

Proof. In fact, they are extremely limited: compositions of Pauli gates produce another Pauli gate (up to a phase shift). We have $XY = iZ, YX = -iZ, YZ = iX = -ZY$, and $ZX = iY = -XZ$. Moreover, $X^2 = Z^2 = Y^2 = I$. Moreover, conjugating[†] a gate by another gate flips the sign, $XYX = X(-XY) = -Y$. In general, for $i \neq j$, $\sigma_i \sigma_j \sigma_i = -\sigma_j$. Thus, the set of Pauli gates do not generate all gates. □

Example 10.7
Recall the Hadamard gate

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

We have $H^2 = I$, and moreover the Hadamard gates play nicely with the Pauli gates: $HXH = Y, HZH = X$, and $HYH = iHXZH = iHXHHZH = iZX = i^2 Y = -Y$. Thus, $\{I, X, Y, Z, H\}$ is also not universal.

Next time, we will talk about what *is* universal. The Euler angle decomposition states that any $U \in U(2)$ can be written as $U = e^{i\phi} R_z(\alpha) R_x(\beta) R_z(\gamma)$, where R_x and R_z are rotations around the x and z axes. We can define $R_j(\theta) = e^{i\theta\sigma_j}$. This decomposition is analogous to the case of 3D real rotations. Using these rotations, we can build up any 2×2 unitary, and obtain universality for the 2×2 case.

[†] Conjugating by A means taking $A^{-1}XA$, and here $A^{-1} = A$.

11 Single-Qubit Gates

We already talked about universal gate sets for classical gates. Today, we will discuss

- Universality for single-qubit gates
- Universality for multi-qubit gates

11.1 Bloch Sphere

Recall that $U(2)$ is the set of 2×2 unitary matrices (with complex entries). These unitary matrices are the operations that act on qubits; that is, $U(2)$ is the set of all single-qubit gates. It turns out that $U(2)$ is secretly the same as rotations and reflections of the *Bloch sphere*, which is a sphere in $\mathbb{R}^3$, or 3-dimensional real vectors.

Definition 11.1
Recall the Pauli matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}.$$ [a]

[a] A mnemonic is that "minus i flies high."

Note that the Pauli matrices are unitary. These matrices satisfy $\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I$. Moreover, note that

$$\sigma_x\sigma_y = i\sigma_z, \sigma_z\sigma_x = -i\sigma_z,$$

and all cyclical permutations of these are true. We can write

$$\sigma_i\sigma_j = \delta_{ij}I + i\varepsilon_{ijk}\sigma_k$$

Definition 11.2 (Levi-Civita symbol)
The **Levi-Civita symbol** is $\varepsilon_{ijk} = \begin{cases} +1 & \text{if } ijk = xyz, zyx, yzx \\ -1 & \text{if } ijk = yxz, zyx, xzy \\ 0 \end{cases}$.

Essentially, if any of the indices repeat, ε is 0, cyclical permutations of xyz are 1, and cyclical permutations of zyx are -1.

Given a 3-dimensional vector $v \in \mathbb{R}^3$, we can construct a matrix $v \cdot \sigma = v_x\sigma_x + v_y\sigma_y + v_z\sigma_z \in \mathbb{C}^{2\times2}$.

Proposition 11.3
The eigenvalues of $v \cdot \sigma$ are $\pm|v|$.

Proof. We can check that

$$(v\cdot\sigma)^2 = \left(\sum v_i\sigma_i\right)\left(\sum v_j\sigma_j\right) = \sum_{ij}(v_iv_j)(\sigma_i\sigma_j).$$

Plugging in the identity for $\sigma_i\sigma_j$, we have

$$= \sum_{ij} v_iv_j\left(\delta_{ij}I\right) + v_iv_j\left(\sum_k \varepsilon_{ijk}\sigma_k\right),$$

and since δ_{ij} is only nonzero when $i = j$, we end up with

$$= |v|^2I + v_iv_j\left(\sum_k \varepsilon_{ijk}\sigma_k\right).$$

From the property of the epsilon symbol, we have $\sum_{ijk} \varepsilon_{ijk} v_i v_j \sigma_k = -\sum_{ijk} \varepsilon_{jik} v_j v_i \sigma_k$, since $\varepsilon_{ijk} = -\varepsilon_{jik} = -\sum_{ijk} \varepsilon_{ijk} v_i v_j \sigma_k$. Thus, this sum is $\sum_{ijk} \varepsilon_{ijk} v_i v_j \sigma_k = 0$; this exploits the antisymmetry of the ε symbol. Thus, this becomes

$$= |v|^2 I.$$

Therefore,

$$(v \cdot \sigma)^2 = |v|^2 I.$$

In the eigenbasis, we have $\begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix}^2 = \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} = \begin{pmatrix} |v|^2 & \\ & |v|^2 \end{pmatrix}$. Thus, taking the square root shows that $\pm|v|$ are the eigenvalues of $v \cdot \sigma$. □

Now, we want to know the eigenvectors. The trace of each of the Pauli matrices is zero, so since the trace is linear, $v \cdot \sigma$ has trace 0. We can call the eigenvectors $|\pm v\rangle \in \mathbb{C}^2$. A vector in spherical coordinates is written as $\begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}$. We can calculate the eigenvectors.

Proposition 11.4
The eigenvector $|v\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\frac{\sin\theta}{2}|1\rangle$ has eigenvalue $+|v| = +1$ for $\sigma \cdot v$. Also, $|-v\rangle = \sin\frac{\theta}{2}|0\rangle - e^{i\phi}\cos\frac{\theta}{2}|1\rangle$ is an eigenvector with eigenvalue $-1 = -|v|$.

Then, we see that v in the Bloch sphere is equivalent to $|v\rangle \in \mathbb{C}^2$. The parametrization of the Bloch sphere is $v = \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}$.

Example 11.5
Here, $\hat{v}$ is a name for the ket, and $|v\rangle$ is the ket, which is two-dimensional.

$\vec{v}$	θ	φ	$\lvert v\rangle$
$\hat{z}$	0	?	$\lvert 0\rangle$
$\hat{-z}$	π	?	$\lvert 1\rangle$
$\hat{x}$	$\pi/2$	0	$\lvert +\rangle$
$\hat{-x}$	$\pi/2$	π	$\lvert -\rangle$
$\hat{y}$	$\pi/2$	$\pi/2$	$\lvert i\rangle$
$\hat{-y}$	$\pi/2$	$\pi/2$	$\lvert -i\rangle$

In 3D space, the angle between v and $-v$ is π. In ket space, the angle between z and $-z$ is $\pi/2$ (the angle is defined in terms of the inner product). The angle is cut in half or doubled when going between spaces. So the ket space is called the "spin 1/2 representation of $SU(2)$." That is, if $|v\rangle, |w\rangle$ has an angle α between them, then v, w has angle 2α. In this parametrization, global phases vanish in the Bloch sphere.

Intuitively, $|\psi\rangle$ has two complex numbers, which is the same as four real numbers, and then there are only three real numbers because the complex vectors are unit vectors, and lastly ignoring global phase yields two real numbers. That is why the Bloch sphere is a 2-dimensional manifold.

Question. *Why are we doing this?*

Answer. *Because there is a very rich theory of decomposing 3D rotations, using the Bloch sphere, we can use these decompositions of 3D rotations for unitary operations on unit complex vectors.*

11.2 Rotations and Gates

By calculation, unitary matrices can be decomposed into points on the sphere, with extra angles.

Theorem 11.6
Any unitary matrix $U \in U(2)$ can be decomposed as

$$U = e^{i\alpha}(\cos\beta I + i\sin\beta v \cdot \sigma),$$

where v is a unit vector.

Then, U in this form becomes a rotation around the v-axis, by angle β. Since the global phase vanishes, α does not matter.

Corollary 11.7
All single-qubit gates are rotations in 3 dimensions.

Rotations in 3D have been studied for hundreds of years, so this is great, since we now know how to decompose these unitary matrices/rotations.

11.3 Universality

Since there are infinitely many rotations in 3D, there is no possible way to use a finite number of gates to produce all rotations in 3D. Thus, the notion of universality has to change for these continuous rotations.

Definition 11.8
A set of gates is universal if any rotation can be approximated arbitrarily well by products of these gates.[a]

[a] It would also work to allow the set of gates to be continuously parametrized, but this way makes the gates more "digital."

Most gate sets are universal, avoiding symmetry groups inside of $SU(2)$. For example, gates that all rotate around the same axis, or gates that always preserve the xy-plane will not work.

Example 11.9
The gate set

$$G = \{H, R_z\theta\},$$

for any θ such that θ/π is irrational, is universal.

Here, $R_z(\theta) = \cos\theta I + i\sin\theta\sigma_z$.

12 Multi-Qubit Gates

Today, we will talk about:

- Multi-qubit universality
- Discrete universal gate sets

12.1 Review

We can exponentiate matrices either applying it to the eigenvalues, if the matrix is diagonal, or take a Taylor series. We can write

$$e^{i\theta\sigma_j} = \sum_{n\geq 0} \frac{(i\theta\sigma_j)^n}{n!} = I + i\theta\sigma_j - \frac{\theta^2}{2}\sigma_j^2 - \frac{i\theta^3}{6}\sigma_j^3 + \frac{\theta^4}{24}\sigma_j^4 + \cdots,$$

and using $\sigma_j^2 = I$, this yields

$$= \left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} + \cdots\right) I + \left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \cdots\right) i\sigma_j = \cos\theta I + i\sin\theta\sigma_j = R_j(\theta).$$

Then, for $|v| = 1$, recall that we calculated that $(v\cdot\sigma)^2 = |v|^2 I = I$. Thus,

$$e^{i\theta v\cdot\sigma} = \sum_{n\geq 0} \frac{(i\theta)^n}{n!}(v\cdot\sigma)^n = \cos\theta I + i\sin\theta v\cdot\sigma = R_v(\theta).$$

This is a rotation around the v-axis.

Therefore, any $U \in U(2)$ can be written as $U = e^{i\phi}R_v(\theta)$. So up to a phase, unitary matrices are the same as rotations up to a phase.

Note 12.1

We have $U(2) = 2\times 2$ unitaries. However, we say that the overall phase "doesn't matter", because $|\langle v_i|e^{i\phi}|\psi\rangle|^2$ does not depend on ϕ. In reality, no outcome will ever be different between two states that differ by an overall phase. We can thus declare states that differ by an overall phase to be the same state (we tend not to actually do this because it makes the math simpler). Then, $SU(2) = \{U \in U(2) : \det U = 1\}$, where $I \in SU(2)$. This removes the degree of freedom that comes from a global phase.

12.2 Universality of Multi-Qubit Gates

Here, CNOT means a CNOT between any two pairs of qubits.[*]

Claim 12.2. *CNOT and arbitrary single-qubit gates are universal.*

In place of CNOT, essentially any entangling 2-qubit gate can be used instead. The important idea is that this can be done at all.

We can think of CNOT as a C-X[†] gate, where the C-U gate can be written as $|0\rangle\langle 0| I + |1\rangle\langle 1| \otimes U = \begin{pmatrix} I & 0 \\ 0 & U\end{pmatrix}$ as a block matrix. Consider $(I\otimes U)CNOT(I\otimes U^\dagger)CNOT$.

Write $U = R_z(\theta)$ and $U^\dagger = R_z(-\theta) = e^{-i\theta Z}$. Then $XU^\dagger X = e^{-i\theta XZX} = e^{i\theta Z}$. We have $Ae^BA^{-1} = e^{ABA^{-1}} = \sum_n \frac{(ABA^{-1})^n}{n!} = \sum_n \frac{AB^nA^{-1}}{n!}$.

In this case,

$$XU^\dagger XU = U^2 = R_z(2\theta).$$

That is, doing an X before and after a z-rotation reverses the direction of the rotation.

[*] Note that in the homework, we could do SWAP with CNOT, so it's only technically necessary to have CNOTs between qubit i and qubit $i+1$.

[†] Here the X gate is the SWAP gate with matrix $\begin{pmatrix} 0 & 1 \\ 1 & 0\end{pmatrix}$.

Thus,

$$= C - XU^\dagger XU = C - R_z(2\theta)$$

if $U = R_z(\theta)$.

Moreover, for $U = R_x(\theta)$, $UXU^\dagger X = I$. Also, if $U = R_y(\theta)$, then $UXU^\dagger X = R_y(2\theta)$.

Therefore, for all U, there exist α, β, γ such that

$$C - R_z(\theta)C - R_y(\beta)C - R_z(\gamma) = C - U.$$

That is because we can write U as an Euler angle decomposition. Note that $(C - U)(C - V) = C - UV$. We can write $C - (e^{i\phi}I) = \begin{pmatrix} I & 0 \\ 0 & e^{i\phi}I \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} \otimes I = e^{i\phi/2}R_z(-\phi/2) \otimes I$.

That is, this is a z-rotation on the first qubit and nothing on the second qubit. Thus, we can do a controlled U for any U.

Now, let's go beyond two qubits. We have $CC\text{-}U$ to be

$$CC\text{-}U = (|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I_2 + |11\rangle\langle 11| \otimes U).$$

Then, $C^k\text{-}U$ can be done by "do U if the first k qubits are $1^k = \underbrace{11\cdots 1}_{k \text{ times}}$.

Given $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, we have

$$CC\text{-}U = \begin{pmatrix} I & & \\ & a & b \\ & c & d \end{pmatrix}$$

or

$$\begin{pmatrix} a & & b & \\ & 1 & & \\ c & & d & \\ & & & 1 \end{pmatrix},$$

where most of it looks like the identity, where there is some with a, b, c, and d. This is what is needed for Gaussian elimination, where these $CC - U$ operations form the "elementary operations." If this is possible, then any nonsingular matrix can be constructed.

Therefore,

$$UT_1 \cdots T_m = I,$$

where T_i are two-level rotations. Therefore,

$$U = T_m^\dagger \cdots T_1^\dagger$$

is also a series of two-level rotations. This is not very efficient: doing this requires about 4^n rotations, which leads to $O(n4^n)$ gates. We saw the same idea for circuits, where our construction leads to $O(n2^n)$ time, but generally we are not constructing arbitrary circuits.

To actually make these, for our example 000 and 011, we need a permutation P such that $P|000\rangle = |110\rangle$ and $P|011\rangle = |111\rangle$. Then, we take $P^\dagger CC\text{-}UP$.

Up to an arbitrary phase, we can always write a ket as $(\cos\theta/2, e^{i\phi}\sin\theta/2)$ and we can correspond ig to the Bloch sphere.

13 Approximate Universality and the Oracle Input Model

13.1 Review

In order to act on the last state of an n-bit string with a unitary U, last time we constructed $C^{n-1}U$, a multiply-controlled unitary, which mixes states $|1^{n-1}0\rangle$ and $|1^n\rangle$.

In general, consider a two-level rotation[*] on $|x\rangle$ and $|y\rangle$, where $x, y \in \{0,1\}^N$. Consider a permutation P such that $P|x\rangle = |1^{n-1}0\rangle$ and $P|y\rangle = |1^n\rangle$. Then, taking $P^\dagger C^{n-1}UP$ mixes $|x\rangle$ and $|y\rangle$ instead.

Example 13.1

Let $x = 0110$ and $y = 1010$. We want to take $x \mapsto 1110$ and $y \mapsto 1111$, and to do so, we can first take $x = 0110 \mapsto 0001 \mapsto 1110$, and then $y = 1010 \mapsto 0000 \mapsto 1111$. First, apply $X_1X_2 = X \otimes I \otimes X \otimes I$, which is a NOT on the first and third bits. This takes y to 0000, and

$$X_1X_2\,|0110\rangle = |1100\rangle\,.$$

Next, we can then apply CNOTs to clean up x while maintaining y. We can apply $CNOT_{14}$ which takes $|1100\rangle$ to $|1101\rangle$, then $CNOT_{41}$ which takes $|1101\rangle$ to $|0101\rangle$, then $CNOT_{42}$, which finally takes $|0101\rangle$ to 0001.

Thus, this composition $CNOT_{42}CNOT_{41}CNOT_{14}X_1X_2$ takes $x \mapsto 0001$ and $y \mapsto 0000$. Now, it suffices to apply C^3U, where U takes $0001 \mapsto 1110$ and $0000 \mapsto 1111$.

This is clearly very inefficient, but we are simply proving that CNOT and single-qubit gates together are universal. In practice, we will focus on finding efficient algorithms to perform a multi-qubit gate.

13.2 Approximate Universality for Single-Qubit Gates

In fact, two gates suffice for approximate universality in quantum computing.

Proposition 13.2

The gate $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{-\pi/4} \end{pmatrix}$ and H are universal for $U(2)$, ignoring phase.[a]

[a] We can alternatively write $SU(2)$, but then we would have to add a phase to T and H to put them in $SU(2)$.

The set $\{T, H\}^*$, which is the set of operations coming from applying T and H each some finite number of times, are countable. However, $U(2)$ is uncountable, so it is impossible to precisely approximate each of the uncountable matrices in $U(2)$ with the countably many matrices in $\{T, H\}^*$.

Definition 13.3

A gate set G is **approximately universal**, or **universal** for $U(2)$ if for all $U \in U(2)$ and for all $\varepsilon > 0$, there exists a string of $V_1, \cdots, V_L \in G$ such that

$$d(U, V_LV_{L-1}\cdots V_1) \le \varepsilon.$$[a]

[a] Any reasonable metric on matrices will work as a choice for d.

Guiding Question

How does L scale with ε?

That is, how many operations does it take to approximate any single-qubit gate to a particular accuracy?

[*] A two-level rotation is of the form $\begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & a & & b & \\ & & & 1 & & \\ & & c & & d & \\ & & & & & 1 \end{pmatrix}$, which acts trivially on all but two bit-strings.

Example 13.4
We can write

$$SU(2) = \{\cos\theta I + i\sin\theta v \cdot \sigma\},$$

where $(\cos\theta, \sin\theta v)$ is on the 3-sphere in $\mathbb{R}^4$. Let the gate set be T and H. Then, each string of T and H will be a point on the 3-sphere $SU(2)$. By drawing a ball of radius ε around each of the points that we are considering, say points in $\{T, H\}^L$ (strings of length exactly L), if the balls cover the 3-sphere $SU(2)$, then this means that each gate in $SU(2)$ can be approximated to ε-accuracy with gates in $\{T, H\}^L$.

Then, take the volume of the 3-sphere to be some constant C. The volume of the ball of radius ε, intersected with the 3-sphere, is $C'\varepsilon^3$, ignoring phase. To obtain a lower bound, we take the best possible case where the points are equally spaced and the balls don't overlap. Covering $U(2)$ requires L, where we take $\{T, H\}^L$ to satisfy at least

$$2^L C'\varepsilon^3 \geq C.$$

That is,

$$L \geq \Omega(\log(1/\varepsilon)).$$

We also want to talk about achievability. For specific gate sets, we can in fact achieve a reasonable bound. This is described by the Solovay-Kitaev-Kuperberg theorem, which states that

$$L \leq O(\log^{1.44}\frac{1}{\varepsilon})$$

suffices.

Failure to be universal is a set of polynomial equations, which are obeyed by a set of measure 0. Because one gate commutes with itself, the scaling is much worse with L, only L, rather than around 2^L for two gates.

Example 13.5 (Rotations around one axis)
Recall that $R_z(\sqrt{2}\pi m) = R_z(\sqrt{2}\pi m \bmod 2\pi)$ can approximate almost any $R_z(\theta)$, and taking $m = 0, \cdots, L$ shows that $L \sim \frac{1}{\varepsilon}$. That is, to approximate $R_z(\theta)$ with ε-accuracy takes $O(1/\varepsilon)$ length.

Question. *Why do we constrain ourselves to discrete gate sets, rather than continuous?*

Answer. *It is true that in real life, the quantum operations we have are continuous. However, this will have applications to quantum error-correction.*

13.3 Introduction to Quantum Algorithms

First, we will talk about input models. A natural input model is the *standard input model.*

Definition 13.6 (Standard Input Model)
In the **standard input model**, an input $x = (x_1, \cdots, x_n)$ is a string of bits, and the quantum computer is initialized to $|x\rangle \otimes |0\rangle^{\otimes i}$, where the 0s are extra working space.

Another potential model is the oracle model, which encodes functions of functions, such as summing the outputs, or taking the derivative.

Definition 13.7
In the oracle model, the input is encoded by a function $f : \{0,1\}^n \to \{0,1\}$, or a truth table, rather than a string. The operations on the input are $F(f) = F(f(000), f(001), \cdots, f(111))$.

In the oracle model, there is on-demand access to the output of the function f given the input. This could be implemented in the standard input model, but in that case every value of f would have to be stored, which is not very efficient. Either we can say $x \to f(x)$ is a subroutine that we have the source code for, or we could say we have access to a black box "oracle" that computes f.

In the oracle model, we can define the *query complexity*, which can be much easier to think about than the standard computational complexity.

Definition 13.8

The **query complexity** of a function F in the oracle model is the number of calls to f are needed.

Example 13.9 ($n = 1$)

Consider an input $f : \{0, 1\} \longrightarrow \{0, 1\}$. We want to compute $F(f)$, which returns whether f is constant, or balanced. That is, F returns whether $f(0) = f(1)$. The query complexity is 2, since it requires one call for $f(0)$ and one call for $f(1)$.

14 Oracles, Deutsch's Algorithm, and Bernsetin Vazirani

14.1 Review

Recall that in the oracle input model, the input is not a bitstring, but rather a black-box function $f : \{0,1\}^n \longrightarrow \{0,1\}$. The output would be some function of this function, such as the sum of all its outputs, or some property of this function.

Example 14.1 (Classical oracles)
On a classical computer, the oracle input model may mean either that we have the source code for f, or we have a black-box way to compute f.[a] A black-box model would make sense in a network case, where we ask how many queries it will take to solve the question.

[a] These two may seem very different, but there is a basic principle, which is not formalized here, from complexity theory, which is that in general there is "no way" to "analyze" source code. In special cases, there may be a way to look at the source code and figure out what it will do, but in general, there is no way to look at the source code and analyze it, other than just running it, which is equivalent to a black box way to compute the function. One way of formalizing this principle is the *halting problem.*

Recall from before that unitary quantum gates come from reversible classical operations.

Proposition 14.2
Reversible operations produce unitary gates.

Proof. A reversible or one-to-one classical operation is a permutation. Consider $U_p = \sum_{x\in\{0,1\}^n} |P(x)\rangle \langle x|$, where P is a permutation of $\{0,1\}^n$. Then, $U_p^\dagger U_p = \sum_x |x\rangle \langle P(x)| \sum_y |P(y)\rangle \langle y|$. Since $\langle P(x)|P(y)\rangle = \delta_{P(x),P(y)} = \delta_{xy}$, using the fact that P is a permutation. Thus, $U_p^\dagger U_p = \sum_{x,y} \delta_{xy} |x\rangle \langle y| = I$. □

Example 14.3 (Bit-flip Quantum Oracle)
One way to create a quantum oracle O_f^{bit} is to decompose the source code for the oracle into classical ANDs and NOTs, and then convert those quantum Toffolis. Then,

$$O_f^{bit} |x\rangle \otimes |0\rangle = |x\rangle \otimes |f(x)\rangle .$$

A more complete definition on a basis would be

$$O_f^{bit} |x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle ,$$

where $\oplus$ denotes XOR or alternatively addition modulo 2. This definition on a basis extends by linearity to all inputs, including entangled states.

This gate is unitary, since the classical gate it comes from is reversible. The classical gate $x, y \mapsto x, y \oplus f(x)$ is reversible since applying it twice is the identity. To recover the classical oracle, we measure the second bit.

Example 14.4 (Phase-flip Quantum Oracle)
Another way to create a quantum oracle is

$$O_f^{phase} |x\rangle = (-1)^{f(x)} |x\rangle .$$

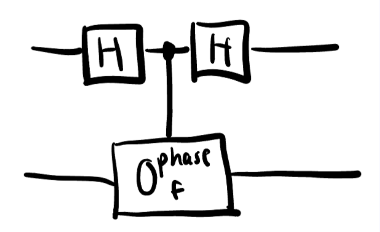
Again, this is unitary.

Given O_f^{bit}, we have

$$O_f^{bit} |x\rangle \otimes |-\rangle = (-1)^{f(x)} |x\rangle \otimes |-\rangle = (O_f^{phase} |x\rangle) \otimes |-\rangle .$$

[a]

To go from phase to bit-flip, it requires a controlled version of the phase-flip oracle. Performing a Hadamard before and after a controlled phase-flip oracle yields a bit-flip oracle.



[a] Note that $X |-\rangle = -|-\rangle$, and $X |+\rangle = |+\rangle$, which are the eigenstates of the X operator.

The oracle model is essentially an assumption, for quantum computing.

14.2 Deutsch's Algorithm, 1985

Consider $f : \{0,1\} \to \{0,1\}$. Deutsch asked, "Is f constant or balanced?" That is, does $f(0) = f(1)$?

Classically, this requires 2 queries, to $f(0)$ and to $f(1)$.

Example 14.5 (Deutsch's Algorithm)
However, in quantum computing, only one query to $f(|+\rangle)$ is required:

$$O_f^{phase} |+\rangle = \frac{1}{\sqrt{2}} \sum_{x\in\{0,1\}} (-1)^{f(x)} |x\rangle .$$

If f is constant, this corresponds to $|0\rangle + |1\rangle$ and $-|0\rangle - |1\rangle$, which are $\pm |+\rangle$, or $|+\rangle$ up to a phase, and if f is balanced, this corresponds to $-|0\rangle + |1\rangle$ or $-|0\rangle - |1\rangle$, which are $\pm |-\rangle$, which are $|-\rangle$ up to a phase.

This suggests that we should then measure in the $|+\rangle, |-\rangle$ basis. Measuring yields

$$P(+) = \left| \langle + | O_f^{phase} | + \rangle \right|^2 = \left| \frac{1}{\sqrt{2}} \sum_x \langle x | \sum_y (-1)^{f(y)} | y \rangle \right|^2 = \frac{1}{2} \left| \sum_x (-1)^{f(x)} \right|^2 = \begin{cases} 1 \text{ if constant} \\ 0 \text{ if balanced} \end{cases} .$$

This is equivalent to applying the Hadamard gate, which is the change of basis into the $|+\rangle, |-\rangle$ basis, then measuring using the standard measurement.

This algorithm is not actually very practical, as it only yields a 2x speedup, but it leads to a lot of interesting theory.

In 1992, Jozsa came up with an improvement. Consider a generalization of Deutsch's problem, with $f : \{0,1\}^n \longrightarrow \{0,1\}$. We say f is *constant* if the result is always 0 or always 1, and *balanced* if the result is 0 half for half of the inputs and 1 for the other half of the inputs. For $n > 1$, there are many possible f that are neither constant nor balanced. This is called a *promise problem* because we are guaranteed that f is valid, which means that f is either constant or balanced. Equivalently, our algorithm is guaranteed to work when f is valid. This happens all the time in computer science.

- The classical query complexity, with deterministic computing, is very hard. We require $2^{n-1}+1$ queries to state that f is certainly constant or certainly balanced.
- The probabilistic complexity states that for $k+1$ queries, $P(\text{error}) \leq 2^{-k}$. This problem is very hard for deterministic computing but not very hard for probabilistic computing.
- It requires a somewhat strange formulation, but comparing the quantum complexity with the classical deterministic complexity does yield an exponential speedup. To analyze this quantum algorithm, note that the Hadamard gate can be written as

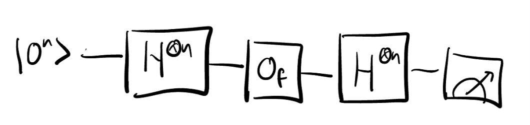
$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \sum_{x,y\in\{0,1\}} (-1)^{xy} \left|x\right\rangle \left\langle y\right| .$$

Applying H to each of the n bits can be written as

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

where we turn bits into strings and multiplication into dot products.

Example 14.6 (Deutsch-Jozsa Algorithm, 1992)
Take the input $\left|0^n\right\rangle$, and apply $H^{\otimes n}$, then O_f,[a] then $H^{\otimes n}$ again, and then measure.



Evaluating the first gate yields

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

Evaluating the second gate then gives

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

Then, applying the last Hadamard yields

$$H^{\otimes n} O_f H^{\otimes n}\left|0^n\right\rangle = \frac{1}{2^n} \sum_{x,y} (-1)^{f(x)+x\cdot y} \left|y\right\rangle .$$

Measuring yields

$$Pr(0^n) = \left| \frac{\sum_x (-1)^{f(x)}}{2^n} \right|^2 .$$

This is essentially the same analysis as Deutsch's algorithm, but on n qubits.

[a] We write O_f to mean O_f^{phase}.

15 Simon's Algorithm

16 Shor's Algorithm

16.1 Classical Factoring

In the past, to factor N, most people have seen an algorithm which could be called "trial division," which means continually trying to divide N by smaller numbers. For a number of size N, there are $\log N$ bits, so polynomial in $\log N$ means "polynomial time" for this kind of problem. The trial division algorithm takes around $\sqrt{N}$ time, which is not polynomial in $\log N$.

The best current factoring algorithm is the *generalized number field sieve* method, which has runtime around $2^{\sim 2 \log^{1/3} N \log\log^{2/3} N}$, which is already subject to some conjectures. With supercomputers, this is still very hard to

In 1994, Shor's algorithm was developed, which takes around $\log^2 N$, subject to some classical preprocessing, and recently in 2023 Regev reduced this time to $\log^{2/3} N$. Note that these are polynomial in $\log N$. This is one of the most promising outcomes of quantum computing, but intellectually, this may not be too surprising.

16.2 Overview of Shor's Algorithm

Shor's algorithm will take more than one lecture to cover.

1. Reduce factoring to period finding. That is, assuming period finding has an algorithm, we can produce an algorithm for factoring. Period finding means that given $f : \mathbb{Z} \longrightarrow S$ with period a, which means that $f(x) = f(x + a)$, period finding finds a. Period finding is an oracle problem.
2. Quantum algorithms for period finding, which uses the quantum fourier transform (QFT), which we will go over later.
3. Provide a quantum algorithm for the QFT.

16.3 Number Theory

Today, we will go over basic number theory that everyone should know.

Euclid's algorithm was also called the pulverizer in Indian antiquity.

Example 16.1 (Euclid's Algorithm; or, The Pulverizer)
Euclid's algorithm finds the GCD, or greatest common divisor, of two integers $y, z \in \mathbb{Z}$. The idea is that $\gcd(y, z) = \gcd(y, z \mod y)$. For example, $\gcd(24, 33) = \gcd(24, 9) = \gcd(6, 9) = \gcd(6, 3) = 3$.

The runtime of Euclid's algorithm is efficient as each step cuts down the size of the numbers exponentially.

Choose a random $a \in \mathbb{Z}$ such that $1 < a < N$. Then, check whether $\gcd(a, N) = 1$. If a and N are not coprime, then we have found a nontrivial divisor of a.

Example 16.2 (Hard Case)
A hard case of Shor's algorithm would be when $N = pq$ for large primes p and q.

Definition 16.3
The order of a is $\min r > 0$ such that $a^r \mod N = 1$.

Such an r exists when a and N are coprime, since we can write down $a, a^2, a^3, \cdots$ all modulo N. Since there are only finitely many values this infinite sequence can take on, in $\{0, 1, \cdots, N-1\}$, by the pigeonhole principle there must exist $x < y$ such that $a^x = a^y \mod N$, so $a^{y-x} = 1 \mod N$ when a and N are coprime by the Chinese Remainder Theorem.

We can define $f(x) = a^x \mod N$. We can compute $f(x)$ efficiently. In fact r is the period of $f(x)$ since $a^{r+x} = a^r a^x = a^x \mod N$.

Now let's convert factorization to period finding. Suppose we know $r = \mathrm{ord}(a)$. Then $a^r = 1 \mod N$ and $a^r - 1 = mN$ for some $m \in \mathbb{Z}$. Suppose that r is even. Then $a^{r/2} + 1 \neq 0 \mod N$, since $(a^{r/2}+1)(a^{r/2}-1) = mN$, so $\gcd(a^{r/2} - 1, N)$ yields a nontrivial factor.

Example 16.4

For $N = 33, a = 2$, the order of 2 mod 33 is $r = 10$. Happily, r is even, and $2^{r/2} + 1 = 33$. Bad luck! This means that 33 divides $(2^5 - 1)(2^5 + 1) = 31 \cdot 33$.

Let's try it again with $a = 5$. Then $r = 10$ and $33 \big| 22 \cdot 24$.

Theorem 16.5 (Chinese Remainder Theorem)

If $N = pq$ for p, q distinct, then $x \in \{0, \cdots, N-1\}$ can be uniquely determined by $x \mod p$ and $x \mod q$.

17 Shor's Algorithm, Part 2

Last time, we talked about how to factor, given order-finding/period-finding. This time, we will talk about how to do order-finding, given the quantum fourier transform, as well as some modular arithmetic.

17.1 Quantum Fourier Transform

We want to factor a number N, using the following steps.

- Choose a random a between 1 and N.
- Then, we choose n such that $N^2 < 2^n \leq 2N^2$.
- Prepare a superposition of bits from 0 to $2^n - 1$,
$$\frac{1}{\sqrt{2}^n} \sum_{x=0}^{2^n-1} |x\rangle \otimes |0\rangle .$$
- Now, we want to compute $f(x) = a^x \bmod N$ on each of these bits in the register, which yields
$$\frac{1}{\sqrt{2}^n} \sum_{x=0}^{2^n-1} |x\rangle \otimes |a^x \bmod N\rangle .$$
Practically, we can use repeated squaring to compute powers, which boils down to basic arithmetic operations, which generates garbage bits which we then decompute and get rid of.
- The next step is a little strange, just like how it was strange in Simon's algorithm: throw away the computed register $|a^x \bmod N\rangle$ by measuring it. After measuring, the superposition over all x will collapse only to the possible values of x that could have led to $a^x \bmod N$. For example, if $r = \mathrm{ord}(a)$, then if $a^{x_0} = 17 \bmod N$ then $a^{x_0+rk} = 17 \bmod N$, so for $0 \leq x_0 \leq r$, we will have $m \approx \frac{2^n}{r}$, where $\frac{2^n}{r}$ may not be an integer, which is why we take n sufficiently large. This becomes
$$\frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + kr\rangle .$$
Recall in Simon's algorithm it was $\frac{|x\rangle + |x \oplus c\rangle}{\sqrt{2}}$. This yields a series of spikes with some periodicity, which we want to find.
- We have the QFT operator
$$U_{QFT} = F = \frac{1}{\sqrt{2}^n} \sum_{x,y=0}^{2^n-1} e^{2\pi i xy/2^n} |x\rangle \langle y| .$$
Here, with $d = 2^n$ this is
$$\frac{1}{\sqrt{d}} \sum_{x,y=0}^{d-1} \omega^{xy} |x\rangle \langle y| ,$$
where $\omega = e^{2\pi i/d.}$ This is
$$\frac{1}{\sqrt{d}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{d-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{d-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{d-1} & \omega^{d-2} & \cdots & \omega \end{pmatrix},$$
where the real part of each row can be visualized as a sine wave with increasing frequency, due to the characterization of complex numbers $\omega^{ix} = \cos(x) + i \sin(x)$. The real part of the column can similarly be visualized as a sine wave with increasing frequency. This is the discrete quantum Fourier transform.
- Next, apply the discrete QFT operator to our superposition of states, which yields $U_{QFT} \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + kr\rangle$, and measure. This will reveal some information about r.

Claim 17.1. *The outcome $y \approx \frac{2^n}{r} j$ is likely.*

Proof. Applying the QFT gives

$$U_{QFT}\frac{1}{\sqrt{m}}\sum_{k=0}^{m-1}|x_0+kr\rangle = \frac{1}{\sqrt{m2^n}}\sum_{k=0}^{m-1}\sum_{y=0}^{2^n-1}\omega^{yx_0}\omega ykr\,|y\rangle\,.$$

Measuring the probability of y yields

$$P(r)=\frac{1}{m2^n}\left|\sum_{k=0}^{m-1}\omega^{ykr}\right|^2.$$

First, note that the x_0-dependence goes away, and does not affect the interference pattern coming from the sum over k. We have

$$U_{QFT}^\dagger U_{QFT}=\frac{1}{d}\sum_{x,y}\omega^{-xy}\,|y\rangle\,\langle x|\sum_{z,w}\omega^{zw}\,|z\rangle\,\langle w|\,.$$

Here, $\langle x|z\rangle=\delta_{xz}$. This yields

$$\frac{1}{d}\sum_{xyw}\omega^{x(w-y)}\,|y\rangle\,\langle w|\,.$$

Next, sum over x. If $w=y$, then $\sum_x\omega^{x(w-y)}=d=2^n$. If $w\neq y$, then $\sum_x\omega^{x(w-y)}=0$, using the formula for a geometric series. Thus, this becomes

$$\frac{1}{d}\sum_{yw}d\delta_{yw}\,|y\rangle\,\langle w|=I_d.$$

So in fact, U_{QFT} is a unitary operator. Intuitively, phases that oscillate rapidly will "cancel out," which is destructive interference, whereas phases that stay the same will add up.

Intuitively, let $y\approx\frac{2^n}{r}j$. Then, $\omega^{ykr}\approx\omega^{2^njkr/r}\approx 1^{jk}=1$.

In more detail, we have

$$\left|\sum_k\omega^{yrk}\right|^2=\left|\frac{1-\omega^{yrm}}{1-\omega^{yr}}\right|^2=\frac{\sin^2(yrm(2\pi)/(2\cdot 2^n))}{\sin^2(yr(2\pi)/(2\cdot 2^n)}.$$

Suppose $y=\frac{2^nj}{r}+\delta$ where $|\delta|<1/2$. Then,

$$Pr(y)=\frac{1}{m2^n}\frac{\sin^2(\pi mr\delta/2^n)}{\sin^2(\pi mr\delta/2^nm}.$$

Here, recall that $mr\approx 2^n$, so this is approximately

$$\approx\frac{1}{m2^n}\frac{\sin^2(\pi\delta)}{\sin^2(\pi\delta/m)}.$$

Using linear approximation for sin near 0, $\alpha/(\pi/2)\leq\sin(\alpha)\leq\alpha$ since m is exponentially large and $\pi\delta$ is constant. So this is

$$\geq\left(\frac{\pi}{2}\right)^2\frac{m}{2^n}.$$

This is true only for y that are close to $2^nj/r$. Since $1\leq y\leq 2^n$, there are r different values of j, each one proportional to $2^n/r$, so the total probability is about $\frac{4}{\pi^2}\frac{rm}{2^n}\approx\frac{4}{\pi^2}$. These are the ones where the phases almost line up, and each one individually has small probability, but if you add them all up, there is a constant chance of hitting one of them. This gives enough information to figure out r. □

Here we basically try to figure out the phase r by adding up different phases and if the phase we have is similar to the phase r, they will constructively interfere.

18 Shor's Algorithm, Part 3

18.1 Continued Fractions

Last time, for $a^x \bmod N$, we computed $r = \operatorname{ord}(a)$ and used this to get a factor with $\gcd(a^{r/2} \pm 1, N)$. We found that $y = \frac{2^n j}{r} + \delta$ with high probability for some $j \in \mathbb{Z}, |\delta| \leq 1/2$. Then, $\frac{y}{2^n} = \frac{j}{r} + \frac{\delta}{2^n}$; note that we chose $2^n \approx N^2$, and that $\frac{\delta}{2^n}$ is much less than $1/2^n$. Also, we know that $0 < r < N$.

How can we extract j/r given $y/2^n$? There is a classical algorithm called continued fractions which does this.

Example 18.1
For $N = 33, a = 5, r = \operatorname{ord}(5) = 10$, we have $33 | 22 \cdot 24$. For $n = 11, 2^n = 2048$. Suppose we get $y = 615$. How can we find that the order is 10? We have $\frac{y}{2^n} = \frac{615}{2048} \approx \frac{p}{q}$ for $q < N = 33$.

In the continued fractions algorithm, we see that $\frac{2048}{615} = 3 + \frac{203}{615}$, so

$$\frac{615}{2048} = \frac{1}{3 + \frac{203}{615}} = \frac{1}{3 + \frac{1}{3+\frac{6}{203}}} = \frac{1}{3 + \frac{1}{3+\frac{1}{33+\frac{5}{6}}}} = \frac{1}{3 + \frac{1}{3+\frac{1}{33+\frac{1}{1+\frac{1}{5}}}}}.$$

If this number were irrational, we would keep going, but for rational numbers it terminates. Zeroing out each remainder provides a fraction approximation with smaller and smaller error.

This series of rational approximations, in this example, are $\frac{1}{3}, \frac{1}{3+1/3} = \frac{3}{10}, \frac{100}{333}$, and so on, which approach $\frac{615}{2048}$. Eventually, the error term will be small enough to be $\delta/2^n$. It turns out that in this example, $\frac{615}{2048} \approx \frac{3}{10}$ is the desired approximation.

Theorem 18.2
If $|\alpha - p/q| \leq \frac{1}{2q^2}$, then p/q appears in the continued fraction series for α.

We chose n sufficiently large to satisfy the conditions of this theorem.

18.2 Quantum Fourier Transform

Let F_n be the QFT on n qubits. Then $F_n = \frac{1}{\sqrt{2^n}} \sum_{x,y=0}^{2^n-1} \omega_n^{xy} \left|y\right\rangle \left\langle x\right|$, where $\omega_n = e^{2\pi i/2^n}$. For F_1, $\omega_1 = -1$ and

$$F_1 = \frac{1}{\sqrt{2}} \sum_{x,y\in\{0,1\}} (-1)^{xy} \left|x\right\rangle \left\langle y\right| = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = H.$$

For $F_2, \omega_2 = i$, so

$$F_2 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix},$$

where each row and column progresses by a different phase.

Example 18.3
What is the circuit for F_2?

To see the circuit for F_2, let $x = x_0 + 2x_1 + \cdots + 2^{n-1}x_{n-1}$ and $y = y_0 + 2y_1 + \cdots + 2^{n-1}y_{n-1}$. Then $\omega_n^{xy} = \omega_n^{xy \bmod 2^n}$, since $\omega_n^{2^n} = 1$. In base 2, $xy \mod 2^n = x_0y_0 + 2(x_1y_0 + x_0y_1) + \cdots + 2^{n-1}(x_{n-1}y_0 + \cdots + x_0y_{n-1})$. We can rewrite

$$x = \overline{x} + 2^{n-1}x_{n-1}, y = y_0 + 2\overline{y},$$

where $\overline{x} = x_0 + 2x_1 + \cdots + 2^{n-2}x_{n-2}$ and $\overline{y} = y_1 + 2y_2 + \cdots + 2^{n-2}y_{n-1}$.

Now,

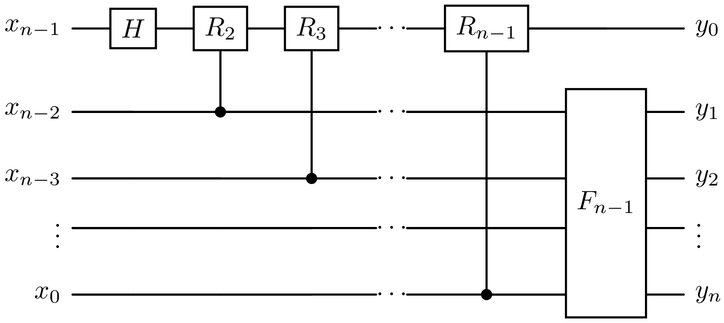
$$\left\langle y\middle|F_n\middle|x\right\rangle = \frac{1}{\sqrt{2}^n}\omega_n^{xy} = \frac{1}{\sqrt{2}^n} \exp\left(\frac{2\pi i}{2^n}\left(2^{n-1}x_{n-1}y_0 + 2^{n-1}x_{n-1}2\overline{y} + \overline{x}y_0 + 2\overline{xy}\right)\right).$$

Since there is a factor of $2^n, 2^{n-1}x_{n-1}2\overline{y} = 0 \mod 2^n$. This becomes

$$\frac{(-1)^{x_{n-1}y_0}}{\sqrt{2}}\omega_n^{\overline{x}y_0}\frac{\omega_{n-1}^{\overline{xy}}}{\sqrt{2^{n-1}}}.$$

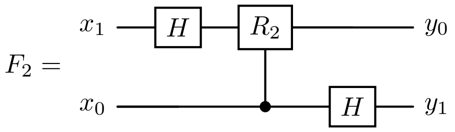
The first term $\frac{(-1)^{x_{n-1}y_0}}{\sqrt{2}}$ looks like a Hadamard, $\omega_n^{\overline{x}y_0}$ looks like a phase shift, and the last term $\frac{\omega_{n-1}^{\overline{xy}}}{\sqrt{2^{n-1}}} = \langle\overline{y}|F_{n-1}|\overline{x}\rangle$, which is a smaller Fourier transform. This then suggests a recursive algorithm.

In particular, for R_i the controlled $\begin{pmatrix}1 & 0\\ 0 & \omega_n\end{pmatrix}$ gate,



We want to decompose all of this into 2-qubit gates. Here, we write $\omega_n^{\overline{x}y_0} = \omega_n^{y_0x_0+2y_0x_1+\cdots} = \omega_n^{y_0x_0}\omega_{n-1}^{y_0x_1}\cdots$. This is a controlled gate C-$\begin{pmatrix}1 & 0\\ 0 & \omega_n\end{pmatrix}$.

In particular,



This is a quantum Fourier transform. Classically, there is a fast Fourier transform. Algorithmically, they live in different worlds, but there are ways to apply ideas from the FFT to the QFT.

19 Quantum Fourier Transform for Phase Estimation

Today, we will talk about a different application of the QFT, which is phase estimation. This will be a different "class" of problem than the one that Shor's algorithm tries to solve.

19.1 Phase Estimation Problem

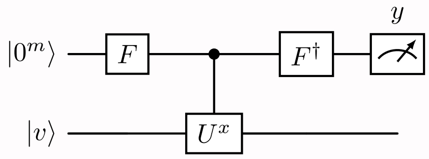
Given an n-qubit unitary U and an eigenstate $|v\rangle$ such that $U|v\rangle = e^{i\theta}|v\rangle$, the **phase estimation problem** is to find θ. For this algorithm, we will use:

- one copy of $|v\rangle$
- the ability to quickly perform $C\text{-}U^{2^k}$, given an input k — writing any number x in binary, we thus have the ability to quickly perform $C\text{-}U^{x}$*

In Shor's algorithm, the inputs and outputs are classical. In constrast, for phase esimation, the inputs U and $|v\rangle$ are presented as a quantum operation and quantum state rather than classical bitstrings. This puts phase estimation in a different category from Shor's algorithm. At the end of the day, for a full algorithm, we will always want a classical input and output, like in Shor's algorithm. In contrast, an algorithm like phase estimation can be thought of as a subroutine or a piece of a potentially larger algorithm. Moreover, there is no classical algorithm to compare phase estimation to, so there is no way to define a "quantum speedup."

Algorithm 19.1

Phase estimation performs the following circuit, where the accuracy depends on the chosen number m. Here, U is an n-qubit unitary, $|v\rangle$ is an eigenstate, and F is the QFT. Also, where $x = F|0^m\rangle$, the CU^x gate performs U^x on the second register:



In phase estimation, we choose a number m, which determines the accuracy of the result, and apply a phase estimation circuit to $|0^m\rangle|v\rangle$, then measure. We can analyze each component of the circuit.

- First, we apply F to the first register of $|0^m\rangle|v\rangle$. Given an input $|0^m\rangle$, applying the QFT F is equivalent to a Hadamard to each component, so
$$F|0^m\rangle \otimes |v\rangle = \frac{1}{\sqrt{2^m}}\sum_x |x\rangle \otimes |v\rangle .$$
- Next, we apply a controlled U to the second register, which is yields
$$\frac{1}{\sqrt{2^m}}\sum_{x=0}^{2^m-1}|x\rangle \otimes U^x|v\rangle = \frac{1}{\sqrt{2^m}}\sum_x |x\rangle \otimes e^{ix\theta}|v\rangle = \frac{1}{\sqrt{2^m}}\sum_x e^{ix\theta}|x\rangle \otimes |v\rangle ,$$
first using that $|v\rangle$ is an eigenvector, and then using the multilinearity of the tensor product. Sometimes, transferring a phase from the second component to the first component using multilinearity is called "phase kickback."
- Then, we apply $F^\dagger$ to the first register. This yields
$$F^\dagger \frac{1}{\sqrt{2^m}}\sum_x e^{ix\theta}|x\rangle \otimes |v\rangle = \frac{1}{2^m}\sum_y\left(\sum_x \exp\left(ix\theta - \frac{2\pi ixy}{2^m}\right)\right)|y\rangle \otimes |v\rangle .$$
- Lastly, we measure the first register in the standard basis and obtain some result y. The probability of obtaining y is the squared norm of the amplitude on this state. The amplitude or coefficient of $|y\rangle$ is
$$\frac{1}{2^m}\sum_{x=0}^{2^m-1}\exp\left(ix\left(\theta - \frac{2\pi}{2^m}y\right)\right).$$

*This is quite a strong assumption, that we can perform CU^{2^k} quickly and not just CU.

In some sense, this amplitude is a comparison between the true phase θ and $\frac{2\pi y}{2^m}$[†], which can be interpreted as measuring how good $\frac{2\pi y}{2^m}$ is as an "estimate" for the true phase. Intuitively, when the estimate is perfect and $\theta = \frac{2\pi}{2^m}y$, the sum of exponentials will compound, and the measurement will always yield y with a probability of 1. In contrast, for poor estimates $\frac{2\pi y}{2^m}$ far from θ, the sum of exponentials will cancel out, and the measurement will yield y with very low probability.

To precisely calculate the probability of measuring y, we can use the geometric series formula. The probability is

$$\frac{1}{2^m}\sum_{x=0}^{2^m-1}\exp\left(ix\left(\theta-\frac{2\pi}{2^m}y\right)\right)=2^{-m}\frac{1-\exp\left(i2^m\left(\theta-\frac{2\pi}{2^m}y\right)\right)}{1-\exp\left(i\left(\theta-\frac{2\pi}{2^m}y\right)\right)}.$$

Using the identity $|1-e^{i\alpha}|^2=\sin^2(\alpha/2)$, the probability, which is the squared norm of the amplitude, is

$$\Pr(y)=4^{-m}\frac{\sin^2\left(2^m\left(\frac{\theta}{2}-\frac{\pi}{2^m}y\right)\right)}{\sin^2\left(\frac{\theta}{2}-\frac{\pi}{2^m}y\right)}.$$

The numerator is upper bounded by 1, and for the sake of time, we use linear pproximation to roughly lower bound the denominator[‡]. Thus, the probability that our measurement results in y is approximately upper bounded by

$$\lessapprox\frac{1}{4^m\left(\frac{\theta}{2}-\frac{\pi y}{2^m}\right)^2}=\frac{\pi^{-2}}{(y-2^m\theta/2\pi)^2}.$$

Thus, the probability of y as the result of measurement essentially depends on the difference between y and θ, after rescaling by $\frac{2^m}{2\pi}$, which is necessary since θ is between 0 and 2π while y goes from 0 to 2^m. The probability that our measurement result y, which we use as an estimate for θ, differs by more than δ from the true angle θ (rescaled) can be calculated as

$$\Pr\left(\left|y-\frac{2^m\theta}{2\pi}\right|>\delta\right)=\sum_{\text{y such that }|y-2^m\theta/(2\pi)|>\delta}\Pr(y)$$

Let $z=|y-2^m\theta/2\pi|$. We can do an upper bound using an integral:

$$\lessapprox 2\int_\delta^\infty dz\frac{1}{\pi^2z^2}=\frac{2/\pi^2}{\delta}.$$

Thus, the probability of being more than 10 away is $\sim 1/10$, the probability of being more than 100 away is $\sim 1/100$, and so on. That is, from our measurement, we learn θ up to accuracy $O(1/2^m)$.

19.1.1 Superposition of Eigenstates

Suppose the input is $\sum_x a_x|v_x\rangle$, where $|v_x\rangle$ is an eigenstate with eigenvalue $e^{i\theta_x}$. It turns out that this argument, run in superposition, will have a phase estimate attached to it. We get $\sum_x a_x|v_x\rangle\sum_y b_{y|x}|y\rangle$, with the property that $|b_{y|x}|^2$ is peaked near $y=2^m\theta_x/2\pi$. For each value of x, we will have a different $b_{y|x}$. Whether or not we can distinguish the eigenvalues depends on the eigenvalue gap. If all the eigenstates have the same eigenvalue, then phase estimation can never tell them apart.

In general, we can identify a specific $|v_x\rangle$ if $|\theta_x-\theta_{x'}|\gg\frac{1}{2^m}$ for all $x'\neq x$.

Example 19.2

Suppose we have $\frac{1}{2}(|v_0\rangle-|v_{\pi/10}\rangle+i|v_{3\pi/10}\rangle+|v_{4\pi/10}\rangle)$. Using phase estimation, get either 0, $\pi/10, 3\pi/10$, or $4\pi/10$, where the quantum result will collapse into one of the $|v_0\rangle, |v_{\pi/10}\rangle, |v_{3\pi/10}\rangle, |v_{4\pi/10}\rangle$ states.

If we had $|(3+10^{-9})\pi/10\rangle$, it becomes $(i|v_{3\pi/10}\rangle+|v_{3.000000\pi/10}\rangle)/\sqrt{2}$.

[†] Here, $\frac{2\pi y}{2^m}$ is essentially y but in different units

[‡] The full analysis is slightly more complicated.

19.2 Discrete Log Problem

Now, we can move on to a different problem, which we will discuss on Wednesday.

- Consider $\mathbb{Z}_N$ to be the integers mod N, $\{0, 1, 2, \cdots, N-1\}$, with $+, -, \times$ defined modulo N.
- Another way to think about this is in terms of equivalence classes, where $N\mathbb{Z} = \{, \cdots, -2N, -N, 0, N, 2N, \cdots\}$ where $\mathbb{Z}_N = \{0 + N\mathbb{Z}, 1 + N\mathbb{Z}, \cdots, (N-1) + N\mathbb{Z}\}$, which is a set of sets, where we can add, subtract, and multiply the sets as usual modulo N. For example, $(3 + 6\mathbb{Z}) + (4 + 6\mathbb{Z}) = 1 + 6\mathbb{Z}$, where $N = 6$.

In division, we have $\gcd(a, N) = ax + Ny$ for some $x, y \in \mathbb{Z}$, and when a and N are coprime, we can write $x = a^{-1} \bmod N$. There is always a multiplicative inverse when N is prime.

20 Discrete Log

20.1 Review

Last time, we talked about phase estimation, which is a very important subroutine. One algorithm that uses phase estimation is discrete log.

20.2 Basic Number Theory

Today, we will be working in $\mathbb{Z}_p$. When p is prime, every integer is relatively prime to p so this forms a field, where each element has a multiplicative inverse.

In $\mathbb{Z}_p$, the following theorem holds.

Theorem 20.1 (Fermat's Little Theorem)
When p is a prime, $x^{p-1} = 1$ for all $x \neq 0$.

A number theory corollary is that the order will always divide $p - 1$. Moreover, we can define primitive roots, where in the following definition we don't consider $g^{p-1} = 1$.

Definition 20.2
An element $g \in \mathbb{Z}_p$ is a **primitive root** if $\{1, g, g^2, \cdots, g^{p-2}\} = \{1, 2, 3, \cdots, p-1\}$. Equivalently, this is true if $\text{ord}(g) = p - 1$.

Example 20.3
For $p = 11$, $g = 2$ is a primitive root because we have $1, 2, 4, 8, 5, 10, 9, 7, 3, 6, 1$, whereas for $g = 3$, which is not a primitive root, we have $1, 3, 9, 5, 4, 1, \cdots$.

20.3 Discrete Log

The discrete log problem is analogous to the usual log.

Problem 20.4
Given p prime, and g a prime root, given g^a, find a.

Clasically, the discrete log problem is $O(p)$, or exponential in the number of bits. The discrete log is interesting because it's used for the Diffie-Hellman key exchange, in cryptography.

Example 20.5 (Diffie-Hellman Key Exchange)
Alice and Bob both agree on p and a primitive root g, which are public knowledge. Then, Alice chooses a private key $a \in \{0, \cdots, p-2\}$ and Bob choose a private key $b \in \{0, \cdots, p-2\}$. Then, Alice and Bob publish their private keys, where Alice publishes g^a and Bob publishes g^b. Now, Alice calculates $(g^a)^b = g^{ab}$ and Bob calculates $(g^b)^a = g^{ab}$, where they both now have the same shared key. This shared key can now be used for various private key cryptosystems.

In the quantum algorithm, take $U_g |x\rangle = |gx\rangle$, all modulo p. Note that U_g can be done efficiently using elementary gates. Moreover, we can do $U_g^{2^k} = |g^{2^k} x\rangle$ efficiently by squaring repeatedly on g. Here, $U_g |1\rangle = |g\rangle$, $U_g^2 |1\rangle = |g^2\rangle, \cdots U_g^{p-1} |1\rangle = |1\rangle$. Then, do a phase estimation on U_g. In the basis $|1\rangle, |g\rangle, |g^2\rangle, \cdots$, U_g looks like $\begin{pmatrix} 0 & 0\cdots & 1 & \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix}$. Since $U_g^{p-1} = I$, the eigenvalues satisfy $\lambda^{p-1} = 1$, so the eigenvaluess are ω^y where $\omega = e^{2\pi i/(p-1)}$ and $y \in 0, 1, \cdots, p-2$. Lastly, the eigenstates are $|\psi_y\rangle = \frac{1}{\sqrt{p-1}} \sum_{x=0}^{p-1} \omega^{xy} |x\rangle$. In fact, the Fourier transform will diagonalize this matrix.

- First, perform phase estimation of U_g on state $|1\rangle$. We have

$$|1\rangle = \frac{1}{\sqrt{p-1}} \sum_{y=0}^{p-2} |\psi_y\rangle = \frac{1}{p-1} \sum_{x,y} \omega^{xy} |g^x\rangle = |1\rangle .$$

  Then, phase estimation yields $\frac{2\pi y}{p-1}$ and collapses the state to $|\psi_y\rangle$, where we need m bits of precision with $2^m \gg p$.

- Next, perform phase estimation with U_{g^a} and state $|\psi_y\rangle$. Recall that the input is p, g, and g^a. We get

$$U_{ga} |\psi_y\rangle = \omega^{ay} |\psi_y\rangle .$$

  Then, phase estimation yields $\frac{2\pi ay}{p-1}$, where ay is modulo $p-1$.

From $\frac{2\pi y}{p-1}$ and $\frac{2\pi ay}{p-1}$, we can recover a, by dividing by a.

21 Grover's Algorithm

In the problems we've achieved quantum speedups so far, we've relied heavily on structure, such as periodicity. However, a huge question in the field is about what kinds of problems can have quantum speedups, and whether structure is necessary.

Guiding Question
What is the source of quantum speedups?

Symmetry or structure may be a source of quantum speedups, or it may instead simply be the "low-hanging fruit" of the field.

Today, we will talk about an algorithm with much less structured input, which achieves a quadratic rather than exponential speedup. That is, the quantum runtime is the square root of the classical runtime.

21.1 Circuit-SAT and the Oracle Problem

The input is a classical circuit $C : \{0,1\}^n \to \{0,1\}$, and CIRCUIT-SAT outputs whether there exists $x \in \{0,1\}^n$ such that $C(x) = 1$. Equivalently, CIRCUIT-SAT outputs $\bigvee_{x\in\{0,1\}^n} C(x)$, which is 1 if any of the $C(x) = 1$ and 0 if all of the outputs are zero.

In complexity theory, there are various classes of problems based on the runtime of the best algorithm to solve them.

- A problem is in P if it can be solved in polynomial time on a classical deterministic computer
- A problem is in BPP if it can be solved in polynomial time on a randomized computer
- A problem is in BQP if it can be solved in polynomial time on a quantum computer

Moreover, a problem is in NP, which stands for nondeterministic polynomial time*, if a solution to the problem can be verified in polynomial time. For example, CIRCUIT-SAT is in NP: given some $x \in \{0,1\}^n$, it takes polynomial time to verify that indeed $C(x) = 1$. In general, CIRCUIT-SAT can be solved in $\approx 2^n\cdot$ (circuit size) time.

Although CIRCUIT-SAT cannot be solved quickly, a closely related problem can be, where the circuit C is not presented as a bunch of gates, but rather an oracle providing the output of C for any input x. The **oracle problem** OR, given a function $f : \{0,1\}^n \to \{0,1\}$ as an input, outputs $\bigvee_x f(x)$, which is the same as whether there exists any $x \in \{0,1\}^n$ such that $f(x) = 1$. The oracle version removes any possible structure, since $f(x)$ provides zero information about $f(x')$, as we do not have access to the underlying circuit. To simplify, we can replace $\{0,1\}^n$ with $[N]$ where $N = 2^n$.

Guiding Question
How many queries are needed to determine OR?

In the deterministic case, N queries are required. In the randomized case, $\Theta(N)$ queries are required for constant error.

21.2 Grover's Algorithm

Grover proved an extremely general case where there is no structure required on the input. Unfortunately, this achieves only a quadratic speedup.

Theorem 21.1 (Grover '96)
Quantum computers can compute OR with $O(\sqrt{N})$ queries.

In fact, this is a lower bound; quantum computers cannot achieve better.

*Not "not polynomial" time!

Theorem 21.2 (BBBV '94)
The lower bound for quantum computers are $\Omega(\sqrt{N})$.

Thus, up to some constants, $\sqrt{N}$ is both an upper and lower bound.

There are a lot of variants of this problem.

Definition 21.3
We call x **marked**, or a solution to the problem, if $f(x) = 1$ and **unmarked** if $f(x) = 0$.

A slightly stronger problem than OR would be "Find x such that $f(x) = 1$, or say that no x exists." Obviously, if this problem can be solved, OR can automatically be solved. Moreover, if OR can be solved, this problem can also be solved, using binary search for x. This takes $\sqrt{N} + \sqrt{N/2} + \sqrt{N/4} + \cdots = O(\sqrt{N})$.

Let's start with some ingredients.

Definition 21.4 (Phase Oracle)
The **phase oracle** is

$$O_f = O_f^{phase} = \sum_x (-1)^{f(x)} |x\rangle \langle x| = I - 2P,$$

where $P = \sum_{x \in f^{-1}(1)} |x\rangle \langle x|$.

Example 21.5
If there is only a single w such that $f(w) = 1$, $O_f = I - 2P = I - 2|w\rangle \langle w|$, which is ones on the diagonal except -1 at (w, w).

Let the superposition state be $|s\rangle = \frac{1}{\sqrt{N}} \sum_{x \in [N]} |x\rangle$.

Definition 21.6
The reflection operator is

$$R_s = 2|s\rangle \langle s| - I.$$

These operators can be written in terms of elementary gates. If $N = 2^n$, then $|s\rangle = H^{\otimes n} |0^n\rangle$, and $R_s = H^{\otimes n}(2|0^n\rangle \langle 0^n| - I)H^{\otimes n}$.

Algorithm 21.7
Starting with $|s\rangle$, we apply O_f and then R_s, and then O_f again, applied T times, then measure.

Very roughly, the amplitude starts at $1/\sqrt{N}$, then applying $R_s O_f$ changes the amplitude to roughly $3/\sqrt{N}$, then applying $R_s O_f$ again changes the amplitude to roughly $5/\sqrt{N}$ again, so overall it takes $O(\sqrt{N})$ time to increase the amplitude to $O(1)$.

Let $M = |f^{-1}(1)|$ be the number of marked inputs, $|\alpha\rangle = \frac{1}{\sqrt{M}} \sum_{x \text{marked}} |x\rangle$ and $|\beta\rangle = \frac{1}{\sqrt{N-M}} \sum_{x \text{unmarked}} |x\rangle$, where $\langle \alpha | \beta \rangle = 0$.

Let $p = M/N$ be the fraction of marked elements. We can write

$$|s\rangle = \sqrt{p}\,|\alpha\rangle + \sqrt{1-p}\,|\beta\rangle.$$

Clasically, there are $\Theta(1/p)$ queries required. We have $O_f |\alpha\rangle = -|\alpha\rangle$ and $O_f |\beta\rangle = |\beta\rangle$. In the $|\alpha\rangle, |\beta\rangle$ basis, $O_f = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$. Moreover, $R_s = 2\begin{pmatrix} \sqrt{p} \\ \sqrt{1-p} \end{pmatrix} \begin{pmatrix} \sqrt{p} & \sqrt{1-p} \end{pmatrix} - I = \begin{pmatrix} 2p-1 & 2\sqrt{p(1-p)} \\ 2\sqrt{p(1-p)} & 1-2p \end{pmatrix}$.

Then,

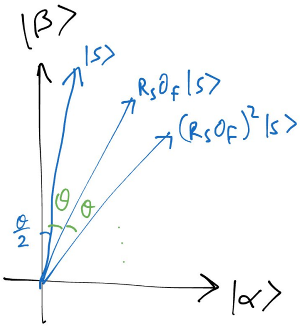
$$R_s O_f = \begin{pmatrix} 1-2p & 2\sqrt{p(1-p)} \\ -2\sqrt{p(1-p)} & 1-2p \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

for some θ, as it takes the form of a rotation matrix. Here, $\theta = \arcsin(2\sqrt{p(1-p)})$.

We have

$$\langle\alpha|s\rangle = \sqrt{p} = \sin\theta/2, \langle\beta|s\rangle = \sqrt{1-p} = \cos\theta/2.$$

If $p \ll 1$, then $\sqrt{p(1-p)} \approx \sqrt{p}$, so $\theta \approx \sin^{-1} 2\sqrt{p} \approx 2\sqrt{p}$, so we want to take T to be approximately $\frac{\pi/2}{\theta}$, rounded.



If T is too large, "overrotating" is bad, which makes it worse. This is different from the classical case.

For an unknown number of marked elements, it suffices to take $T = 1, 2, 4, 8, \cdots, \sqrt{N}$.k

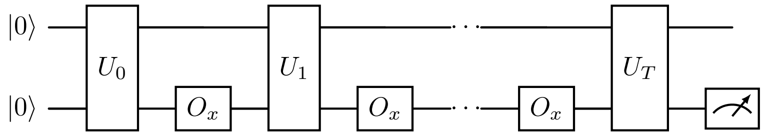
22 Grover's Algorithm with Phase Estimation

An issue last time was that overrotation would make Grover's algorithm fail again, if p is unknown. Grover's algorithm iterates $R_sO_f = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$ in $|\alpha\rangle$, the marked states, and $|\beta\rangle$, the unmarked states. Here, where $p = M/N$, $\theta = \sin^{-1}(2\sqrt{p(1-p)}) \approx 2\sqrt{p}$. One way to estimate p is by using phase estimation on R_sO_f. The eigenvalues are $e^{\pm i\theta}$, since $R_sO_f = e^{i\theta Y} = \cos\theta I + i\sin\theta Y$. Phase estimation will give us an estimate $\hat{\theta} \approx \theta \pm O(2^{-m})$, using m bits of accuracy. This takes 2^m times to yield m bits of precision, so the cost is $O(2^m)$.

Here, $p = 0$ or M/N, and $\theta = 0$ or $2/\sqrt{N/M}$ if $T \sim \sqrt{N/M} \sim \frac{1}{\sqrt{p}}$, where the estimate is $\hat{\theta} \approx \theta \pm O(1/T)$, taking cost $O(T)$. This is called the approximate counting algorithm.

Then$\hat{p} = \sin^2(\theta/2) \approx p$, and $d\hat{p}/d\hat{\theta} = 2\sin(\hat{\theta}/2)\cos(\hat{\theta}/2) = \sin\hat{\theta} \approx 2\sqrt{\hat{p}} \approx \sqrt{p}$. Then $\hat{p} \approx p \pm O(2\sqrt{p}/T)$, in this case $1/N \pm O(1/\sqrt{N}T)$. Assume that $M = N/2$ or $N/2+1$ So we need $T \approx N$.

The most general possible algorithm is, for $M = 1$,, where $O_x|y\rangle = (-1)^{\delta_{x,y}}|y\rangle$,



Then $|\psi_t^x\rangle = U_tO_xU_{t-1}O_x\cdots O_xU_0|0\rangle$, where $|0\rangle$ means many zeroes, and $|\psi_t\rangle = U_tU_{t-1}\cdots U_0|0\rangle$. The progress measure is $D_t = \sum_{x\in[N]} ||\,|\psi_t^x\rangle - |\psi_t\rangle\,||^2$.

1. $D_t \leq 4t^2$ for $t = 0, 1, \cdots, T$
2. Finding x perfectly: $D_T \geq 2N - 2\sqrt{N}$
3. Finding x with probability 1/2: $D_t \geq \Omega(N)$

We have

$$D_{t+1} = \sum_x ||U_{t+1}O_x|\psi_t^x\rangle - U_{t+1}|\psi_t\rangle||^2 \leq \sum_x (||U_{t+1}O_x|\psi_t^x\rangle|| + ||U_{t+1}|\psi_t\rangle||)^2$$

$$= \sum_x ||\,|\psi_t^x\rangle - |\psi_t\rangle\,||^2 + \sum_x ||(O_x - I)|\psi_t\rangle||^2 + 2\sum_x ||\,|\psi_t^x\rangle - |\psi_t\rangle\,|| \cdot ||(O_x - I)|\psi_t\rangle||.$$

Here, $O_x = I - 2|x\rangle\langle x|$ and $(O_x - I)|\psi\rangle = -2|x\rangle\langle x|\psi\rangle$. So then $\sum_x ||(O_x - I)|\psi_t\rangle||^2 = 4\sum_x |\langle x|\psi_t\rangle|^2 = 4$.

Using Cauchy-Schwarz on the last term yields $\sum_x ||\,|\psi_t^x\rangle - |\psi_t\rangle\,|| \cdot ||(O_x - I)|\psi_t\rangle|| \leq \sqrt{\sum_x ||\,|\psi_t^x\rangle - |\psi_t\rangle\,||^2 \sum_x ||(O_x - I)|\psi_t\rangle||^2}$ $\sqrt{(4D_t)}$.

$$\leq D_t + 4 + 2\sqrt{4D_t}.$$

Using induction, this shows that $D_t \leq 4t^2$. THis means that the progress measure doesn't increase too quickly.

23 BBBV LB for OR

Last time, we showed that $D_t = \sum_{x\in[N]} || \, |\psi_t^x\rangle - |\psi_t\rangle \, ||^2 \leq 4t^2$.

The setup is slightly different than in Grover's algorithm:

- Run a unitary circuit $U_T O_X U_{T-1} \cdots U_0 \, |0\rangle$, without knowing x. Then, given a random $x \in [N]$, measure and output whether x is marked, or nothing is marked.

However, we don't get another run of the random oracle to check whether x is marked: we can only use the state we already have. If the state we have provides enough information, we can determine whether x is marked or not, even without running the oracle on x.

Now, we can analyze how this last step works. Suppose we measure in the $|v_1\rangle, \cdots |v_d\rangle$ basis, where d is the total dimension of the quantum computer. Then, let $p_i = |\langle v_i|\psi\rangle|^2$.

Definition 23.1
The **variational distance** between probability distributions p and q is

$$\frac{1}{2}\sum_i |p_i - q_i| = \frac{1}{2}||p-q||_1.$$

Moreover, the usual norm can be denoted as

$$\sqrt{\sum_i |p_i - q_i|^2} = ||p-q||_2.$$

Then, we can consider the difference between two probability measures evaluated on events: $\max_E |p(E) - q(E)| \leq \frac{1}{2}||p-q||_1$.

Example 23.2
Let p be the probability measure $(0.2, 0.2, 0.1, 0, 0.5)$ and $q = (0.3, 0, 0, 0.2, 0.5)$. Then, take $E = \{2, 3\}$. Here, $||p-q||_1 = \frac{1}{2}(0.1 + 0.2 + 0.1 + 0.2 + 0) = 0.3$ and $|p(E) - q(E)| = 0.3$. Considering $E = \{1, 4\}$ yields $|p(E) - q(E)| = 0.5 - 0.2 = 0.3$.

Analogously, consider states $|\alpha\rangle, |\beta\rangle$ measured in an orthonormal basis $|v_1\rangle, \cdots, |v_d\rangle$, where $a_i = |\langle v_i|\alpha\rangle|^2$ and $b_i = |\langle v_i|\beta\rangle|^2$. Then,

$$\begin{aligned}
\frac{1}{2}||a-b||_1 &\leq 2|| \, |\alpha\rangle - |\beta\rangle \, ||_2 \\
&= \sum_i (|\langle v_i|\alpha\rangle| + |\langle v_i|\beta\rangle|)(|\langle v_i|\alpha\rangle| + |\langle v_i|\beta\rangle|) \\
&\leq \sqrt{\sum_i (|\langle v_i|\alpha\rangle| + |\langle v_i|\beta\rangle|)^2 \sum_j (|\langle v_j|\alpha\rangle| + |\langle v_j|\beta\rangle|)^2},
\end{aligned}$$

using Cauchy-Schwarz. Using the triangle inequality, $\sum_i (|\langle v_i|\alpha\rangle| + |\langle v_i|\beta\rangle|)^2 \leq \sum_i |\langle v_i| (|\alpha\rangle - |\beta\rangle)|^2$. Using the fact that the v_i form an orthonormal basis, this is $= || \, |\alpha\rangle - |\beta\rangle \, ||^2$.

Thus, we get

$$\begin{aligned}
&= \sum_j (|\langle v_j|\alpha\rangle| + |\langle v_j|\beta\rangle|)^2 \\
&= \sum_j |\langle v_j|\alpha\rangle|^2 + |\langle v_j|\beta\rangle|^2 + 2|\langle v_j|\alpha\rangle \langle v_j|\beta\rangle| \\
&\leq 2 + 2 \\
&= 4.
\end{aligned}$$

Suppose the success probability is at least 1/2 for at least 1/2 of the x's:

$$D_T \geq \frac{N}{2}\frac{1}{2}\frac{1}{4^2} = cN,$$

and

$$D_T \leq 4T^2.$$

So $T \geq \Omega(\sqrt{N})$.

A related problem: We have $f : [N] \to S$. Is f one-to-one or does there exist $x \neq y$ such that $f(x) = f(y)$? Element distinctness is $\Theta(N^{2/3})$.

The collision problem is whether f is 1-1 or 2-1, which is $\Theta(N^{1/3})$.

Simon's problem is $O(\log N)$, since N is the size of the input, $N = 2^n$. Since there is much more structure, an exponential speedup is achieved.

Another problem with no speedup is parity, for $f : [N] \to \mathbb{Z}_2$, which is $\sum_{x\in[N]} f(x)$. This is $\Theta(N)$.

23.1 Quantum Dynamics and Simulation

In terms of quantum mechanics, some quantities in the world are like qubits, such as a spin-1/2 particle, in $\mathbb{C}^2$. There are also spin 1 particles, which are described by a 3-level system.

Given a particle on an interval $[0, L]$, there is a "position" and a "momentum." Restricting the particle to a lattice with spacing $a = L/N$, the position is in $\mathbb{C}^N$. A superposition of different positions provides the momentum by taking the QFT. Given a superposition of position $\psi_0 \left|0\right\rangle + \cdots + \psi_{N-1} \left|N-1\right\rangle$, the momentum is $U_{QFT}\left(\psi_0 \left|0\right\rangle + \cdots + \psi_{N-1} \left|N-1\right\rangle\right) = \tilde{\psi_0} \left|0\right\rangle + \cdots + \tilde{\psi}_{N-1} \left|N-1\right\rangle.$

A state with momentum p can be represented as $\frac{1}{\sqrt{N}}(1, \omega^p, \omega^{2p}, \cdots, \omega^{p(N-1)})$.

For a particle in 3D, we have $\mathbb{C}^n \otimes \mathbb{C}^n \otimes \mathbb{C}^N$.

24 Quantum Dynamics & Hamiltonian Simulation

Today, he handed out leftover Halloween candy.

24.1 Introduction and Setup

Let's first do a quick review of how everything in the universe works. One of the most important applications of quantum computing is simulating quantum dynamics in the physical world. Natural systems evolve similarly to how quantum circuits operate, where various operations are applied to a state that changes over time. One important difference is that in real life, time is continuous, while for quantum circuits, time is discrete. Another important difference is that in real life, there is a spatial dependence of operations. For example, forces depend on distance or location.

For this setup, we make a few assumptions as axioms, in order to obtain a bit more structure. In discrete time, linear operators U act to take $|\psi\rangle \to U|\psi\rangle$. In continuous time, we can also assume linearity, so $\frac{d}{dt}|\psi(t)\rangle = f(|\psi(t)\rangle) = A|\psi(t)\rangle$, where A is a linear operator.* Moving the derivative into the bra,

$$\frac{d}{dt}\langle\psi| = \langle\psi| A^\dagger.$$

Thus, as the norm should not change,

$$0 = \frac{d}{dt}\langle\psi|\psi\rangle = \langle\psi|A|\psi\rangle + \langle\psi|A^\dagger|\psi\rangle = \langle\psi|(A + A^\dagger)|\psi\rangle.$$

Therefore, since this equation holds for all values of ψ, $A + A^\dagger = 0$.

Definition 24.1
If $A^\dagger = -A$, then A is **skew-Hermitian**.

Thus, operators on continuous quantum states A are skew-Hermitian.

Proposition 24.2
Skew-Hermitian matrices A can be written as $A = -iH$ where $H = H^\dagger$, called the **Hamiltonian**, is Hermitian.

Writing $A = -iH$, the equation for the continuous time evolution of a quantum state,

$$\frac{d}{dt}|\psi(t)\rangle = -iH|\psi(t)\rangle,$$

which is the very important **Schrodinger equation**.

Proposition 24.3
The Hamiltonian measures energy, in that an eigenstate $|\varphi_E\rangle$ such that $H|\varphi_E\rangle = E|\varphi_E\rangle$ has energy E. Moreover, $\langle\psi|H|\psi\rangle$ is the average energy of ψ, decomposed into eigenstates.

Looking at units of $\frac{d}{dt}|\psi(t)\rangle = -iH|\psi(t)\rangle$, on the left is frequency, and on the right is energy. This is similar to, using the speed of light, $E = mc^2$, which means that energy and mass are "equivalent." Using the Schrodinger equation predicts that energy and frequency are also "equivalent" or have a "fundamental relationship." In fact, this fundamental relationship is given by Planck's constant, which is a very small number, which reflects the fact that quantum phenomena occur at very small scales.

Definition 24.4
Planck's constant is $\hbar \approx 1.055 \cdot 10^{-35} J \cdot s = \frac{J}{Hz}$.

Thus, Schrodinger's equation is sometimes written as $\frac{d}{dt}|\psi\rangle = \frac{-iH}{\hbar}|\psi\rangle$, but in this class we will use units such that $\hbar = 1$. We can solve the differential equation.

*From now on, we may simply write $|\psi\rangle$ rather than $|\psi(t)\rangle$.

Proposition 24.5
The solution to the Schrodinger equation given an initial condition $|\psi(0)\rangle$ is $|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$.[a]

[a] Here we take the matrix exponential.

Corollary 24.6
In an eigenbasis, given an initial condition $|\psi(0)\rangle = \sum_E c_E |\varphi_E\rangle$, the solution at time t is

$$|\psi(t)\rangle = \sum_E c_E e^{-iEt}|\varphi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$

Theorem 24.7 (Relationship between Hermitian and Unitary Matrices)
If $H = H^\dagger$, then e^{-iHt} is unitary.

If $H = I$, then the solutions will be $|\psi(t)\rangle = e^{-it}|\psi(0)\rangle$, which is simply an overall phase. Overall phases don't matter, only phase differences, so identity terms in H can be ignored.

24.2 Examples of Quantum Systems

This is all a bit abstract, so we can look at some examples.

Example 24.8 (Spin-1/2 Particle)
Consider a single spin-1/2 particle. The dimension is $d = 2$. Any 2×2 Hermitian matrix can be written as $H = \vec{v}\cdot\vec{\sigma} = v_1\sigma_1 + v_2\sigma_2 + v_3\sigma_3$, a linear combination of the Pauli matrices. Then, $e^{-iHt} = \cos(|\vec{v}|t)I - i\sin(|\vec{v}|t)\frac{\vec{v}}{|\vec{v}|}\vec{\cdot}\sigma$.

Consider a chain of spin-1/2 particles.

Example 24.9 (TFIM)
One model for a chain of spin-1/2 particles is the transverse-field Ising model, where all the particles are aligned in one direction. Suppose $H = -\sum_{i=1}^{n} Z_i Z_{i+1}$, where Z_i is a Pauli matrix. Then, the eigenvalues of Z_i are 1 and -1 and the eigenvalues of $Z_i Z_{i+1}$ are 1 for eigenvectors that look like $|0\rangle|0\rangle$ and $|1\rangle|1\rangle$ and -1 for eigenvectors $|0\rangle|1\rangle$ and $|1\rangle|0\rangle$. The transverse-field model adds a field in the X direction with strength Γ, so that

$$H = -\sum_{i=1}^{n} Z_i Z_{i+1} - \Gamma\sum_{i=1}^{n} X_i.$$

Example 24.10 (Heisenberg Model)
Another model is $H = \pm\sum_{i=1}^{n-1} SWAP_{i,i+1}$. In the ferromagnetic model, $H = -\sum_{i=1}^{n-1} SWAP_{i,i+1}$ and alignment of the spins is energetically favorable. The antiferromagnetic model has the opposite sign, so that anti-alignment is favored.

Example 24.11 (Particle in 1-D)
Consider a chain of N particles spaced by a in one dimension, as in last class, where we take a lattice on a line with spacing a and length $L = Na$. We attach some "energy" to each point in space, such as for a gravitational field. The potential energy at a point x is then $\sum_{x=0}^{N-1} V(x) \left|x\right\rangle \left\langle x\right|$. Also, the kinetic energy term is $\frac{1}{2ma^2} \sum_{x=0}^{N-1} \left|x\right\rangle \left\langle x+1\right| + \left|x\right\rangle \left\langle x-1\right|$, where the $\left|x\right\rangle \left\langle x\right|$ terms become the identity so we omit them. Then, the Hamiltonian is

$$H = \sum_{x=0}^{N-1} V(x) \left|x\right\rangle \left\langle x\right| + \frac{1}{2ma^2} \sum_{x=0}^{N-1} \left|x\right\rangle \left\langle x+1\right| + \left|x\right\rangle \left\langle x-1\right|.$$

This couples a diagonal term with an "adjacent" term. This is diagonalized by the Fourier transform, so $F(\sum_x \left|x\right\rangle \left\langle x+1\right| + \left|x\right\rangle \left\langle x-1\right|)F^\dagger = \sum_{z=0}^{N-1} \cos(2\pi z/N) \left|z\right\rangle \left\langle z\right|$, which allows us to compute this on a quantum computer.

On a quantum computer, we can implement e^{-iHt} by $\left|x\right\rangle \mapsto \left|x\right\rangle \left|V(x)\right\rangle \mapsto e^{-iV(x)t} \left|x\right\rangle \left|V(x)\right\rangle \mapsto e^{-iV(x)t} \left|x\right\rangle$.

25 Hamiltonian Simulation

Today, we will talk about Hamilton simulation with applications to ground-state energy estimation and adiabatic algorithm.

Recall from last time the Schrodinger equation

$$\frac{d}{dt}|\psi\rangle = -iH|\psi\rangle$$

with solution

$$|\psi(t)\rangle = \exp(-iHt)|\psi(0)\rangle .$$

Proposition 25.1
If A and B commute, for the matrix exponential, $e^{A+B} = e^A e^B$.

For example, if $A = X_1 + X_2 + \cdots + X_n$, then $e^{-iAt} = e^{-itX_1} \cdots e^{-itX_n}$, since the X_i are each on different systems so they commute.

Definition 25.2
The **commutator** is $[A, B] = AB - BA$.

The commutator is zero if A and B commute. Moreover, the commutator works nicely with the tensor product.

Proposition 25.3
If $[A, C] = 0$ and $[B, D] = 0$, then $[A \otimes B, C \otimes D] = 0$.

Similarly, if $B = Z_1Z_2 + Z_2Z_3 + \cdots + Z_{n-1}Z_n$, then $e^{-iBt} = e^{-itZ_1Z_2} \cdots e^{-itZ_{n-1}Z_n}$.

To determine the solution where the Hamiltonian is a sum of two actions in general, we can expand using the Taylor series.

Theorem 25.4 (Trotter-Suzuki)
If $H = A + B$, then e^{-iHt} and $e^{-iAt}e^{-iBt}$ agree to the first order. In particular, $e^{-iHt} - e^{-iAt}e^{-iBt} = \frac{t^2}{2}[A, B] + O(t^3)$.

Proof. If $H = A + B$, then we can write

$$\begin{aligned} e^{-iAt}e^{-iBt} &= \left(I - iAt - \frac{A^2t^2}{2} + \cdots\right)\left(I - iBt - \frac{B^2t^2}{2} + \cdots\right) \\ &= I - it(A + B) - \frac{t^2}{2}(A^2 + 2AB + B^2) + O(t^3). \end{aligned}$$

However, $e^{-iHt} = I - it(A + B) - \frac{t^2}{2}(A^2 + AB + BA + B^2) + O(t^3)$. □

Definition 25.5
The **operator norm** of an operator A is

$$||A|| = ||A||_{op} = ||A||_\infty = ||A||_{2\to 2} = \max\{||A|\psi\rangle|| : |||\psi\rangle|| = 1.\}$$

Equivalently, if A is diagonalizable, then $||A|| = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\}$.

Given a goal operator U, if $\tilde{U}$ is achieved, then $||(U - \tilde{U})|\psi\rangle|| \leq ||U - \tilde{U}||$. That is, the worst-case error on any input is bounded above by the operator norm.

Proposition 25.6

The operator norm satisfies:

- **Nonnegativity**: $||A|| \geq 0$
- **Triangle inequality**: $||A + B|| \leq ||A|| + ||B||$
- **Unitary invariance**: If U is unitary, then $||UA|| = ||A|| = ||AU||$
- **Hybrid argument**: If U_i are unitary, $||U_1 \cdots U_T - \tilde{U}_1 \cdots \tilde{U}_T|| \leq ||U_1 - \tilde{U}_1|| + \cdots + ||U_T - \tilde{U}_T||$

Example 25.7

Consider the case of the spin chain where $A = X_1 + \cdots + X_n$ and $B = Z_1 Z_2 + \cdots Z_{n-1} Z_n$. Then $||A|| = n$ and $||B|| = n$ as well. Next, $[A, B] = \sum_{i,j} [X_i, Z_j Z_{j+1}]$, where the commutator is 0 if $i \neq j, j+1$. Therefore, $||[A, B]|| = O(n)$. Then the Trotter error, splitting time into r intervals, is $||(e^{-i(A+B)t/r})^r - (e^{-iA(t/r)} e^{-iB(t/r)})^r||$ is $\frac{t^2}{r} ||[A, B]|| + \cdots$. The error is thus $\varepsilon = \frac{t^2}{r} O(n)$ and so the time scaling factor must be $\sim \frac{t^2 r}{\varepsilon}$, for a desired error ε.

To get a tighter error bound, a higher-order Trotter-Suzuki formula can be used. In general, for a pth order approximation, the timescale for an error ε will be roughly $r \sim \frac{t^{1+1/p}}{\varepsilon^{1/p}}$. Then $\min_p \frac{S^p t^{1/p}}{\varepsilon^{1/p}} \sim \exp(O(\sqrt{\log t/\varepsilon}))$.

26 Missing

27 Quantum Error Correction

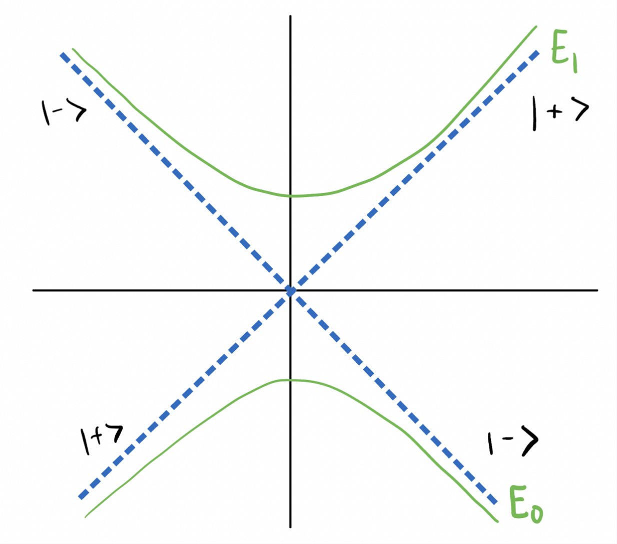
27.1 Review: Adiabatic Theorem

Consider reparametrization $H(s) = H(t/T)$, where we rescale, and making sure that $T \gg \left(\frac{1}{\min_s E_1(s) - E_0(s)}\right)^2$ so that we stay in the ground state.

For the 1-qubit case, the Landau-Zener transition provides a good example.

Example 27.1 (Landau-Zener Transition)
Consider $H(s) = Z + sX$, where $s \in \mathbb{R}$. The eigenvalues are $E_0, E_1 = \pm\sqrt{1+s^2}$.[a]

As $s \to \infty$, $H \to X$, and the eigenvectors are $\left|+\right\rangle, \left|-\right\rangle$, and as $s \to -\infty$, $H \to -X$, and the eigenvectors are $\left|+\right\rangle, \left|-\right\rangle$.



If you move very slowly, you stay in the same energy level but move to a different state, and if you move very quickly, you stay in the same state, which is at a different energy level.

[a] In fact, the eigenvalues of $aI + bX + cY + dZ = a \pm \sqrt{b^2+c^2+d^2}$.

27.2 Adiabatic Optimization

Consider $H_0 = -\sum_{i=1}^n X_i$, with an initial ground state $\left|+\right\rangle^{\otimes n}$. Then, $H_1 = \text{diag}(f) = \sum_{z\in\{0,1\}^n} f(z) \left|z\right\rangle \left\langle z\right|$. The ground state encodes the solution to an NP-complete problem, $\left|z\right\rangle$ for $z = \text{argmin}\, f(z)$. Then, the ground state of $(1-s)H_0 + sH_1 = H(s)$.

We can minimize:

$$\min_{|\psi\rangle} \left\langle\psi|H|\psi\right\rangle = s\sum_z |\psi_z|^2 f(z) + (1-s)\left\langle\psi|H_0|\psi\right\rangle.$$

Here, $nI + H_0 = \sum_{i=1}^n I - X_i = 2\sum_{i=1}^n \left|-\right\rangle\left\langle-\right|$, as $X = \left|+\right\rangle\left\langle+\right| - \left|-\right\rangle\left\langle-\right|$. Then,

$$\frac{1}{2}\left\langle\psi|nI + H_0|\psi\right\rangle = \sum_{i=1}^n \sum_{z\in\{0,1\}^n} |\psi_z - \psi_{z+e_i}|^2,$$

where $e_i = (0,\cdots,0,1,0,\cdots,0)$. That is, $z + e_i$ differs from z by a bit-flip in one position. Here we use $||\left\langle-\right|_i \left|\psi\right\rangle||^2 = \sum_z |\psi_z - \psi_{z+e_i}|^2$.

The classical analogue is "simulated annealing," taking inspiration from cooling metals, where the goal is to minimize over a probability distribution, encouraging it to spread out:

$$\min_P \left(\sum_z p(z) f(z) - \underbrace{T}_{\text{temp}} \underbrace{\sum_z p(z) \log \frac{1}{p(z)}}_{\text{entropy of } p} \right).$$

The entropy is another way of measuring how "spread out" the probability distribution is, compared to a more "local" concept in the quantum case.

Adiabatic optimization and simulated annealing are two heuristics that are used for optimization problems.

27.3 Preparing Ground States

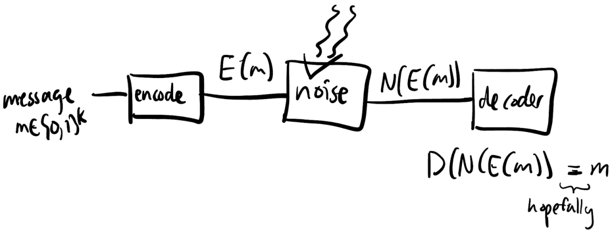
Another way to use this is for ground state estimation, which is NP-hard. One example is that a molecule at room temperature should have electrons generally at ground state. One way of finding ground states is "simulating nature" and asking molecules to be a ground state, and finding their electron states. To prepare ground states, take $H_0 = T$ to be a part of the full Hamiltonian $H_1 = T + V$. It's easy to diagonalize T or V, but not their sum. Then, in the adiabatic framework, we can start at T and gradually increase V and hope that the ground state is still a ground state of $sT + (1 - s)V$.

Consider H on n qubits, so there are $O(n)$ terms. Suppose each term has energy $O(1)$. Then the spectrum goes from $O(n)$ to $-O(n)$, and there are 2^n eigenvalues, so "small gaps" between eigenvalues are expected. The only way to satisfy the adiabatic condition is to have a large gap, which is more likely towards the ends of the spectrum.

27.4 Error Correction

So far, we have assumed that all quantum gates are perfect — however, in reality, there is always some error or noise. Error correction solves many issues: it allows the route to quantum computing to be feasible, given noise, and also allows for "digital" or "discrete" quantum computing.

Classical error correction was already pretty amazing.



Provided a message, we encode the message, noise strikes and we decode the message again, hoping that the message can still be recovered given that there is noise. We do this by adding redundancy.

There are various ways of formalizing this. For example, for randomized errors, recall the binary symmetric channel, where we flip each bit with probability p. Another way of analyzing this is by taking worst-case analysis, assuming that at most np bits are flipped.

Definition 27.2
The message is composed of **logical bits**, and the encoded message is composed of **physical bits**.

One example is the repetition code. One example is the repetition code. One example is the repetition code.

Example 27.3 (Repetition Code)
The encoding takes $0 \mapsto 000, 1 \mapsto 111$, and the decoder is majority vote (given 010, guess that the original bit was 0.) The probability of a logical error, or an error in decoding to the original logical bit of the message, is the probability that 2 or 3 bits are flipped, which is $3p^2 + p^3$ for a binary symmetric channel. Thankfully, this probability of a logical error is $O(p^2)$, which is smaller than p for sufficiently small p.

In *fault-tolerant* computing, we consider cases where encoding and decoding also have some error probability.

28 Classical & Quantum Error Correction

28.1 Repetition Code

Recall the repetition code.

Example 28.1
Encode k bits, with noise from a binary symmetric channel with constant p, say $1/10$. In the ℓ-bit repetition code, $x_1 \cdots x_k \mapsto x_1^\ell \cdots x_k^\ell$, where k logical bits map to $n = k\ell$ physical bits.

Consider the probability of error, the probability that some x_i is decoded incorrectly:

$$\begin{aligned} Pr[\text{decode } x_i \text{ wrong}] &= Pr[\text{at least } \ell/2 \text{ bits are flipped by noise}] \\ &= Pr\left[\text{Bin}(\ell, p) \geq \frac{\ell}{2}\right]^* \end{aligned}$$

It turns out that for the binomial distribution, this probability is approximately exponentially shrinking with ℓ:

$$Pr[\text{decode } x_i \text{ wrong}] \approx e^{-c\ell},$$

where $c > 0$ is some constant depending on p. That is, the error probability decreases exponentially with the block length — that is, the more bits, the more reliable the code.

Unfortunately, this is only the probability of a particular bit is wrong — in general, for k bits, we want all k bits to be correct. Thus, the probability of decoding a particular x_i wrong should be at most $1/k$. That is,

$$Pr[\text{any bit is wrong}] \approx ke^{-c\ell},$$

so we need $\ell \sim \log k$ in order for this probability to be constant. In this case, $n = k\ell \sim k \log k$.

Definition 28.2
The **code-rate** is k/n.

Shannon found that it's possible to both have constant probability of error, and constant rate of transmission.

28.2 General Error-Correcting Codes

First, we ignore efficiency and analyze the existence of codes with certain properties. In general, consider the space $\{0,1\}^n$, and let C be a set of codewords $C \subseteq \{0,1\}^n$. Let $|C| = 2^k$.[†]

Definition 28.3
Let dist be the **Hamming distance**, which is the number of positions where two codewords differ.

For example, $\text{dist}(00100, 11100) = 2$.

Definition 28.4
Let d be the **code distance**, where

$$d = \min_{x,y \in C, x \neq y} \text{dist}(x, y).$$

There are two kinds of errors: a bit is replaced by a ?, where we know that this bit has been corrupted. A harder kind of error is a bit-flip error, where a 1 turns into a 0, or vice versa, which is more difficult because we don't necessarily know where the error occurred.

Proposition 28.5
The code C can correct $d - 1$ erasure errors or $\lfloor \frac{d-1}{2} \rfloor$ bit-flip errors.

[†] We ignore the "map" from messages to codewords, since we want to analyze the error-correcting properties of the code, or set of codewords, itself.

Example 28.6
Consider the $\ell = 5$ repetition code. Let $C = \{00000, 11111\}$. Here, $d = 5$, so given 4 erasure errors, such as in ???0?, it's still clear that the original codeword was 00000. Moreover, given 2 bit-flip errors, such as in 10110, it's still clear that the original codeword was 11111.

Definition 28.7
A **good code family** satisfies $\lim_{n\to\infty} \frac{k}{n} > 0$, and $\lim_{n\to\infty} \frac{d}{n} > 0$.

For the repetition code, worst-case errors are much worse than random or average-case errors.

One of Shannon's insights is that it's much better to encode a string of bits, than each bit one-by-one.

Example 28.8 (Parity Code)
Take $x_1 \cdots x_k \mapsto x_1, \cdots, x_k, x_1 + \cdots x_k \mod 2$, where we add one parity bit. The set of codewords C is $\{x \in \mathbb{F}_2^n : (1, \cdots, 1) \cdot x = 0\}$, which is the set of strings with parity zero. Then, the number of bits we can encode is $k = n - 1$, and $d = 2$.

This is an example of an important family of codes called linear codes.

Definition 28.9
A linear code consists of codewords C that form a k-dimensional subspace of $\mathbb{F}_2^n$ and a linear encoding $E : \mathbb{F}_2^k \to \mathbb{F}_2^k$.

Definition 28.10
An $[n, k, d]$-code is a code with n physical bits, k logical bits, and distance d.

The parity code is an $[n, n-1, 2]$-encoding.

Example 28.11 (Hamming code)
Codewords are 7 bits long (in general, codewords can be $2^s - 1$ bits long.) Denote the original 4-bit message to be (x_3, x_5, x_6, x_7), where we avoid indices that are powers of 2. We map this by taking

$$(x_3, x_5, x_6, x_7) \mapsto (x_3 + x_5 + x_7, x_3 + x_6 + x_7, x_3, x_5 + x_6, +x_7, x_5, x_6, x_7),$$

where we take the binary expansions $3 = 011, 5 = 101, 6 = 110$, and $7 = 111$, and create parity check bits considering the other bits that have a 1 in the appropriate binary position. Flipping one bit makes 2-3 parity bits mismatch, and flipping two bits makes at least one parity check bit mismatch. In fact, this code can thus detect up to 3 errors. The Hamming code is an [7, 4, 3]-encoding.

This is another linear code.

28.3 Linear Codes

Linear codes are as good as any other code that people have found so far. In general, linear codes can be described by a set of linear constraints. Given a message $m \in \mathbb{F}_2^k$, encode $m \to mG$, where G is a matrix in $\mathbb{F}_2^{k\times n}$. We can also define a parity check matrix $H \in \mathbb{F}^{n\times(n-k)}$, since there are $n - k$ parity check bits. We should satisfy $GH = 0$, so $mGH = 0$. That is, for any valid encoding, mGH should be zero, or pass all the parity checks. Given an error e, we get $mG = e$. Then the syndrome is $(mG + e)H = mGH + eH = eH$. If the distance is high enough, and the error is small enough, it is possible to reconstruct what e is.

The problem "find e to minimize the number of 1s in e such that $eH =$ syndrome" is NP-complete. That is, finding the smallest "weight" error, given the syndrome, for a general code, is hard. A code provided by an adversary is likely hard to decode, but as a code designer, don't pick a hard code.

28.4 Quantum Codes

The quantum code $|\psi\rangle \to |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$ is both impossible, due to no-cloning, and difficult to decode. The arguments against quantum error correction include:

- No-cloning
- Measurement collapse
- Continuous errors

29 Quantum Error Correction

Recall the issues with quantum error correction: no-cloning, measurement collapse, and continuous errors. The solution is isometric encoding. For square matrices, unitaries are isometric, but there are rectangular isometric matrices as well that satisfy $E^\dagger E = I$, which increases dimension. If an isometry E is rectangular $m \times n$,then the rank of E is $\min(m, n)$, so $E^\dagger E = I$, but $EE^\dagger \neq I$.

29.1 3-Qubit Bit-Flip Code

Let $E\left|0\right\rangle = \left|000\right\rangle, E\left|1\right\rangle = \left|111\right\rangle$. Then, $E^\dagger E = \left|0\right\rangle\left\langle 0\right| + \left|1\right\rangle\left\langle 1\right| = I_2$, but $EE^\dagger = \left|000\right\rangle\left\langle 000\right| + \left|111\right\rangle\left\langle 111\right| \neq I_8$. Thus E is an isometric encoding.

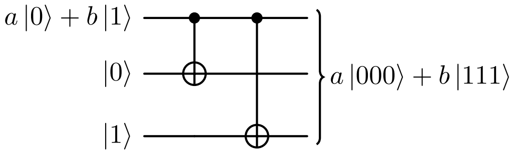
In fact, this code can correct one of $\{I, X_1, X_2, X_3\}$.

Definition 29.1
The **codespace** of an isometric encoding E is $\operatorname{im} E$.

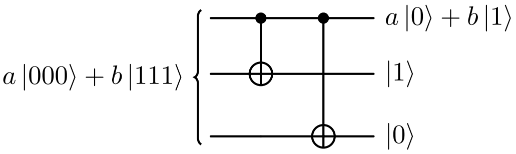
The **codespace** is $C = \operatorname{im} E = \operatorname{Span}\{\left|000\right\rangle, \left|111\right\rangle\}$, where $E(a\left|0\right\rangle + b\left|1\right\rangle) = a\left|000\right\rangle + b\left|111\right\rangle$. We have

$$X_1 C = \operatorname{Span}\{\left|100\right\rangle, \left|011\right\rangle\}, X_2 C = \operatorname{Span}\{\left|010\right\rangle, \left|101\right\rangle\}.$$

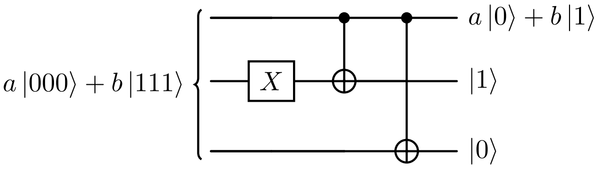
The encoding circuit is


$a\left|0\right\rangle + b\left|1\right\rangle$
$\left|0\right\rangle$
$\left|1\right\rangle$
$a\left|000\right\rangle + b\left|111\right\rangle$

To decode with no errors, we can take

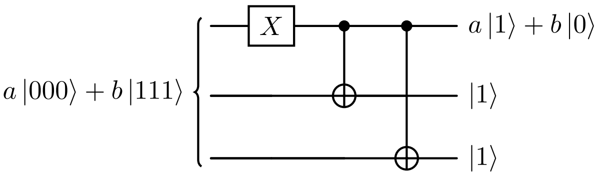

$a\left|000\right\rangle + b\left|111\right\rangle$
$a\left|0\right\rangle + b\left|1\right\rangle$
$\left|1\right\rangle$
$\left|0\right\rangle$

Assuming some error X_2, we get

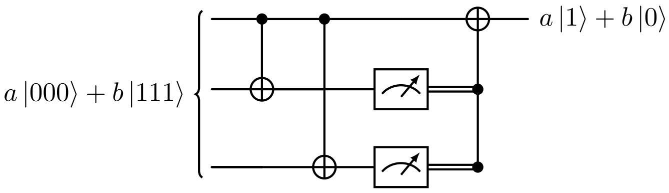

$a\left|000\right\rangle + b\left|111\right\rangle$
X
$a\left|0\right\rangle + b\left|1\right\rangle$
$\left|1\right\rangle$
$\left|0\right\rangle$

which does not affect the first qubit. The error X_3 will be equivalent.

Assuming some error X_1, we get

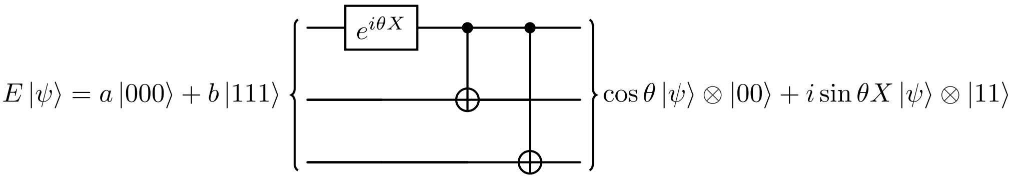

$a\left|000\right\rangle + b\left|111\right\rangle$
X
$a\left|1\right\rangle + b\left|0\right\rangle$
$\left|1\right\rangle$
$\left|1\right\rangle$

which provides the qubit $a\left|1\right\rangle + b\left|0\right\rangle$ in the first system rather than the desired $a\left|0\right\rangle + b\left|1\right\rangle$. Thus, in general, to correct for X_1, X_2, or X_3, we can take the decoding gate


$a\,|000\rangle + b\,|111\rangle$
$a\,|1\rangle + b\,|0\rangle$

Mathematically, this works, but this leaves an unprotected qubit in the first wire, which is vulnerable to noise. We managed to measure where the error was without collapsing the data, and it turns out that measurement collapse acts on continuous error to produce a discrete set of errors. The solution to measurement collapse is that we learn about errors, not logical qubits.

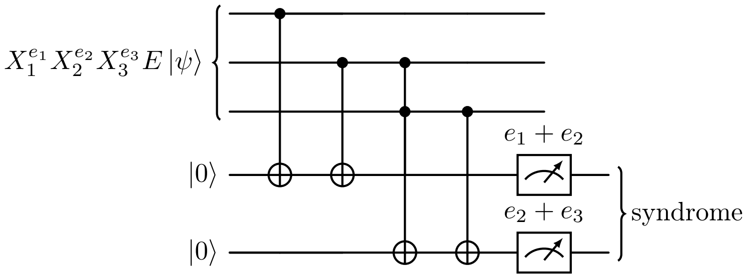
The solution to continuous errors is to collapse it by syndrome measurement. For example, with error $e^{i\theta X}$ in the first bit, taking $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$, here we get


$E\,|\psi\rangle = a\,|000\rangle + b\,|111\rangle$
$e^{i\theta X}$
$\cos\theta\,|\psi\rangle \otimes |00\rangle + i\sin\theta X\,|\psi\rangle \otimes |11\rangle$

The syndrome will collapse down to one of the errors, which will be able to be corrected.

Thus, discrete errors are sufficient for any linear combination.

In general, we get

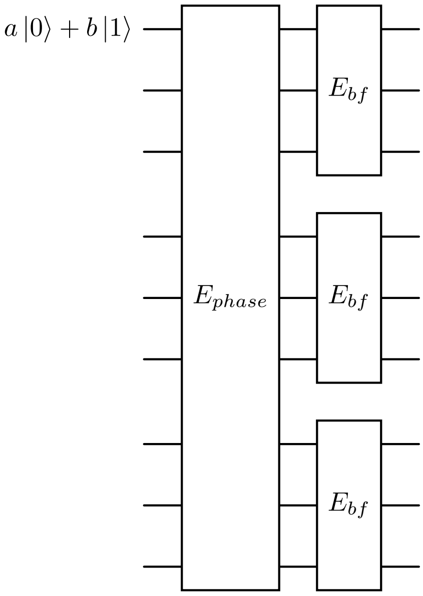


Measuring yields the syndrome, which is equivalent to parity checks $(1,1,0)$ and $(0,1,1)$. Based on the error correction syndrome, we get $00 \to I, 01 \to X_3, 10 \to X_1, 11 \to X_2$. This is the same as the classical case.

However, we may still have Z or phase errors. Take $E\,|0\rangle = H^{\otimes 3}\,|000\rangle = |{+}{+}{+}\rangle$ and $E\,|1\rangle = |{-}{-}{-}\rangle$. Then, $Z_1 C = \text{Span}\{|{-}{+}{+}\rangle, |{+}{-}{-}\rangle\}$, and so on, and we can similarly correct $\{I, Z_1, Z_2, Z_3\}$. We have $Z_i(a\,|000\rangle, b\,|111\rangle) = a\,|000\rangle - b\,|111\rangle = EZ(a\,|0\rangle + b\,|1\rangle)$, so X errors take probabilities $p \to 3p^2 = O(p^2)$, while Z errors take $p \to 3p = O(p)$.

29.2 9-Qubit Shor Code

This leads us to the 9-qubit Shor code. Let E_{bf} be the 3-qubit bit-flip encoding and E_{phase} be the 3-qubit phase-flip encoding. We concatenate the bitflip and phase flip encodings. We have an "outer code" and an "inner code"



Since $E_{bf}|\pm\rangle = \frac{|000\rangle \pm |111\rangle}{\sqrt{2}}$, this yields

$$a\left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3} + b\left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right)^{\otimes 3}.$$

In fact, the 9-qubit Shor code can correct any single-qubit error in $\{I, X_1, X_2, X_3, Y_1, Y_2, Y_3, Z_1, Z_2, Z_3\}$.

1. Decode each 3-qubit block and find the X errors, then output 3 qubits
2. Decode these 3 qubits using the phase flip code

A Y-error is simply the product of one X and one Y error, so it can also be corrected.

30 Missing

31 CSS Codes, Continued

31.1 CSS Codes

Today, we will finish covering the theory of CSS codes, which are inspired by classical linear codes, and cover two important examples, the toric code and stabilizer codes.

Let C_2 be any linear code, or a subspace of $\mathbb{F}_2^n$. Let $d_2 = \dim C_2$. Consider $H^{\otimes n} |a + C_2\rangle$. The Hadamard gate can be thought of as the 2-dimensional Fourier transform. A Fourier transform of a translation becomes a phase shift, and the Fourier transform of a uniform superposition becomes a delta function. Therefore, we expect $H^{\otimes n} |a + C_2\rangle$ to map to the orthogonal subspace, which we can compute.

$$\begin{aligned} H^{\otimes n} |a + C_2\rangle &= H^{\otimes n} 2^{-d_2/2} \sum_{x \in C_2} |a + x\rangle \\ &= 2^{-(n+d_2)/2} \sum_{y \in \{0,1\}^n} \sum_{x \in C_2} (-1)^{y \cdot (a+x)} |y\rangle \\ &= 2^{-(n+d_2)/2} \sum_{y \in \{0,1\}^n} (-1)^{y \cdot a} \sum_{x \in C_2} (-1)^{y \cdot x} |y\rangle . \end{aligned}$$

To compute the sum over x, consider two cases:

- If $y \in C_2^\perp$, then $x \cdot y = 1$ for all $x \in C_2$, so $\sum_{x \in C_2} (-1)^{x \cdot y} = |C_2| = 2^{d_2}$.
- If $y \notin C_2^\perp$, there exists $z \in C_2$ such that $y \cdot z = 1$. Note that $w \mapsto w + z$ is a bijection on C_2, so $z + C_2 = C_2$. Thus, $\sum_{x \in C_2} (-1)^{x \cdot y} = \sum_x (-1)^{(x+z) \cdot y} = (-1)^{z \cdot y} \sum_x (-1)^{x \cdot y} = -\sum_{x \in C_2} (-1)^{x \cdot y}$, which implies that $\sum_{x \in C_2} (-1)^{x \cdot y} = 0$.*

Thus, we get

$$H^{\otimes n} |a + C_2\rangle = 2^{-n/2 + d_2/2} \sum_{y \in C_2^\perp} (-1)^{a \cdot y} |y\rangle .$$

We have $2^{-(n-d_2)/2} = \frac{1}{\sqrt{|C_2^\perp|}}$, and $(-1)^{a \cdot y} |y\rangle = Z^a |y\rangle$, so this is

$$= Z^a |C_2^\perp\rangle .$$

Example 31.1

Consider $H^{\otimes 4} |C_2\rangle = |C_2^\perp\rangle$, where $C_2 = \text{Span}\left\{ \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \right\}$, and $C_2^\perp = \text{Span}\left\{ \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \right\}$. Then, $|C_2\rangle = \frac{1}{2}(|0000\rangle + |0100\rangle + |1000\rangle + |1100\rangle) = |{+}{+}00\rangle$, and $|C_2^\perp\rangle = |00{+}{+}\rangle$.

Moreover, we have $|a + C_2\rangle = X^a |C_2\rangle$†, so

$$H^{\otimes n} X^a |C_2\rangle = (H^{\otimes n} X^a H^{\otimes n}) H^{\otimes n} |C_2\rangle = Z^a |C_2^\perp\rangle .$$

31.2 Z-Error Correction

We try to bring quantum error correction into the realm of classical error correction. We assumed that C_1 could correct X-errors, and $C_2^\perp$ could correct Z-errors.

We have $\text{CSS}(C_1 : C_2) = \text{Span}\{|x + C_2\rangle : x \in C_1\}$, and $H^{\otimes n}\, \text{CSS}(C_1 : C_2) = \text{CSS}(C_2^\perp : C_1^\perp)$. We claim that $\text{CSS}(C_1 : C_2)$ can correct X errors like C_1 and correct Z errors like $C_2^\perp$.

- Start with a codeword $\sum_{a \in C_1} \psi_a |a + C_2\rangle$.‡

*This is a similar argument as to why the sum over roots of unity is zero, since we are summing over a group.

†Note that $X^a |b\rangle = |a + b\rangle$, where $+$ is considered in $\mathbb{F}_2^n$.

‡There is some redundancy here, as the same term may appear multiple times in the sum, which is fine. Recall that $a + C_2 = a' + C_2$ if and only if $a - a' \in C_2$, where $a - a' = a + a'$ for $\mathbb{F}_2$.

- Now, consider a Z^b error, which gives us $\sum_{a\in C_1} \psi_a Z^b |a+C_2\rangle$.
- Applying the Hadamard yields $\sum_{a\in C_1} H^{\otimes n}\psi_a Z^b |a+C_2\rangle = \sum_{a\in C_1} \psi_a H^{\otimes n} Z^b |a+C_2\rangle$. Like we did with the X, we move the Hadamard past the Z-error:

$$\sum_{a\in C_1} H^{\otimes n}\psi_a Z^b |a+C_2\rangle = \sum_a \psi_a (H^{\otimes n} Z^b H^{\otimes n})(H^{\otimes n} |a+C_2\rangle) = \sum_a \psi_a X^b Z^a |C_2^{\perp}\rangle .$$

  When measuring in the standard basis, the only effect is from X^b, which brings us back to the realm of classical error correction. Using parity checks and checking the syndrome as usual, we can apply classical error-correction techniques.

  Adding ancilla qubits, we do CWOTS, where we get $\sum_a \psi_a X^b Z^a |C_2^{\perp}\rangle |bH\rangle$, where the second register is the syndrome and H is the parity check matrix of $C_2^{\perp}$. This is analogous to our X-error correction. We inherit the guarantee from classical codes: assuming that b is correctable by C_2, we can learn b from the syndrome and correct it by applying X^b, undoing the error, and apply $H^{\otimes n}$ again to bring the code state back to where it started.

A big goal in error correction is self-correcting quantum memory. A classical analogy is the magnetic region in the memory of a computer. In a ferromagnet, each bit wants to point in the same direction of its neighbors, and if one of the bits is accidentally flipped, it's more energetically favorable to flip back to its neighbors.

31.3 Examples

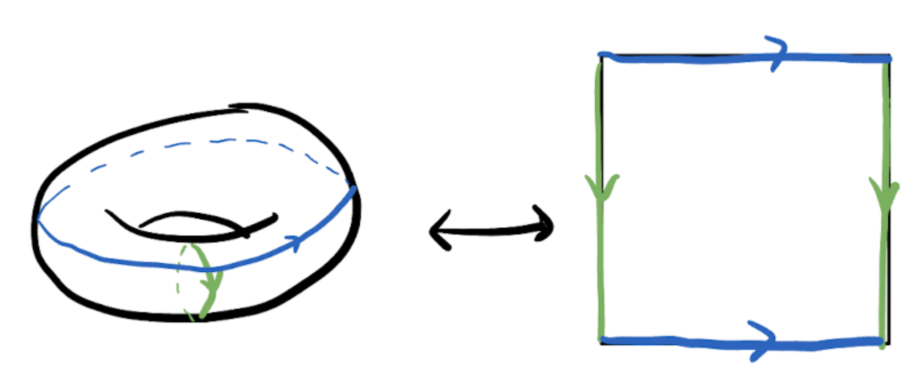
We can look at some examples of CSS codes.

Example 31.2 (Shor Code)

In fact, the Shor code is an example of a CSS code. We have $|0_L\rangle, |1_L\rangle = \left(\frac{|000\rangle \pm |111\rangle}{2}\right)^{\otimes 3}$. Modifying this a little, we take $|0\rangle \mapsto |v_0\rangle = |+++\rangle + |---\rangle$, where only the even Hamming weights will contribute so we get $|v_0\rangle = \frac{|000\rangle+|110\rangle+|101\rangle+|011\rangle}{2}$, and $|1\rangle \mapsto |v_1\rangle = \frac{|+++\rangle-|---\rangle}{2}$, so only the odd Hamming weights contribute: $|v_1\rangle = \frac{|001\rangle+|010\rangle+|100\rangle+|111\rangle}{2}$. Now, we let $|0_L\rangle = |v_0\rangle^{\otimes 3}$ and $|1_L\rangle = |v_1\rangle^{\otimes 3}$.

Example 31.3 (Toric Code)

Consider a torus, which can be drawn as a square with the sides identified.

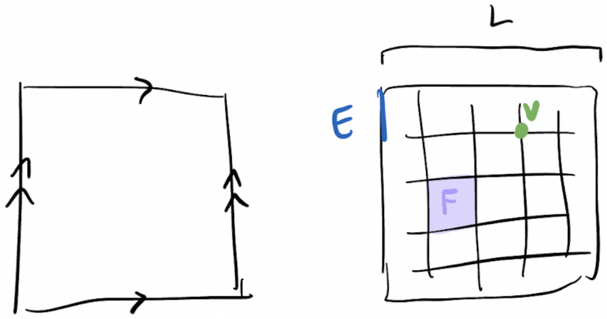


Now, create a grid and place a qubit on each edge. One feature of a code is LDPC, which stands for low density parity check, and means that each parity check does not check many bits. Another feature is for a code to be spatially local, where qubits only depend on nearby qubits. Next time, we will finish discussing the toric code.

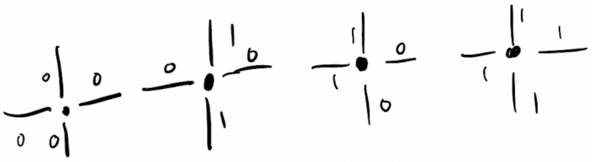
32 Toric Code

32.1 Toric Code

The toric code is on a torus, which we can identify with a square with edges identified. We create an $L \times L$ grid. Let V be the L^2 vertices, E be the $2L^2$ edges, and F be the L^2 faces. Then, take $n = 2L^2$ qubits on the edges.

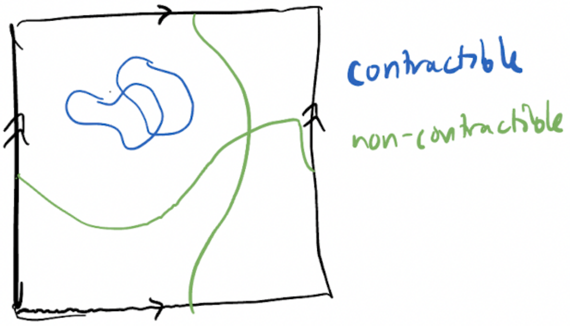


We have $\mathrm{CSS}(C_1 : C_2) = \mathrm{Span}\{|a + C_2\rangle : a \in C_1\}$, where $C_1 \supseteq C_2$. We can index by the edges to get $\mathbb{F}_2^E = \mathbb{F}_2^{2L^2} = \mathbb{F}_2^n$. Then, $C_1 = \{x \in \mathbb{F}_2^E$ such that $\forall v \in V, \sum_{e \sim v} V = 0$.



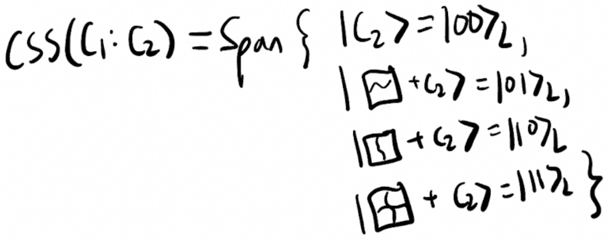
In physics, this is a gauge condition and the toric code is $\mathbb{Z}_2$-gauge theory.

Draw an edge where $x_e = 1$; this is an example of something that satisfies these constraints. We have closed curves, which can be contractible or non-contractible.



Now, let $C_2 = \mathrm{Span}\{$ for all $f \in F\}$.

The set C_2 includes contractible loops, as well as pairs of horizontal or vertical non-contractible loops. So



. We can have logical X operators, which act on edges, and logical Z operators, which act on faces.

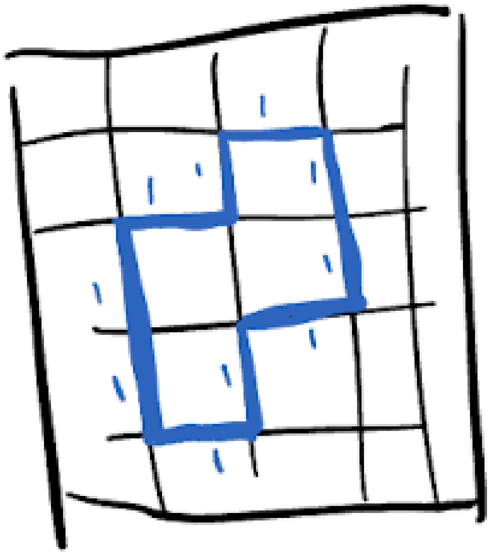


Figure 1: C_1

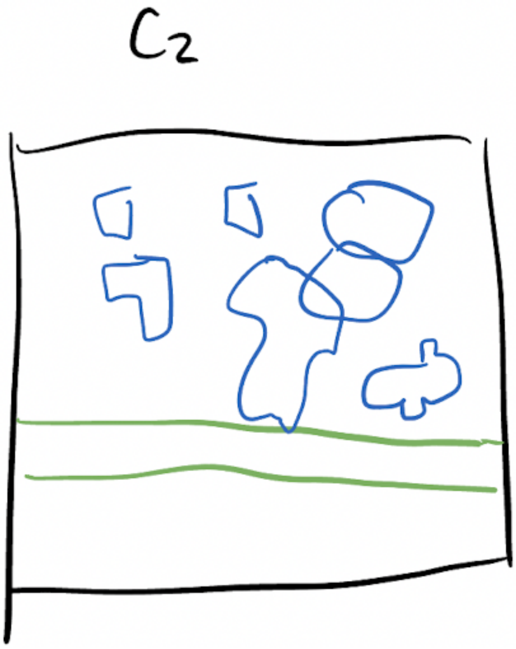
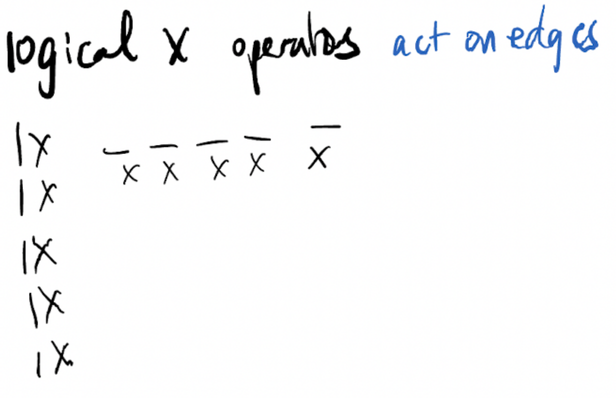
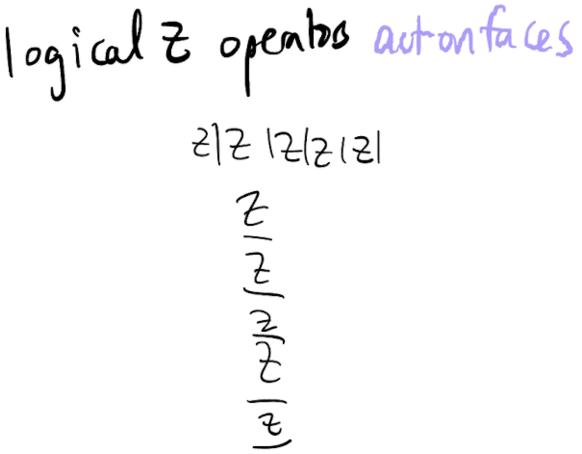


Figure 2: C_2

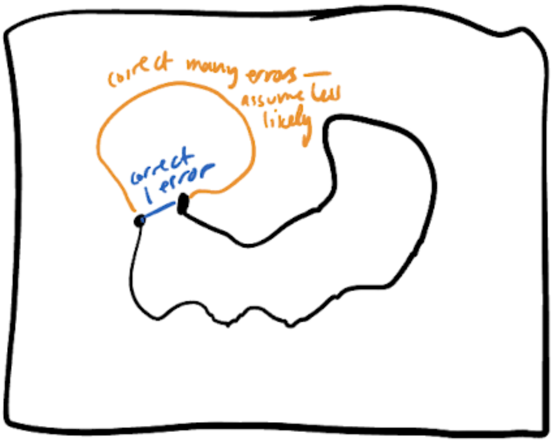
Let d be the distance, which is the weight of a non-identity logical operator. The weight of an operator is the number of nonzero qubits. So $d = L \sim \sqrt{n}$.



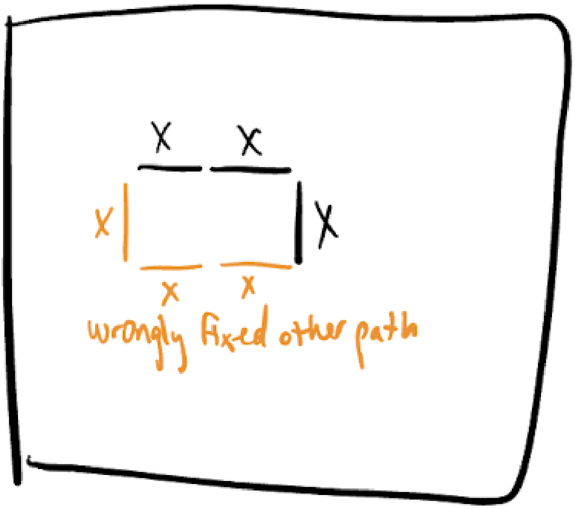


Let's consider X errors. Given an X error, we can see their endpoints. The vertices correspond to the constraints (constraints are satisfied for each vertex).

The syndrome is the set of parity checks that are violated, where the parity checks are on each vertex. From the "unhappy" vertices, we can find edges to fill in the path between them, using the shortest possible path by assuming that there are "few" or "sparse" errors.



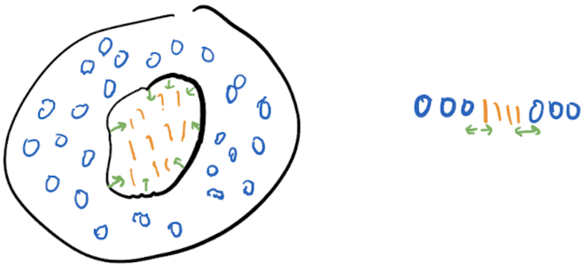
Suppose in reality that there were three X errors, so we have two endpoints with a distance of three, and we corrected it using three X errors, but on the wrong path between the endpoints. Since the wrong path and the right path of X errors actually have the same length, it's ambiguous which has fewer errors. It turns out that the right path (which we didn't correct) and the wrong path (which we wrongly corrected) form a loop, and in fact since the cosets of $|C_2\rangle$ are invariant under closed loops, it doesn't actually matter which path we used. So as long as we pair the right endpoints, it doesn't matter what loops we use to close them.



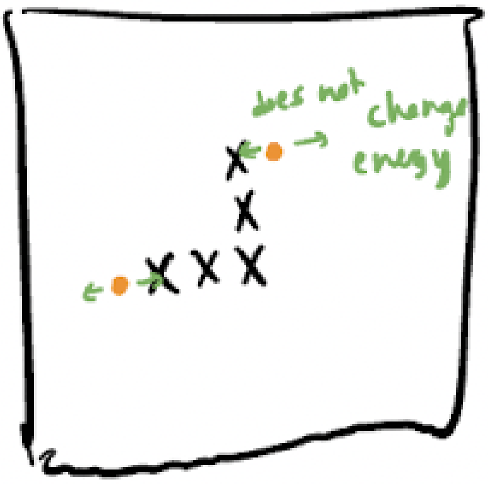
32.2 Self-Correcting Memory

This is a more technically vague ramble that we will discuss now, in a very broad sense.

In a self-correcting classical memory, let the energy be the number of parity checks violated. The code space has zero energy and more incorrect parity checks have higher energy. Then the lowest energy state is the code space. (This is how ferromagnetism works, which is essentially a repetition code. If a bit has neighbors pointing up, it will have lower energy if it also points up, so if the bit is wrongly flipped to down, it is likely to flip back to up, which is a lower-energy state.) In two dimensions, if a "bubble" of 1s appears, then the energy scales with the perimeter or boundary. Thus, the bubble will shrink automatically due to energy minimization, which is a kind of "surface tension." In one dimension, there is only an energy difference of 2, but it doesn't shrink when the defect of 1s shrinks, so the endpoints will do a simple random walk, and might collide and disappear, but might also increase the "bubble" or defect. So one dimension is not good for a stable classical memory, while two dimensions is good.



People want a "self-correcting" quantum memory. For the toric code, given a pair of defects, thinking of the energy as the number of constraints violated, making a chain longer or shorter will also only move the defects. So there is no "string tension" pulling the endpoints/chain back together. Unfortunately, the toric code is not a self-correcting memory for the same reason that a 1D repetition code is not.



However, we could potentially create a "self-correcting" quantum memory using "sheet" operators, rather than "string" operators. Then we need X errors that are planes, and Z errors that are planes, which leads to four dimensions. So unfortunately, we can do self-correcting quantum memory (provably) in four dimensions, but so far we don't have anything for three dimensions.

33 Quantum Key Distribution

Quantum key distribution addresses how to obtain quantum keys. Classically, people will use protocols such as RSA, which rely on a computational assumption that an eavesdropper cannot solve some kind of difficult computational problem. We cannot have information theoretic security, where the keys that Alice and Bob send do not have enough information for an eavesdropper, which quantum key distribution solves.

Assume we have Alice and Bob, with an eavesdropper Eve, where Alice and Bob have a quantum channel that Eve can see and disrupt, as well as an authenticated classical channel that Alice and Bob can send messages through and Eve can see but not disrupt, such as the internet. Some people call this "quantum key expansion," where we assume that we have an authenticated classical channel. We will later see some benefits of the quantum channel, relative to the classical channel. One example comes from the no-cloning theorem, where Eve cannot copy the information without measuring and causing detectable damage to the message.

Proposition 33.1 (Chernoff Bound)
The binomial distribution is such that $P[\mathrm{Bin}(n,p)=k]=\binom{n}{k}p^k(1-p)^{n-k}$. The Chernoff bound states that

$$P(|\,\mathrm{Bin}(n,p)-np|\geq n\delta)\leq e^{-n\delta^2/2}.$$

Example 33.2 (BB84 Encryption)
Alice picks random bases $b_1,\cdots,b_n$ and key bits $k_i,\cdots,k_n$. For example, she might pick random bases $|0\rangle,|1\rangle$ and $|+\rangle,|-\rangle$, and key bits 11001000, which yields $|0\rangle,|1\rangle,|+\rangle,|+\rangle,|0\rangle,|-\rangle,|-\rangle,|1\rangle$.

Then, Bob picks random bases and measures, which yields for example $|0\rangle,|+\rangle,|+\rangle,|1\rangle,|0\rangle,|1\rangle,|-\rangle,|1\rangle$. Then, they both announce their bases and keep positions with the same bases. The length is approximately $n/2$. We check a random sample for differences and discard these bits. We get $0,+,0,+$, $+/-$, which goes to $0/1$. This is pretty accurate, by the Chernoff bound.

So first, Alice sends n qubits, and does not send Bob her encoding bases until he's received all the qubits.

Example 33.3 (Intercept-Resend Attack)
Eve chooses a random basis, measures, and sends Bob the outcome instead. Eve can either measure the qubits before Bob receives Alice's basis, or she can know the bases once she sends on the qubits to Bob, at which point she no longer has the qubits.

Half the time, Eve will guess the correct basis to measure on, so $P(\text{wrong basis for each bit})=1/2$. If she gets the basis wrong, she will cause some damage. If Alice sends $|0\rangle$, and Eve measures in the wrong basis and it collapses to $|+\rangle$, and Bob measures again in $|0\rangle,|1\rangle$, then Bob will get the correct $|0\rangle$ with probability $1/2$. Similarly, if Eve measures in the wrong basis and it collapses to $|-\rangle$, then Bob will similarly get the correct $|0\rangle$ with probability $1/2$. Thus, $P(\text{bits disagree}|\text{wrong basis})=1/2$ as well. Then, the error rate will be $1/4$: Bob will measure the correct qubit with probability $1/4$, assuming that Eve guesses the correct basis $1/2$ the time.[a]

So if the error rate is $1/4$ from Alice to Bob, they will assume that there may be an eavesdropper. However, in general in quantum systems, there will usually be some error rate, say 5%. So Alice and Bob need information reconciliation, and privacy amplification.

Using classical linear codes, Alice gets $a\in\mathbb{F}_2^m$ and Bob gets $b=a+e$, where $\mathrm{wt}(e)\approx\varepsilon m$. Alice and Bob agree on a code with check matrix H. Alice sends aH, which is sacrificed (Eve gets a copy of it), and Bob computes $(a+b)H=eH$, which allows him to calculate aH from the channel, where Bob knows b or e, so this keeps $\dim\ker H$ bits.

[a] This is not the only interception that Eve can do, and it gets a little more complicated, but the analysis can be reduced to more simple cases. For now, we only analyze this case.

33.1 Privacy Amplification

Alice and Bob both know $a \in \mathbb{F}_2^m$, publish random $M \in \mathbb{F}_2^{m\times\ell}$, $\ell \leq m$. The secret key is aM. Suppose Eve knows each bit with probability 2/3. Then, if $M = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$, then aM is the parity. So the probability that Eve can guess aM is at most $1/2 + e^{-O(m)}$.

Let $a = 011, e = 010, b = 001, H = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, aH = (10), bH = (01)$.

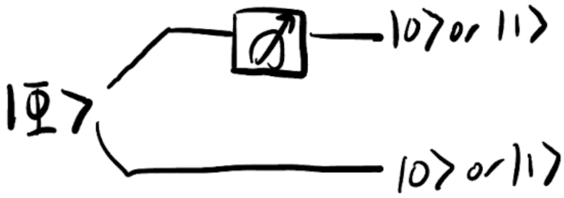
34 More Quantum Key Distribution and Density Matrices

Today, we will finish talking about QKD and introduce density matrices.

For quantum key distribution, entanglement is not necessary, and it suffices to send and receive single qubits. However, for the proof, considering entanglement will be helpful, only as a mathematical concept.

Consider the EPR pair $|\Phi\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$.

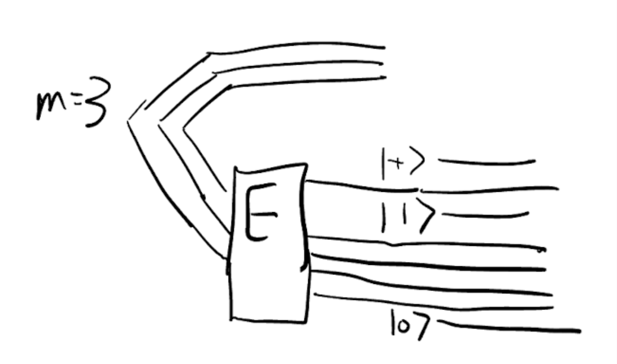
If Alice and Bob both have $|\Phi\rangle$, let Alice choose a basis $|0\rangle, |1\rangle$ or $|+\rangle, |-\rangle$, then measures and gets a random outcome. If Bob measures, he will get the same state.[*]



Now, we will discuss a protocol that is equivalent to the protocol that we talked about previously, but is more technologically difficult to implement and mathematically easier to analyze.

Protocol 34.1 (Lo-Chau QKD)
Consider $C_1 \subseteq C_2 \subseteq \mathbb{F}_2^n$ such that $\dim C_1 - \dim C_2 = m$. Alice prepares $|\Phi\rangle^{\otimes m}$. Then, Alice encodes half in $X^s Z^t \, \mathrm{CSS}(C_1 : C_2)$ for random s, t, and intersperses test qubits randomly set to $|0\rangle, |1\rangle, |+\rangle, |-\rangle$.[a]

In this diagram, the encoding has $m = 3$ and 5 output qubits, with three interspersed test qubits.



The encoding protects Alice's qubits, while the test qubits will measure how much noise Eve is introducing (there is natural noise, which the encoding protects against, as well as noise from Eve).

Next, Bob announces receipt of the qubits.

Then, Alice announces s, t and the identities and bases of the test qubits.

Next, Bob measures each of the test qubits, from which he can check whether they match the original test qubits, and estimate the amount of channel noise using the Chernoff bound. So Bob now has the error rate through the channel, and he can decode.

Now, he has obtained $|\Phi\rangle^{\otimes m}$, and both can measure in the $|0\rangle, |1\rangle$ basis and get m bits of the key, due to entanglement.

[a] The CSS code can be known by Eve, but it's not useful because it's randomly shifted. The normal repetition code might be 000 and 111, but the shifted repetition code would be 100 and 011, which is no longer helpful to Eve.

Encoding.

Let's look at this more carefully. We have $|\Phi\rangle^{\otimes m} = \frac{1}{\sqrt{2}^m} \sum_{x \in \mathbb{F}_2^2} |x\rangle |x\rangle$.

[*] In general, with different bases, Bob will get the same state except with the coefficients as the complex conjugates.

The CSS code works with cosets. Consider $\{|x + C_2\rangle : x \in C_1/C_2\}$. There are 2^m cosets, and we can choose a coset representative for each coset, which is an arbitrary choice. Choose $x_0, \cdots, x_{2^m-1}$ such that each $x_i + C_2$ is distinct. Then, we can define $C_1/C_2 = \{x_0, \cdots, x_{2^m-1}\}$. This labels our 2^m codewords, so we can encode numbers 0 through $2^m - 1$.

Now, we have

$$\frac{1}{\sqrt{2}^m} \sum_{x\in C_1/C_2} |x\rangle |x\rangle \xrightarrow{\text{encode}} \frac{1}{\sqrt{2}^m} \sum_{x\in C_1/C_2} |x\rangle \sum_{y\in C_2} \frac{(-1)^{t\cdot(x+y)}}{\sqrt{|C_2|}} |x + y + s\rangle \,,$$

which is a shifted CSS code.

Decoding. Why are we doing this? We already had a perfectly good description of this encoding. Now, we can analyze more easily mathematically.

Quantum error correction corrects superpositions of "all possible attacks," so this protocol will be secure against general attacks, but this is equivalent to the kind of QKD protocol we have already seen (which we will see next). If this CSS code can decode 10% of the qubits, and Eve does any superposition of whatever she wants on 10% of the qubits, this code will correct it. If Eve messes with more than 10% of the qubits, Bob will be able to detect it based on the test qubits[†], and they can abort the protocol.

Correcting X errors.

In this case, Bob will get some superposition of $|x + y + s + e\rangle$, where e is the vector of X errors induced by Eve. Assume that $\mathrm{wt}(e)$ is small. We have $\ker H = C_1$, and since $x, y \in C_1$, $(x + y + s + e)H = eH + sH$, and since Alice announced s, Bob knows s and can obtain eH, the syndrome, which can be used to diagnose the error.

Now, Bob knows $x + y$ for a random $y \in C_2$, which yields the output x.

Privacy Amplification. Choose a matrix M and output $(x+y)M = xM$, which uniquely determines $x \in C_1/C_2$, the coset representative. Then, choose $C_2 = \ker M$.

34.1 Density Matrices

The following ensembles are indistinguishable:

- $|0\rangle$ with probability 1/2, $|1\rangle$ with probability 1/2
- $|+\rangle$ with probability 1/2, $|-\rangle$ with probability 1/2
- One qubit of $|\Phi\rangle$

We can describe these using density matrices.

34.2 Observables

Measure in an orthonormal basis $|\varphi_1\rangle, \cdots, |\varphi_d\rangle$, and output x_i upon outcome $|\varphi_i\rangle$.

We have $\mathbb{E}[x|\,|\psi\rangle] = \sum_{i=1}^d |\langle\varphi_i|\psi\rangle|^2 x_i = \langle\psi| \sum_{i=1}^d x\,|\varphi_i\rangle\langle\varphi_i|\,|\psi\rangle = \langle\psi|X|\psi\rangle$, where $X = \sum_{i=1}^d x\,|\varphi_i\rangle\langle\varphi_i|$ is an observable. This is just a "change of syntax" from our previous formulation.

The density matrix ρ will tell us $\mathbb{E}[X]$ for any observable X. We have $\mathbb{E}[X] = \mathrm{Tr}[\rho X]$.

So in the ensemble, we only really want an expected value or average.

[†] The test qubits require more technology, which is why it's harder to implement. But it's easier to analyze.

35 Density Matrices

35.1 Operator Formalism

For some operator Λ, where $|\varphi_i\rangle$ are the eigenvectors with eigenvalues λ_i, the eigenvectors are orthonormal if the operator is Hermitian, so we can write $\Lambda = \sum_i \lambda_i |\varphi_i\rangle\langle\varphi_i|$. Then, measuring the observable λ on the state ψ yields

$$P(\lambda = \lambda_i) = |\langle\psi|\varphi_i\rangle|^2.$$

The expected value of the observable λ can be denoted

$$\mathbb{E}[\lambda] = \langle\Lambda\rangle = \langle\psi|\Lambda|\psi\rangle.$$

All of this is the same as our previous conceptualization of measurement, simply packaged differently.

35.2 Ensembles

Consider an ensemble of quantum states, which is a probability distribution over quantum states. An ensemble $|\psi_i\rangle$ with probability p_i can be written as $\{(p_i, |\psi_i\rangle)\}$. It's also possible to take a continuous distribution over quantum states, but for simplicity we use notation for discrete distributions.

In the beginning of the class, we talked about quantum amplitudes as analogous to classical probabilities, where the sum of quantum amplitudes norm squared is 1, while the sum of classical probabilities is 1 directly. Ensembles, where we have a random quantum state, combine classical probabilities with quantum states. Note that the ensemble $\{(p_i, |\psi_i\rangle)\}$ is different from $\sum_i \sqrt{p_i}|\psi_i\rangle$, which may not be a unit vector since the ψ_i do not have to be orthogonal to each other.

Now, considering $\mathbb{E}[\lambda]$ must be taken over two sources of randomness: this is the average over $\{p_i\}$ as well as the quantum measurement. Thus,

$$\begin{aligned}\mathbb{E}[\lambda] &= \langle\Lambda\rangle \\ &= \sum_{i=1}^{m} p_i \langle\psi_i|\Lambda|\psi_i\rangle.\end{aligned}$$

To separate out the ensemble-dependent quantities from the measurement-dependent quantities, we will write down some math.

Definition 35.1
The **trace** of a matrix is $\operatorname{Tr} M = \sum_i M_{ii} = \sum_i \langle i|M|i\rangle$.

Proposition 35.2

- Trace of products: $\operatorname{Tr}(AB) = \operatorname{Tr}(A)\operatorname{Tr}(B)$.
- **Cyclic property:** $\operatorname{Tr}(ABC) = \operatorname{Tr}(BCA) = \operatorname{Tr}(CAB)$
- **Trace of scalar:** $\operatorname{Tr}(a) = a$

If A and B are nonsquare, it is still possible to take the trace of $\operatorname{Tr}(AB)$ as long as AB is square.

Thus, viewing these quantities as matrices, $\operatorname{Tr}\langle\psi_i|\Lambda|\psi_i\rangle = \operatorname{Tr}\Lambda|\psi_i\rangle\langle\psi_i|$, so

$$\mathbb{E}[\lambda] = \sum_i p_i \operatorname{Tr}\Lambda|\psi_i\rangle\langle\psi_i| = \operatorname{Tr}\left(\Lambda\sum_i p_i \operatorname{Tr}|\psi_i\rangle\langle\psi_i|\right).$$

This suggests the definition of a density matrix.

Definition 35.3
The **density matrix** of an ensemble $\{(p_i, |\psi_i\rangle)\}$ is $\rho = \sum_{i=1}^{m} p_i|\psi_i\rangle\langle\psi_i|$, so $\langle\Lambda\rangle = \operatorname{Tr}(\Lambda\rho) = \operatorname{Tr}(\rho\Lambda)$.

Thus, ensembles with the same density matrix are indistinguishable with respect to measurement, and we could even consider them to be the same ensemble.

35.3 Examples of Density Matrices

Consider an example of ensembles which are called pure states, and capture information about quantum mechanics as we discussed at the beginning of this class.

Example 35.4 (Pure States)
For a pure state $\{(1, |\psi\rangle)\}$, then $\rho = |\psi\rangle\langle\psi|$. Now, $e^{i\phi}|\psi\rangle$ has density matrix $\rho = e^{i\psi}|\psi\rangle e^{-i\psi}\langle\psi| = |\psi\rangle\langle\psi|$, so our density matrix formalism captures the notion that an overall phase "doesn't matter."

Another class of ensembles capture information about probability distributions.

Example 35.5 (Probability Distributions)
Consider $\{(p_i, |i\rangle)\}$. Then, $\rho = \sum_{i=1}^{d} p_i |i\rangle\langle i| = \begin{pmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_n \end{pmatrix}$, which has the probabilities along the diagonal.

Thus, we see how density matrices generalize quantum mechanics as well as classical probability distributions.

Example 35.6 (Maximally Mixed State)
Consider the maximally mixed state $(1/2, |0\rangle), (1/2, |1\rangle)$. Then, $\rho = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$. Alternatively, consider the maximally mixed state $(1/2, |+\rangle), (1/2, |-\rangle)$. The density matrix is $\frac{1}{2}(|+\rangle\langle+| + |-\rangle\langle-|) = \frac{1}{2}I$, which is the same. These two different ensembles give rise to the same density matrix. Underlying the security of QKD is partially the fact that Eve could not perform any measurement that would give her information about the basis. The ensembles in the $|0\rangle, |1\rangle$ or in the $|+\rangle, |-\rangle$ basis look the same. In d dimensions, for an orthonormal basis $|v_1\rangle, \cdots |v_d\rangle$, the maximally state is $\rho = \frac{1}{d}\sum_{i=1}^{d} |v_i\rangle\langle v_i| = \frac{1}{d}I_d$.

In general, we have four categories of computation:

	deterministic	random
classical	Bitstrings $[d]$	Probability distributions $p_1, \cdots, p_n$
quantum	$\|\psi\rangle \in \mathbb{C}_d$, $\langle\psi\|\psi\rangle = 1$	Density matrices $\rho = \sum p_i \|\varphi_i\rangle\langle\varphi\|_i$

To determine whether a given matrix is a density matrix, we have the following conditions.

Theorem 35.7
If $A = A^{\dagger}$[a], then the following are equivalent:

- $\langle\psi|A|\psi\rangle \geq 0$ for all $|\psi\rangle$
- All eigenvalues of A are ≥ 0
- $A = B^{\dagger}B$ for some B.

If A satisfies any of these conditions, we say A is **positive semi-definite**.

[a] All Hermitian matrices have real eigenvalues and an orthonormal eigenbasis

Proof. Given (1), suppose $A|v\rangle = \lambda|v\rangle$ for $|v\rangle \neq 0$. Then, $\langle v|A|v\rangle = \lambda\langle v|v\rangle$, and since A is Hermitian, $\lambda \geq 0$, and clearly $\langle v|v\rangle \geq 0$.

Given (2), we show (3). The spectral theorem states that A has an orthonormal basis, so $A = \sum_i \lambda_i |v_i\rangle\langle v_i|$, where $\lambda_i \geq 0$. Then, writing $B = \sum_i \sqrt{\lambda_i}|w_i\rangle\langle v_i|$ yields $B^{\dagger}B = A$, where $|w_i\rangle$ is any other computational basis.

Given (3), we show (1):

$$\langle\psi|A|\psi\rangle = \langle\psi|B^{\dagger}B|\psi\rangle = ||B|\psi\rangle||^2 \geq 0.$$

□

Theorem 35.8

A matrix ρ is a valid density matrix if and only if

- **Trace is one**: $\mathrm{Tr}(\rho) = 1$
- **Positive semi-definite**: $\rho \geq 0$[a]

[a] Analogous to classical probabilities between 0 and 1.

Proof. Given a density matrix $\rho = \sum_i p_i |\varphi_i\rangle \langle\varphi_i|$, the trace is $\mathrm{Tr}\,\rho = \sum_i p_i \mathrm{Tr}\,|\varphi_i\rangle\langle\varphi_i|$. Then, $\mathrm{Tr}\,|\varphi_i\rangle\langle\varphi_i| = \mathrm{Tr}\,\langle\varphi_i|\varphi_i\rangle = \langle\varphi_i|\varphi_i\rangle = 1$. So $\mathrm{Tr}\,\rho = \sum_i p_i(1) = 1$. Moreover, taking $B = \sum_i \sqrt{p_i}\,|w_i\rangle\langle\varphi_i|$, we can compute that $\rho = B^\dagger B$ for any orthonormal basis $|w_i\rangle$.

Given ρ such that $\mathrm{Tr}\,\rho = 1$ and $\rho \geq 0$, using the spectral theorem, $\rho = \sum_{i=1}^d \lambda_i |v_i\rangle\langle v_i|$, where from ρ being positive semi-definite, $\lambda_i \geq 0$. Since the trace is 1, this corresponds to $\sum_i \lambda_i = 1$. Then this is directly a density matrix where $|v_i\rangle$ are orthonormal.

□

Example 35.9

Consider $\{(1/2, \sqrt{3/4}\,|0\rangle + \sqrt{1/4}\,|1\rangle), (1/2, \sqrt{3/4}\,|0\rangle - \sqrt{1/4}\,|1\rangle)\}$, which is a random phase on $|1\rangle$. Then, the density matrix is

$$\rho = \frac{1}{2}\begin{pmatrix} 3/4 & \sqrt{3}/4 \\ \sqrt{3}/4 & 1/4 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 3/4 & -\sqrt{3}/4 \\ -\sqrt{3}/4 & 1/4 \end{pmatrix} = \begin{pmatrix} 3/4 & 0 \\ 0 & 1/4 \end{pmatrix},$$

which corresponds to a different ensemble $\{(3/4, |0\rangle), (1/4, |1\rangle)$.

36 Density Matrices

36.1 2-Dimensional Density Matrices

Consider $\rho \in \mathbb{C}^{d\times d}$ such that $\rho = \rho^\dagger$ and $\rho \geq 0$*, and $\operatorname{Tr}\rho = 1$.

Consider $d = 2$. There are 4 complex numbers, which is 8 real degrees of freedom. The Hermitian condition provides 4 constraints, so there are 4 real degrees of freedom. Thus, for $a_0, \cdots, a_3 \in \mathbb{R}$, $\rho = \frac{a_0 I + a_1\sigma_1 + a_2\sigma_2 + a_3\sigma_3}{2}$, where we divide by 2 to make future computations easier. We have $\operatorname{Tr}\rho = I = a_0$, and $\vec{a} = (a_1, a_2, a_3)$, so $\rho = \frac{I + \vec{a}\cdot\vec{\sigma}}{2}$. The eigenvalues of $\vec{a}\cdot\vec{\sigma} = \pm|\vec{a}|$.† Thus, $\operatorname{eig}(\rho) = \frac{1\pm|\vec{a}|}{2}$.

To be positive semi-definite, we need $\frac{1\pm|\vec{a}|}{2} \geq 0$, so $|\vec{a}| \leq 1$. This is a ball for $\vec{a}$, which is called the Bloch ball. Earlier, we talked a little about the Bloch sphere. Given a pure qubit $\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$, then a spin-1/2 in the direction $\vec{v}$ can be written as $\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle = \begin{pmatrix} \cos\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} \end{pmatrix}$. Then, the density matrix is

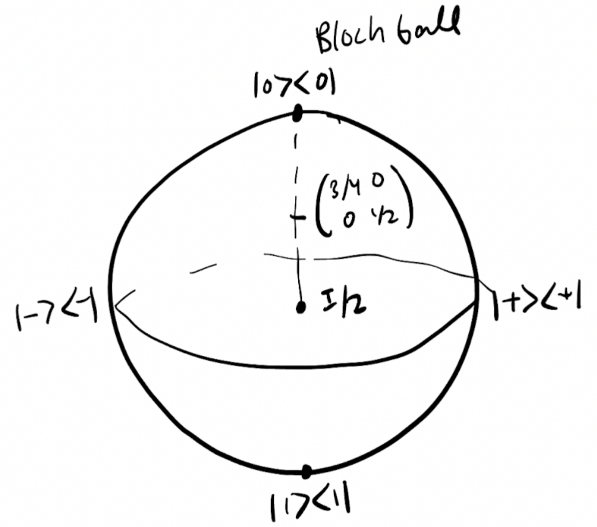
$$|v\rangle\langle v| = \begin{pmatrix} \cos^2\frac{\theta}{2} & e^{-i\phi}\cos\frac{\theta}{2}\sin\frac{\theta}{2} \\ e^{i\phi}\cos\frac{\theta}{2}\sin\frac{\theta}{2} & \sin^2\frac{\theta}{2} \end{pmatrix}.$$

We can use trigonometric identities and simplify to get

$$\begin{pmatrix} \frac{1+\cos\theta}{2} & \frac{e^{-i\phi}\sin\theta}{2} \\ \frac{e^{i\phi}\sin\theta}{2} & \frac{1-\cos\theta}{2} \end{pmatrix} = \frac{I}{2} + \frac{1}{2}\cos\theta\sigma_z + \cdots = \frac{I + \vec{v}\cdot\vec{\sigma}}{2}.$$

Note that the pure states have $|\vec{a}| = |\vec{v}| = 1$, so they lie on the surface of the sphere. On the other hand, mixed states will have $|\vec{a}| < 1$, and lie on the interior of the sphere.

When $\vec{a} = \vec{0}$, the density matrix will be $\frac{I}{2}$, which is the maximum mixed state. Similarly, $\begin{pmatrix} 3/4 & 0 \\ 0 & 1/4 \end{pmatrix}$ will lie on the line between $|0\rangle\langle 0|$ and $I/2$.



The eigenvalues measure how "mixed" a state is, while eigenvectors measure the directions. Density matrices generalize probability distributions to quantum states.

Consider an observable M, $\langle M\rangle = \operatorname{Tr}[\rho M]$. Then, we have $\rho = \frac{I+\vec{a}\cdot\vec{\sigma}}{2}$. For $M = \sigma_j$, $\operatorname{Tr}[\rho M] = a_j$, since $\operatorname{Tr}\frac{1}{2}\sum_{i=0}^{3} a_i\sigma_i \cdot \sigma_j = a_j$, so if $\delta_0 = I$, then $\operatorname{Tr}\sigma_i\sigma_j = 2\delta_{ij}$.

If you are maximally mixed, a will be zero so every measurement will be random. Reconstructing the density matrix from measurements is called *quantum state tomography.*

*Positive semi-definite

†Remember this by $(0, 0, a_3)\cdot\vec{\sigma}$, which has $\pm a_3$ on the diagonals, and rotation doesn't change the eigenvalues.

36.2 Thermal States

Consider the system in state x that has energy E_x. At thermal equilibrium, $Pr(x) = \frac{e^{-\beta E_x}}{Z}$, where $Z = \sum_x e^{-\beta E_x}$, and $|beta = \frac{1}{k_B T}$, where we can set $1 = K_B = 1.38 \cdot 10^{-23} \frac{J}{K}$. This says that the chance of having high energy goes down exponentially with energy.

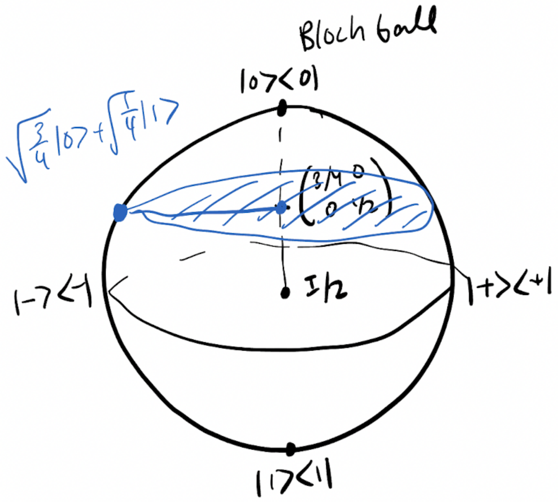
The Hamiltonian $H = \sum_x E_x |\varphi_x\rangle \langle\varphi_x|$. Then $\rho_\beta = \sum_x \frac{e^{-\beta E_x}}{Z} |\varphi_x\rangle \langle\varphi_x| = \frac{e^{-\beta H}}{\operatorname{Tr} e^{-\beta H}}$. If $T \to \infty, \beta \to 0, \rho_\beta \to \frac{I}{d}$ and if $T \to 0, \beta \to \infty, \rho_\beta \to |\varphi_0\rangle \langle\varphi_0|$.

A lot of things are thermal states, but usually we don't let quantum computers completely equilibriate so they are not usually thermal states.

36.3 Dynamics

Let $|\psi\rangle \to U|\psi\rangle$, and $|\psi\rangle\langle\psi| \to U|\psi\rangle\langle\psi|U^\dagger$. As a linear combination, $\rho \to U\rho U^\dagger$. In the qubit case, a unitary corresponds to a rotation of the Bloch ball.

To measure, measure in $|v_1\rangle, \cdots, |v_d\rangle$ basis, Then, $Pr(i) = \langle v_i|\rho|v_i\rangle$, where the post-measurement state is $|v_i\rangle$. An unknown outcome, the state is $\sum_i \langle v_i|\rho|v_i\rangle |v_i\rangle\langle v_i| = \sum_i |v_i\rangle\langle v_i| \rho |v_i\rangle\langle v_i|$. Then, for $|v_1\rangle = |1\rangle, \cdots, |v_d\rangle = |d\rangle$, then $\rho \mapsto \begin{pmatrix} \rho_{11} & 0 & \cdots & 0 \\ 0 & \rho_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \rho_{dd} \end{pmatrix}$. This corresponds to zeroing out the off-diagonal elements, and on the Bloch ball corresponds to zeroing out the x and y elements.


Bloch ball
$|0\rangle\langle 0|$
$\sqrt{\frac{3}{4}}|0\rangle + \sqrt{\frac{1}{4}}|1\rangle$
$\begin{pmatrix} 3/4 & 0 \\ 0 & 1/2 \end{pmatrix}$
$|-\rangle\langle -|$
$I/2$
$|+\rangle\langle +|$
$|1\rangle\langle 1|$

This is called decoherence, where T_1 is the time to reach thermal equilibrium and T_2 is dephasing: $\dot\rho = \frac{\rho_\beta - \rho}{T_1} - \frac{\begin{pmatrix} 0 & \rho_{01} \\ \rho_{10} & 0 \end{pmatrix}}{T_2}$. Usually T_2 is a lot quicker; the environment simply learns whether you are in the 0 or 1 state. Here $\rho_\beta = \frac{e^{-\beta H}}{\operatorname{Tr} e^{-\beta H}}$ is the thermal state.

Now, take $\rho = \frac{I + \vec{a}\cdot\vec{\sigma}}{2}$. Then $\sigma_3 \rho \sigma_3 = \frac{I + (-a_1, -a_2, a_3)\cdot\sigma}{2}$, which looks like a 180 degree rotation about the z axis.

We have $\rho \to \frac{\rho + \sigma_3 \rho \sigma_3}{2} = \frac{I + a_3 \sigma_3}{2}$.

37 No Section Title

Density matrices are a way of modeling noise. One type of noise is called *dephasing*, which can either be a $|0\rangle, |1\rangle$ measurement, or a random $\{I, Z\}$ gate. There are other ways of of modeling dephasing, such as I 90% of the time and Z 10% of the time, but this is a simple model.

Supose we use a repetition code, with $a\,|000000\rangle + b\,|111111\rangle$, which leaves the state more vulnerable to dephasing as the number of repetitions increases, since dephasing any one of the repeated qubits dephases the entire state. In practice, this state has the density matrix $a\,|000000\rangle\,\langle 000000| + b\,|111111\rangle\,\langle 111111|$. On classical computers, it's common to use codes similar to repetition codes that are good for X errors but more vulnerable to Z errors or dephasing noise.

37.1 Depolarizing Noise

Consider a 1/4 probability of I, X, Y, or Z. Starting with $\rho = \frac{I + \vec{a}\cdot\vec{\sigma}}{2}$ maps to $\frac{I\rho I + X\rho X + Y\rho Y + Z\rho Z}{4}$. Using the Bloch ball perspective, recall that Z left the identity part of ρ alone, and the Z part alone, while flipping the X and Y components. Similarly, X leaves the I and X parts alone and flips Y and Z.

In total, this becomes $\frac{I}{2} + \frac{1}{8}(\vec{a} + (a_1, -a_2, -a_3) + (-a_1, a_2, -a_3) + (-a_1, -a_2, a_3)) \cdot \sigma = \frac{I}{2}$. Equivalently, we could apply I or Z randomly, then I or X randomly, which takes $\rho \mapsto \frac{I + a_3\sigma_3}{2} = \frac{1+a_3}{2}\,|0\rangle\,\langle 0| + \frac{1-a_3}{2}\,|1\rangle\,\langle 1|$, which is maximally mixed.

When we talked about the Lo-Chau protocol, we said to apply X or Z randomly to each of the qubits. Then, Eve will see only a maximally mixed state, while Bob will be able to decode with no problem as the sequence of X and Z are revealed later. Here it is $X^s Z^t\,\mathrm{CSS}(C_1 : C_2)$.

Another place where density matrices can come from is from entangled states. For example, half of a Bell state looks maximally mixed. A feature of quantum mechanics is that a state can be globally pure but locally mixed.