CS 231 Quantum Computation and Quantum Complexity[1]

Scribe Notes

Anurag Anshu

August 16, 2022

[1]This is the collection of scribe notes for CS 231 Quantum Computation and Quantum Complexity taught at Harvard in Spring 2022 by Anurag Anshu. A significant part of the course material is adapted from a quantum computing course co-taught with Umesh Vazirani at University of California, Berkeley in Fall 2021.

Contents

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 1

January 26, 2022 *Scribe: Benji Kan*

1.1 The Quantum Formalism

Welcome to CS 231! Anurag Anshu is the instructor for this course, and the teaching fellows are Chi-Ning Chou and Prayaag Venkat.

Course Logistics

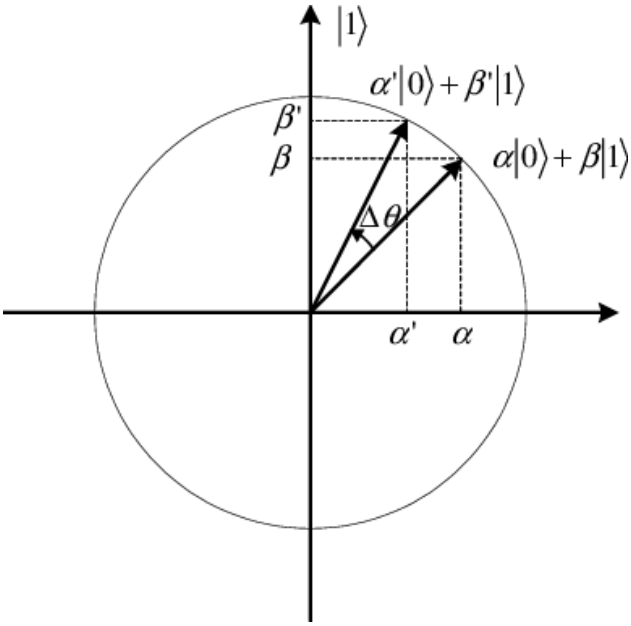
- **Project topics**. There will be two options for final projects: a) come up with a quantum analog to a classical idea in CS or b) write a summary report on some interesting topics in quantum computing. Projects are tentatively individual but may be completed in pairs for more advanced subjects.
- **Office Hours**. Anurag's office hours are Wednesdays 11:30-12:30, and the TFs will send out a survey to decide times on Monday or Tuesday.
- **Homeworks**. Homeworks are due at 11:59pm on Tuesdays, and the highest $n-3$ out of all n assignments ($n \approx 12 \pm 1$) will be counted for the final grade. Moreover, each student has a budget of 10 late days, of which a maximum of 2 late days may be used on any one assignment.
- Homework 0 should be submitted on Gradescope by Thursday night, and no late days may be used for this.
- Ed will be the main platform for announcements.

Today, we will discuss quantum theory, multiple quantum systems.

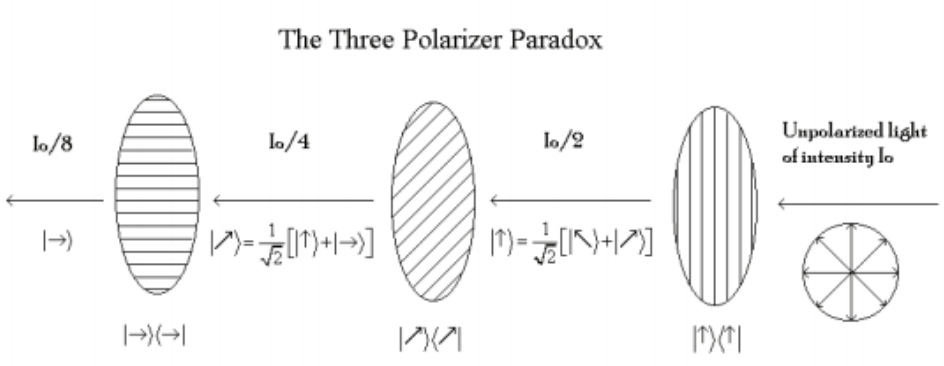
1.1.1 Quantum Theory

We begin with a meme about the double-slit experiment, and see a demonstration of interference with polarizing filters.

A photon can be viewed as a **qubit** and can be represented as a vector in the 2D complex vector space with axes $|0\rangle, |1\rangle$.



The first polarizer is oriented to measure photons in the $\{|0\rangle, |1\rangle\}$ basis and blocks photons in the $|1\rangle$ state, only allowing photons in the $|0\rangle$ state to pass through. On the other hand, the third polarizer blocks photons in the $|0\rangle$ state, only allowing photons in the $|1\rangle$ state to pass through. The second polarizer is rotated 45° relative to the other two, measuring in the $\{|+\rangle, |-\rangle\}$ basis where $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. So, without the second polarizer, no light passes through the two-polarizer system. However, inserting the second polarizer between the other two projects the state of intermediate photons onto $|+\rangle$, which allows some light to pass through the third polarizer.



From this demo, we conclude that **measurement changes the state**.

Components of quantum theory

We describe some essential components of quantum formalism for quantum computation.

- *Quantum states*: positive semi-definite matrix with trace one (also called **density matrix**).

  Rank one matrices are **pure** quantum states (meaning complete knowledge of system). We can write any state in a density matrix form as

  $$\rho = p_1 |\psi_1\rangle \langle\psi_1| + p_2 |\psi_2\rangle \langle\psi_2| + \dots \tag{1.1}$$

  PSD means that $p_1, p_2, \cdots \geq 0$, and trace one ensures $p_1 + p_2 + \cdots = 1$.
  A rank one matrix occurs when $p_1 = 1, p_2 = p_3 = \cdots = 0$. Otherwise if the matrix is not rank one, it is in a **mixed state.** The maximally mixed state for a qubit occurs when the density matrix is $\frac{1}{2} |0\rangle \langle 0| + \frac{1}{2} |1\rangle \langle 1| = \frac{I}{2}$.

- *Unitary transformation*: a probability preserving, reversible, transformation (e.g. in a quantum circuit).

  These can essentially be viewed as a change of basis. An important example is the **Hadamard transform** $\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$, which takes $\{|0\rangle, |1\rangle\} \to \{|+\rangle, |-\rangle\}$, as in the second polarizer in our earlier demo.

- *Measurement*: observation on the system in a chosen orthonormal basis.

  From a quantum information perspective, measurements change the description of the system to gain information about it. For example, the second polarizer in the demo takes $|0\rangle \langle 0| \to \frac{1}{2} |+\rangle \langle +| + \frac{1}{2} |-\rangle \langle -|$. Thus, the act of measuring also provides an outcome.

1.1.2 Multiple quantum systems

To describe a system of two qubits, we set up the joint vector space $\mathcal{H}_A \otimes \mathcal{H}_B$, which is a **tensor product** of the individual vector spaces of each atom. (Following classical notation, we have used **registers** for the 1st (A) and 2nd (B) qubits to avoid ambiguity.) So, if the initial basis vectors of each atom are $\{|0\rangle, |1\rangle\}$, then one of the four new basis vectors for the joint space is

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}_A \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}_B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{1.2}$$

the other three joint basis vectors are defined in the natural way.

The allowed states include pure states such as $|0\rangle_A \otimes |0\rangle_B, |0\rangle_A \otimes |1\rangle_B, |1\rangle_A \otimes |0\rangle_B, |1\rangle_A \otimes |1\rangle_B$. They also include non-pure entangled superpositions of these states such as $\frac{1}{\sqrt{2}} |0\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \otimes |1\rangle$. We will see that this entanglement is a source of quantum advantage.

CNOT gate

We consider a simple quantum circuit operation. Let U be the unitary for a CNOT (controlled NOT) gate. The CNOT gate is a two qubit gate that acts as follows:

$$U|0\rangle|0\rangle = |0\rangle|0\rangle$$
$$U|0\rangle|1\rangle = |0\rangle|1\rangle$$
$$U|1\rangle|0\rangle = |1\rangle|1\rangle$$
$$U|1\rangle|1\rangle = |1\rangle|0\rangle$$

We observe that applying the CNOT to the $|+\rangle_A|0\rangle_B$ state *generates entanglement*:

$$U|+\rangle_A|0\rangle_B = U\left(\frac{1}{\sqrt{2}}|0\rangle_A + \frac{1}{\sqrt{2}}|1\rangle_A\right)|0\rangle_B = U\frac{1}{\sqrt{2}}|0\rangle_A|0\rangle_B + U\frac{1}{\sqrt{2}}|1\rangle_A|0\rangle_B = \frac{1}{\sqrt{2}}|0\rangle_A|0\rangle_B + \frac{1}{\sqrt{2}}|1\rangle_A|1\rangle_B$$

We note that **amplitudes** are related to probabilities, but may be negative or even complex.

1.1.3 Looking ahead

- Some topics we will cover in the next few weeks include: quantum gates, more entanglement, quantum algorithms, quantum NP, non-locality, cryptography
- More importantly, we will learn about some far-reaching phenomena in the field.
  - **Static vs dynamic**: favorable translations between dynamic processes and static objects. In this class, we will see examples such as the Quantum Cook-Levin theorem, Hamiltonian simulation. Classical analogues include the Cook-Levin Theorem—which translates a classical circuit (a dynamic object) to the constraint satisfaction problem (a static object)—and approximating permanents with random walks.
  - **Discrete vs continuous**: the tension between classically natural discrete phenomena and the inherently continuous components of quantum theory. In this class, we will discuss quantum gates, Pauli matrices, Jordan's lemma. Jordan's lemma will lead to quantum walks, quantum signal processing, tests of entanglement, and rewinding in cryptography.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 2

January 28, 2022 *Scribe: Cole French*

2.1 Announcements

2.1.1 Office hours

All office hours will be in **SEC 3.317**. See Ed for updates post-publication.

Anurag
Wednesdays 11:30am–12:30pm

Chi-Ning
Mondays 4–5pm
Fridays 11:30am–12pm

Prayaag
Tuesdays 4–5pm

For Chi-Ning's first Monday office hours on January 31, he will lead a special section covering math and physics background for the course. See Ed for details.

2.1.2 Homework 1

Homework 1 will be released on February 2 and due on February 8.

2.2 Roadmap

Today, our primary objective is to understand *qubits*.

- How do we define a qubit?
- What are gate sets?
- How is entanglement useful?
- What are Pauli operators?

At the end, we will review Homework 0.

2.3 Defining a qubit

> A qubit ... is the basic unit of quantum information.
>
> Wikipedia

As all good monologues do, we begin with a vague quote from a familiar online source. We can ask ourselves, Is this a good definition? The answer is, No, because we have simply substituted one unknown term, "qubit," for another, "quantum information."

A good definition is *useful* and makes sense in a *realistic* situation. In our quest for a good definition, we will use an adversarial framework: Two people, call them Anurag and Boaz, meet up for coffee. Boaz remarks that he has come into possession of a qubit. Of course, Anurag believes him, but just to be sure: **What questions can Anurag ask to convince himself Boaz' mystery object is actually a qubit?**

2.3.1 First candidate definition

A qubit is a 2D complex vector space.

If this is a qubit, then Anurag would like proof that Boaz has such a vector space. However, he cannot request the entire vector space, because it contains infinitely many vectors. Therefore, Anurag must request some finite subset of vectors $\{|\psi_n\rangle\}$. But Boaz could just have these lying around in his secret warehouse! In addition to possessing a physical object, Boaz must evidence some "quantum power"—an ability to manipulate the objects quantum mechanically.

2.3.2 Second candidate definition

A qubit is a 2D complex vector space *and* a set of unitaries acting on the vectors in the space.

Now, Anurag would like to verify that Boaz can manipulate quantum state. But which unitaries should he ask for? Note that there are infinitely many of them; any unitary U satisfies

$$U^\dagger U = \mathbb{1}$$
$$\det U = 1,$$

from which we can determine that

$$U = \begin{pmatrix} a+bi & c+di \\ -c+di & a-bi \end{pmatrix}$$

for some $a, b, c, d \in \mathbb{R}$. Thus, any vector in $\mathbb{R}^4$ describes a unitary transformation.

Fortunately, Anurag does not need every unitary; he can ask Boaz for a *gate set*, a collection of gates that can be applied in some sequence to approximate any unitary. We will consider the gate set $\{H, T\}$ that consists of the Hadamard gate introduced last lecture and the T gate.

Approximation

Before we discuss the Hadamard and T gates specifically, we should clarify what it means to approximate a unitary U. We can quantify such an approximation using a *norm*.

For example, the *Frobenius norm* is a norm that can be thought of as an extension of the L^2 norm to matrices. While the L^2 norm measures the sum of the squares of the components of a vector, the Frobenius norm measures the sum of the squares of the diagonal entries of a matrix. Thus, for some matrix M, it is given by

$$\|M\|_F = \sqrt{\mathrm{Tr}(M^\dagger M)}.$$

This means we can measure the Euclidean distance between two matrices just as we do between two vectors: For some approximating sequence of gates $HTHT\ldots$, our approximation error is $\|U - HTHT\ldots\|_F$.

Note that the Frobenius norm is just one quantification of approximation. It is usually convenient, but nothing stops us from choosing other norms.

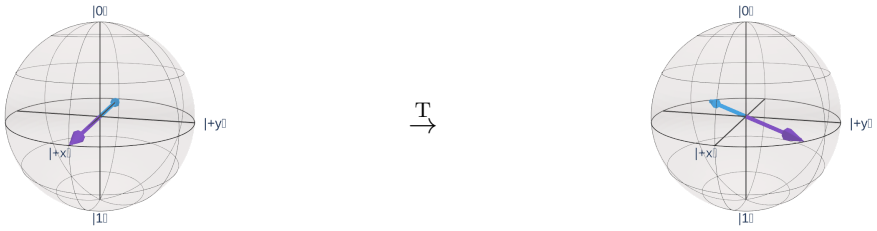
Now, let us return to our gate set:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$
$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

Recall that the Hadamard gate takes $\{|0\rangle, |1\rangle\}$ to $\{|+\rangle, |-\rangle\}$, creating a quantum state that has an equal chance of being measured as either $|0\rangle$ or $|1\rangle$ in the standard basis. We can visualize this on the *Bloch sphere*, which we describe later in these notes.

Notice how the resulting states are perpendicular to the Z-axis; this exhibits the equal probability of being measured as $|0\rangle$ or $|1\rangle$.

Meanwhile, the T gate rotates a quantum state by $\pi/4$ radians around the Z-axis of the Bloch sphere.



Back to the task at hand, how can Anurag confirm Boaz has a gate set? One method is that he can choose one of two quantum states, $|0\rangle$ or $|+\rangle$, with equal probability, and ask Boaz to transform it with the Hadamard gate. $H|0\rangle = |+\rangle$ and $H|+\rangle = |0\rangle$. Note that they would have to repeat this interaction a number of times to improve Anurag's certainty that Boaz has a Hadamard gate. For example, with just one interaction, Boaz could return either of the two quantum states arbitrarily for a $1/2$ chance of guessing correctly.

OK, so this seems like a good definition of a qubit. Are there any others?

2.3.3 Third candidate definition

> A qubit is a 2D complex vector space and the *Pauli* X and Z operators.

Compared to the previous definition, we have substituted our two matrices forming a gate set, H and T, for two new matrices, X and Z. Therefore, we hope X and Z exhibit some kind of "universality," so that they are just as useful.

In fact, they are even more useful. The Pauli matrices are

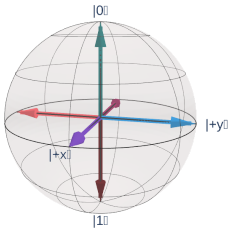
$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$
$$Y = iXZ = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

and they have some nice properties: They...

- ...are Hermitian and unitary, each with two eigenvalues, 1 and -1. The corresponding normalized eigenvectors are

	$\lambda = 1$	$\lambda = -1$
X	$\frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$	$\frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right)$
Z	$\left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$	$\left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$
Y	$\frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 \\ i \end{smallmatrix}\right)$	$\frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 \\ -i \end{smallmatrix}\right)$,

  and each of them forms half of an axis of the Bloch sphere.



- ...anticommute, so $XZ = -ZX$, for example.
- ...can be put into a linear combination with one another and the identity matrix to equal any (2D) matrix.

This last property includes any quantum state and any unitary, so this is a great definition of a qubit. For example, here is our gate set written as linear combinations of the Pauli matrices:

$$H = \frac{1}{\sqrt{2}}(X + Z)$$
$$T = \frac{1}{2}\left(1 - \sqrt{i}\right)Z + \frac{1}{2}\left(1 + \sqrt{i}\right)\mathbb{1}.$$

Uncertainty principle

We can use the Pauli matrices to demonstrate a fundamental property of quantum mechanical systems, the *uncertainty principle*, which limits the accuracy of measurements of a system. For motivation, consider an ensemble of particles that we prepare in state $|\Psi\rangle$. Suppose our preparation process is so consistent, that each time we prepare a particle and measure its position, we get nearly identical results. This means the variance in the position,

$$\sigma_x^2 = \left\langle x^2 \right\rangle - \langle x \rangle^2 ,$$

is incredibly small.

However, Heisenberg's uncertainty principle asserts that

$$\sigma_x \sigma_p \geq \frac{\hbar}{2},$$

where $\hbar$ is a constant.[1] That is, as the particle's variance in position vanishes, its variance in *momentum* grows larger. This means there is some inherent, physical limit on how accurately we can prepare the particle's position and momentum!

Heisenberg's uncertainty principle is one instance of the general uncertainty principle for two operators A, B:

$$\sigma_A \sigma_B \geq \frac{1}{2i} \langle [A, B] \rangle ,$$

[1] The reduced Planck constant

where $[A, B]$ is called the *commutator* and is simply defined as $AB - BA$. (It is called this, because it equals 0 exactly when A and B commute.) Now, remember that one of the properties of the Pauli matrices is that they *anti*commute;

$$\begin{aligned}[X, Z] &= XZ - ZX \\ &= 2XZ = -2ZX.\end{aligned}$$

In particular, this value is nonzero, so there is an uncertainty principle for the Pauli matrices that you (yes, you!) can compute with the above formula.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 3

August 16, 2022 *Scribe: Seung Hwan An*

3.1 Quantum key Distirbution

Quantum Key Distribution : Alice and Bob want to share a secret shared key (a random string). The key is used to perform one time pad. Eve wants to learn the key (or any information regarding it). Alice and Bob are allowed to abort the protocol if they find that Eve has tampered with the message. Conditioned on no-abort, a secret key must be shared.

Suppose the shared quantum secret bit that Alice and Bob wants to share is

$$\frac{1}{2}|0\rangle\langle 0|_A \otimes |0\rangle\langle 0|_B + \frac{1}{2}|1\rangle\langle 1|_A \otimes |1\rangle\langle 1|_B$$

This is the same as sharing a secret bit that has probability 1/2 of being 0 and 1/2 of being 1.

No cloning theorem Alice and Bob can't just share the secret key using classical realm, due to copying attack by Eve, who can copy each message from Alice and essentially simulate Bob. In the quantum world, Alice and Bob can, however, share the secret key, due to:

Theorem 3.1.1 (No Cloning Theorem)**.** *There is no unitary/valid quantum transformation U such that*

$$U|0\rangle_A \otimes |0\rangle_B = |0\rangle_A \otimes |0\rangle_B \qquad U|+\rangle_A \otimes |0\rangle_B = U|+\rangle_A \otimes |+\rangle_B$$

Proof. Unitary transformation conserves inner product, and are linear. If such U existed, then

$$1/\sqrt{2} = \langle 00|+0\rangle = \langle 00|U^\dagger U|+0\rangle = \langle 00|++\rangle = 1/2$$

a contradiction. □

BB '84 Quantum Key Distribution Protocol This protocol was introduced by Charles Bennett and Gilles Brassard [BB84]. This protocol assumes a public channel over which Alice and Bob can announce classical messages. The following steps are run in parallel n times.

Step 1. Alice generates 2 random bits (b, a), where b is the basis and a is the state.

Step 2. Alice sends to Bob the following state in a register M according to this table:

$b\backslash a$	0	1
0	$\|\psi_{00}\rangle = \|0\rangle$	$\|\psi_{10}\rangle = \|1\rangle$
1	$\|\psi_{01}\rangle = \|+\rangle$	$\|\psi_{11}\rangle = \|-\rangle$

The current quantum state in the protocol is represented as:

$$\frac{1}{4}\sum_{a,b=0}^{1} |a\rangle\langle a| \otimes |b\rangle\langle b| \otimes |\psi_{ab}\rangle\langle\psi_{ab}|$$

To get the state that Eve observes, take the partial trace to get:

$$\begin{aligned}\frac{1}{4}\sum_{a,b=0}^{1} |\psi_{ab}\rangle\langle\psi_{ab}| &= \frac{1}{4}\big(\,|0\rangle\langle 0| + |1\rangle\langle 1| + |+\rangle\langle +| + |-\rangle\langle -|\,\big) \\ &= \frac{1}{2}\big(\,|0\rangle\langle 0| + |1\rangle\langle 1|\,\big) = \frac{\mathbb{1}_M}{2}.\end{aligned}$$

Step 3. Bob generates another random bit: b', and then measures M in:

- $\{|0\rangle, |1\rangle\}$ basis if $b' = 0$.
- $\{|+\rangle, |-\rangle\}$ basis if $b' = 1$.

If $b = b'$, then Bob's measurement outcome would be a. If $b \neq b'$, then Bob's measurement outcome would be a random bit. Denote the result of measurement as a'.

Step 4. Alice and Bob announce b and b'.

- If $b = b'$, then we are done.
- If $b \neq b'$, Bob rejects the key.

Step 5. Alice takes a small random subset $S \subseteq [n]$ and announce it to Bob. They both check that $a_i = a'_i$ for all $i \in S$ by announcing these bits.

Analysis: First, suppose that Eve has not interfered with the communication between Alice and Bob. Then, Alice and Bob shares n keys, each of which with probability $1/2$ is accepted or rejected. For the keys that are accepted *and* unannounced at Step 5, Alice has now successfully shared the secret key a_i's to Bob.

Secondly, assume for now the adversary Eve applies some unknown non-identity unitary transformation U on each message qubit. Assume $a = 0$. Then, if $b = b' = 0$, Bob would measure $U|0\rangle$ in $\{|0\rangle, |1\rangle\}$ basis, and therefore would get $a' = 1$ with probability $|\langle 1|U|0\rangle|^2$, which is non-zero. If $b = b' = 1$, Bob would measure $U|+\rangle$ in $\{|+\rangle, |-\rangle\}$ basis, and therefore would get $a' = 1$ with probability $|\langle -|U|+\rangle|^2$, which is also non-zero. Therefore, for every $i \in S$, there is a non-zero probability p that $a_i \neq a'_i$. Since the unitary transformation is done globally, while the random bits chosen by Bob and Alice are independent of each other, Bob is able to detect the presence of Eve with probability $p^{|S|}$.

In the case where Eve applies an arbitrary operation, instead of a unitary transformation on each qubit, a more involved analysis involving quantum De Finetti's theorem is used to reduce this to the above case.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 4

August 16, 2022 *Scribe: Matthew Nazari*

4.1 Recap

Let us clarify the difference between pure and mixed states. Say that there are some photons in a box. We know beforehand each photon is in the state

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

If we draw a photon $|\psi\rangle$ from this box and measure in the $\{|0\rangle, |1\rangle\}$ basis, we will observe the state $|0\rangle$ with $|\langle 0|\psi\rangle|^2 = 50\%$ probability and the state $|1\rangle$ with $|\langle 1|\psi\rangle|^2 = 50\%$ probability. This is because we know for certain $|\psi\rangle = |+\rangle$ and thus in a maximally entangled superposition between the states $|0\rangle$ and $|1\rangle$.

Compare this to a different situation where 1/3 of the photons in that box are in the state $|-\rangle$ and 2/3 are in the state $|+\rangle$. Suddenly, according to classical probability, we can no longer be certain of the state $|\psi\rangle$ we draw from the box. The state $|\psi\rangle$ is a random variable in the classic sense, which is a *mixed state* in the quantum sense.

Mixed state: A (classically) probabalistic ensemble of pure states $\{|\psi_i\rangle\}$, commonly represented in density matrix notation by a convex combination the pure states:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \text{ where } \sum_i p_i = 1.$$

This representation makes it convinient to calculate probabilities. Consider a box $\{(|\psi_i\rangle, p_i)\}$ where p_i is the proportion or probability of drawing the state $|\psi_i\rangle$ (each state belonding to the D-dimensional Hilbert space with basis $\{|1\rangle, \ldots, |D\rangle\}$). We represent a state drawn from this box with the density matrix ρ. The probability of observing, say, $|3\rangle$ is then

$$\sum_i p_i |\langle\psi_i|3\rangle|^2 = \langle 3|\rho|3\rangle = \mathrm{Tr}(|3\rangle\langle 3|\,\rho).$$

This is analogous to taking an expectation from classical probability theory.

What if, instead, we are interested in the probability of not just state $|3\rangle$, but one of the states $|1\rangle$, $|2\rangle$, and $|3\rangle$. This probability is given by

$$\sum_{i=1}^{3} \mathrm{Tr}(|i\rangle\langle i|\,\rho) = \mathrm{Tr}\left(\sum_{i=1}^{3} |i\rangle\langle i|\,\rho\right).$$

Here, $\sum_{i=1}^{3} |i\rangle\langle i|$ is the sum of three rank 1 matrices. The value $\mathrm{Tr}(\Pi\rho)$ calculates the probability of measuring the outcome that we encoded into Π. We call the operator Π a *projector* because it projects ρ onto the subspace spanned by $\{|1\rangle, |2\rangle, |3\rangle\}$. This kind of measurement is called a *projective measurement*.

Projective measurement: A set of *measurement operators* $\{\Pi_i\}$ such satisfy the completeness equation

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

To calculate the probability of measuring an outcome i from a mixed state ρ, you simply calculate

$$\mathrm{Tr}(\Pi_i \rho).$$

Note that we can assign real values m_i to each outcome Π_i. This object $M = \sum_i m_i \Pi_i$ is a random variable in the classical sense, and an *observable* in the quantum sense. The "expected value" of the observable M is simply $\mathrm{Tr}(M\rho)$.

4.2 Quantum Key Distribution

Suppose Alice and Bob want to share a joint random bit that nobody else knows. The goal, mathematically, is to share the state

$$\frac{1}{2} |0\rangle \langle 0|_A \otimes |0\rangle \langle 0|_B + \frac{1}{2} |1\rangle \langle 1|_A \otimes |1\rangle \langle 1|_B .$$

This is quantum notation for a perfectly correlated random bit shared between Bob and Alice (two individual bits, but both sharing the same bit value 0 or 1). Consider the following protocol that achieves this goal:

Protocol 1

1. Alice prepares two qubits

$$\frac{1}{\sqrt{2}} |0\rangle_A \otimes |0\rangle_B + \frac{1}{\sqrt{2}} |1\rangle_A \otimes |1\rangle_B .$$

2. Alice sends the second qubit B to Bob.

3. Alice and Bob both measure in a basis that they agree on beforehand, like $\{|0\rangle, |1\rangle\}$.

The object Alice prepared is known as an *EPR pair*.

EPR pair: Named after Einstein, Podolsky and Rosen, an EPR pair refers to maximally entangled quantum states of two cubits between the $|0\rangle$ and $|1\rangle$ basis states:

$$\frac{1}{\sqrt{2}} |0\rangle_A \otimes |0\rangle_B + \frac{1}{\sqrt{2}} |1\rangle_A \otimes |1\rangle_B , \quad \frac{1}{\sqrt{2}} |0\rangle_A \otimes |0\rangle_B - \frac{1}{\sqrt{2}} |1\rangle_A \otimes |1\rangle_B ,$$

$$\frac{1}{\sqrt{2}} |0\rangle_A \otimes |1\rangle_B + \frac{1}{\sqrt{2}} |1\rangle_A \otimes |0\rangle_B , \quad \frac{1}{\sqrt{2}} |0\rangle_A \otimes |1\rangle_B - \frac{1}{\sqrt{2}} |1\rangle_A \otimes |0\rangle_B .$$

While yes this protocol would work, it has a major flaw: consider if there was a malicious agent, call her Eve, attempting to intercept communication between Alice and Bob. Obviously, Alice and Bob want to keep their bit secret from the world. Hence, they need to be able to abort if Eve has tampered in any way with their communications. Hence, we use a different protocol first proposed in [Eke91] :

Protocol 2

1. Alice prepares n EPR pairs to Bob.
2. Alice sends the second qubits $B_1, \ldots, B_n$ to Bob.
3. Bob takes $n/2$ random pairs and performs some test called the CHSH test. If this test fails, Eve may have intercepted their communication and the protocol is aborted. Otherwise, continue to the next step.
4. Measure all remaining $n/2$ pairs in a basis agreed on beforehand.

4.3 CHSH Test

Imagine a game where Alice and Bob each recieve a random bit x and y such that

$$x, y \sim \text{Unif}(\{0, 1\}).$$

Alice knows x but Bob does not (and vice versa with y). They cannot communicate with eachother, which means neither can tell the other what bit they recieved. Alice and Bob must independently decide on bits a and b to output. They win if they output bits such that

$$x \oplus y = a \wedge b.$$

What strategy will maximize their probability of success? Classically, it turns out that no strategy can do better than probability $\frac{3}{4}$. This is true even if Alice and Bob share a source of randomness. If Alice and Bob are permitted to share quantum entanglement, however, we can do better with chance $\cos^2(\pi/8) \approx 0.85$.

The CHSH test, therefore, is to randomly select the $n/2$ EPR pairs and see if we win the game above about ≈ 0.85 of the time. If it is less, then Eve may have tampered and we abort.

We now detail what happens in the quantum case. Let's say Alice and Bob share the state $|\psi\rangle_{AB}$. This does not necessarily have to be an EPR pair. If Alice recieves an x, she will perform the projective measurement $\{P_x^0, P_x^1\}$. Similarly, Bob recieves y performs the measurement $\{Q_y^0, Q_y^1\}$. (Note $P_x^1 = I - P_x^0$ and $Q_y^1 = I - Q_y^0$). If Alice measures P_x^0, she sets $a = 0$. If she measures P_x^1, she sets $a = 1$. Bob does this similarly with b.

What, then, is the probability that we win the game? That is, what is the probability of $x \oplus y = a \wedge b$? We calculate it by

$$\frac{1}{4} \sum_{x,y=0}^{1} \sum_{a,b\,:\,x\oplus y = a\wedge b} \langle\psi| P_x^a \otimes Q_y^b |\psi\rangle = \frac{1}{4} \operatorname{Tr}\Big(|\psi\rangle\langle\psi| P_x^a \otimes Q_y^b\Big).$$

This looks like a projective measurement where $|\psi\rangle\langle\psi|$ is our observable M and $P_x^a \otimes Q_y^b$ is our density matrix ρ.

Let us clean this using observables

$$A_x = P_x^0 - P_x^1, \quad B_y = Q_y^0 - Q_y^1.$$

An example of an obervable A_x is the Pauli operators $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ and $X = |+\rangle\langle +| - |-\rangle\langle -|$. Notice that

$$P_x^a = \frac{1}{2}(I + (-1)^a A_x), \quad Q_y^b = \frac{1}{2}(I + (-1)^b B_y).$$

The probability, therefore, can be rewritten as

$$\begin{aligned}
&\frac{1}{4}\sum_{x,y=0}^{1}\sum_{a,b\,:\,x\oplus y=a\wedge b}\langle\psi|\tfrac{1}{4}(I+(-1)^a A_x)(I+(-1)^b B_y)|\psi\rangle \\
&=\frac{1}{16}\left(8+2\sum_{x,y=0}^{1}(-1)^{x\wedge y}\,\langle\psi|\tfrac{1}{4}A_x\otimes B_y|\psi\rangle\right).
\end{aligned}$$

Notice we are able to write the above because all terms that solely depend on a or b are cancelled due to the alternative signs caused by the $(-1)^a$ and $(-1)^b$ factors.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 5: BQP and Grover Search

February 9, 2022 *Scribes: Isaac Struhl and Ben Harpe*

5.1 Outline

Today's lecture notes encompass the following (but some might spill over to next class):

- Quantum circuits
- BQP
- Search problems
- BQP vs NP
- The "polynomial method", a lower bound on quantum search

5.2 Models of Quantum Computation

In quantum computing there are 3 models that are extremely popular:

- **Quantum circuit model** (today's focus)
- Measurement-based model (potentially good for a project in this class!)
- Adiabatic model (don't worry about, but good to know exists)

5.2.1 Quantum circuit model

Say you have a set of n qubits. They together form a vector space of dimension 2^n, and there are many unitaries that one can apply on this set of qubits. A unitary that applies to these qubits will have dimension $2^n \times 2^n$. **The quantum circuit model says you can approximate any unitary using a sequence of elementary gates.** [1] This is analogous to the classical computational model where NAND can generate any transformation. Let's define the 3 gates:

First, the Hadamard:

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

second, the T:

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

[1] This is known as the Solovay-Kitaev theorem.

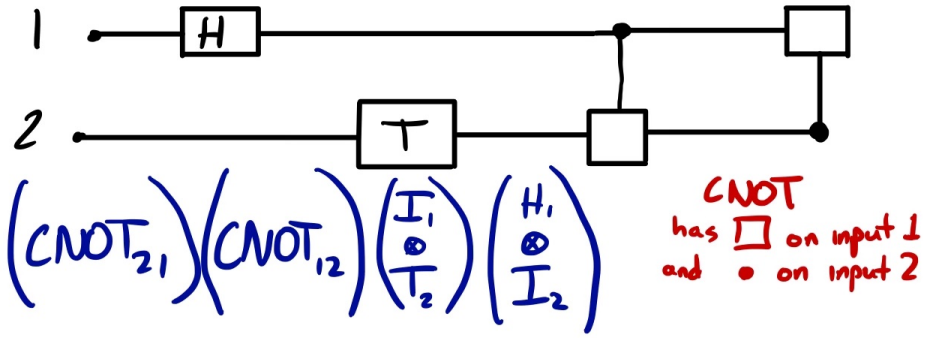
and third, CNOT:

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

The CNOT gate directly works on two qubits, conditionally flipping the second bit. The other two gates are used on a single qubit, but can be used in the form $\mathbb{1} \otimes H$ to preserve the first qubit and generate the result on a second register. As an example, the reversible XOR is given by CNOT:

$$CNOT(|a\rangle \otimes |b\rangle) = |a\rangle \otimes |a \oplus b\rangle$$

A **quantum circuit** is just a collection of these gates applied to registers in sequence. Note that any gate can include the identity $\mathbb{1}$ to preserve the value of any register. The following diagram gives an example of a simple quantum circuit.



In the above drawing, note that the circuit representation moves left to right but the unitary representation moves right to left. Additionally, when drawing a CNOT gate, a box represents the first input and a dot represents the second.

5.2.2 Performing a computation

How do we perform a computation? In general, a computation just applies a unitary on a collection of qubits. So, we can generate this computation with a circuit, because we can use combinations of H, T, and CNOT to approximate any unitary. As with classical computers, the larger the circuit required, the more expensive the computation. Say we're given the following function we want to compute:

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

To perform this computation, we have the following requirements:

- Q_n needs to have poly(n) gates
- All of the gates must be specified by a classical computer - "uniformity". This essentially means is that the quantum circuit should not require you to solve an unsolvable problem (e.g. the halting problem)

Let x be an input to the function, so

$$|x\rangle \equiv |x_1\rangle \otimes |x_2\rangle \otimes \ldots \otimes |x_n\rangle$$

is the n-dimensional qubit input. We can also add a (potentially large, but polynomial in n) set of empty registers ("work resistors", also known as "ancilla qubits.")

$$a(n) = |0\rangle \otimes |0\rangle \otimes \ldots \otimes |0\rangle$$

to help with the computation (though we will see in a later lecture that this isn't strictly necessary). We then apply a quantum circuit to this set of registers.

5.2.3 Interpreting the output

The goal of any computation is to eventually get a result, which requires *measurement*. We measure in the **computational basis**, $\{|0\rangle, |1\rangle\}$, interpreting the output as a classical result.

We can measure on one qubit, but what does it look like on the full space? This looks like

$$\{\Pi_0 = |0\rangle\langle 0|_0 \otimes \mathbb{1}, \quad \Pi_1 = |1\rangle\langle 1|_0 \otimes \mathbb{1}\}$$

i.e., we measure on one qubit, and apply the identity (do nothing) to the rest. Now let's check correctness. We have

$$|\Psi_x\rangle = Q_n |x\rangle |0\rangle^{a(n)}$$

If the result of the measurement is $\Pi_{f(x)}$, then we're right. Thus, the probability of getting the correct answer is

$$\boxed{\mathrm{Tr}\{\Pi_{f(x)} \cdot |\Psi_x\rangle\langle\Psi_x|\} = \langle\Psi_x| \Pi_{f(x)} |\Psi_x\rangle}$$

In general, we want to have bounded error, e.g. correctness with probability $\geq \frac{2}{3}$ for all inputs x (though $\frac{2}{3}$ is an arbitrary choice).

Summary

This is BQP. BQP is a complexity class, which is a family of boolean functions. Intuitively, it is the set of computational problems that can be efficiently solved by a quantum computer. The classical analog is BPP, *probabilistic* instead of *quantum*. Any classical algorithm can be generated by the quantum algorithm; that is, **BPP is inside BQP**. A formal definition is in Lecture 5 notes on Perusall.

5.3 What can we do with quantum computation?

This and next lecture will try to convince us that quantum computers *cannot* solve NP-hard problems. We'll look at the **search problem**, and then next lecture we'll make the connection to classical computational hardness.

5.3.1 Search problem

A search problem is a problem of searching for a solution. If you're given a hard problem like 3-SAT, there's not much structure, so you can't do much better than searching for a solution.

5.3.2 Oracle model

There is a black box function $f : \{0,1\}^n \to \{0,1\}$. You can't interact with the internals of the function in any way, but you can feed in inputs x and observe outputs $f(x)$. The search problem is as follows:

$$\text{Find } s \in \{0,1\}^n \text{ s.t. } f(s) = 1$$

As we saw in homework 0, classically, we need to ask $O(2^n)$ times. In the hardest case, there is a single s such that $f(s) = 1$ (that is, there is only one string that the function accepts, and we need to find it). **However, in the quantum case, we can do it in $O(\sqrt{2^n})$ due to Grover's search algorithm. Crazy!**

5.3.3 Classical interlude

The classical search algorithm is (without memoization)

(1) Choose a random $x \in \{0,1\}^n$

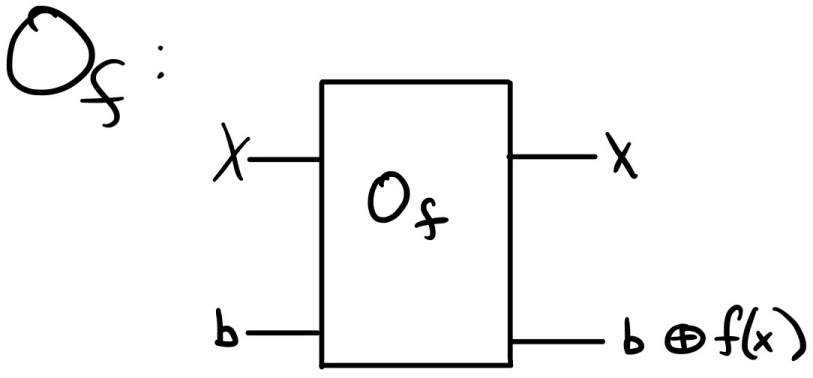
(2) Input x to oracle

(3) If $f(x) = 1$, output x. Else, go to (1).

The quantum version will replace these classical random processes with quantum ones.

5.3.4 Quantum version

Firstly, we need a reversible oracle, because in the quantum world, we can't re-copy states. Fortunately, this is an easy redesign. Considering the following oracle that oracle takes in an input x and a bit b, and outputs x and $b \oplus f(x)$. This way, we can feed in $b \oplus f(x)$ as the second bit with the same input x, and we'll get back b, since $b \oplus f(x) \oplus f(x) = b$. Note that this oracle is different from the oracle defined in 3.2, because in the former, we have no way of getting back to the original bit. See the diagram below.



With a pure state $|b\rangle$,

$$|x\rangle |b\rangle \overset{O_f}{\to} |x\rangle |b \oplus f(x)\rangle$$

but this is just a classical query; we can do better with superposition, for example $|b\rangle = |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$:

$$|x\rangle |-\rangle \overset{O_f}{\to} |x\rangle \frac{|f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}}$$

So, if $f(x) = 0$, nothing has changed, we get out $|-\rangle$. If it is 1, we get a minus sign: $-|-\rangle$. **This is a change in phase; the oracle is called a "phase oracle"**, written as

$$\boxed{\tilde{O}_f}$$

5.3.5 Grover's search algorithm

We now have the tools to define Grover's search algorithm:

(1) Generate a quantum version of randomness, a superposition on n qubits:

$$H|0\rangle \otimes H|0\rangle \otimes \ldots \otimes H|0\rangle \equiv H^{\otimes n}|0\rangle^{\otimes n}$$

Which looks like

$$|+\rangle \otimes |+\rangle \otimes \ldots \otimes |+\rangle$$

For the remainder of the algorithm, we'll refer to this superposition of qubits as a single vector: $|\Psi\rangle$

(2) Query the oracle $\tilde{O}_f$.

(3) Apply the following unitary:

$$U = I - 2|\Psi\rangle\langle\Psi|$$

Note that this is a unitary because $U^\dagger U = 1$ and $U^\dagger = U$. We can also write this as

$$U = I - 2|\Psi\rangle\langle\Psi| = H^{\otimes n}\left(I - 2|0\rangle^{\otimes n}\langle 0|^{\otimes n}\right)H^{\otimes n} = H^{\otimes n}\left(\bar{R}\right)H^{\otimes n}$$

Now $\bar{R}$ looks like a reflection, since

$$\bar{R}|x\rangle = |x\rangle \;\; \forall x \in \{0,1\}^n \backslash \{0^n\}$$

and

$$\bar{R}|0\rangle^{\otimes n} = -|0\rangle^{\otimes n}$$

5.4 Next Class

Next class, we will analyze this protocol, discuss the polynomial method, and discuss the relationship between BQP and NP.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 6

February 11, 2022 *Scribes: Wenjie Gong, Aayush Karan*

A brief notational comment: Although we often write a pure state $|\psi\rangle\langle\psi|$ as $|\psi\rangle$ for notational simplicity, it is important to note that

$$|\psi\rangle\langle\psi| + |\phi\rangle\langle\phi| \neq |\psi\rangle + |\phi\rangle\,.$$

Indeed, in density matrix formalism,

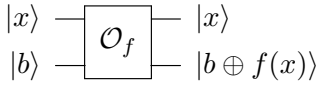
$$\begin{aligned}|\psi\rangle + |\phi\rangle &= (|\psi\rangle + |\phi\rangle)(\langle\psi| + \langle\phi|) \\ &= |\psi\rangle\langle\psi| + |\psi\rangle\langle\phi| + |\phi\rangle\langle\psi| + |\phi\rangle\langle\phi|\end{aligned}$$

Now, we begin with a discussion of Grover's search algorithm.

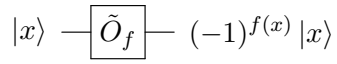
6.1 Grover's Algorithm

Consider the following problem setup.

Goal: Given a function $f : \{0,1\}^n \to \{0,1\}$, $\exists$ a unique $s \in \{0,1\}^n$ such that $f(s) = 1$. For all other $x \in \{0,1\}^n$, $x \neq s$, $f(x) = 0$. We want to find s. Note that as only a single s satisfies $f(s) = 1$, this is the hardest case for both classical and quantum computation.
The algorithm allows interaction with an oracle $\mathcal{O}_f$, where $|x\rangle\,|b\rangle \xrightarrow{\mathcal{O}_f} |x\rangle\,|b \oplus f(x)\rangle$.



As discussed in the pre-lecture notes, the action of $\mathcal{O}_f$ is equivalent to the *phase oracle* $\tilde{\mathcal{O}}_f$, defined as $|x\rangle \xrightarrow{\tilde{\mathcal{O}}_f} (-1)^{f(x)}\,|x\rangle$.



Here, x is an n-bit input, $|x\rangle \equiv |x_1\rangle \otimes |x_1\rangle \ldots |x_n\rangle$.

6.1.1 The Algorithm

Given this oracle $\tilde{\mathcal{O}}_f$, Grover's algorithm carries out the following steps while keeping track of a current quantum state $|\chi\rangle$.

Procedure

Step 1. Initialize the running state $|\chi\rangle$ as

$$H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \equiv |\psi\rangle\,. \tag{6.1}$$

Step 2. Apply $\tilde{\mathcal{O}}_f$ to $|\chi\rangle$. In the computational basis, note that $\tilde{\mathcal{O}}_f$ will add a phase of -1 only to $|x\rangle$ such that the corresponding $x \in \{0,1\}^n$ satisfies $f(x) = 1$; in particular, if and only if $x = s$. Otherwise, $|x\rangle$ is mapped to itself. We may succintly write

$$\tilde{\mathcal{O}}_f = I - 2|s\rangle\langle s|.$$

Step 3. Reflect the current state about $|\psi\rangle$ by applying

$$2\,|\psi\rangle\langle\psi| - I = H^{\otimes n}(2\,|0\rangle\langle 0|^{\otimes n} - I)H^{\otimes n}. \tag{6.2}$$

This adds a phase of -1 to anything orthogonal to $|\psi\rangle$ while leaving $|\psi\rangle$ unchanged. The culmination of Steps 2 and 3 creates a new state $|\chi'\rangle$, so set the running state $|\chi\rangle = |\chi'\rangle\,.$

Step 4. Return to Step 2 and repeat $T \sim O(\sqrt{2^n})$ times.

Step 5. Measure in the computational basis, $\{|x\rangle\,; x \in \{0,1\}^n\}$.

6.1.2 Analysis

Now, let's analyze why this algorithm works. Note that our initial state $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$, which is a linear superposition of all possible n-bit states $|x\rangle$, can be written as:

$$|\psi\rangle = \frac{1}{\sqrt{2^n}}\,|s\rangle + \sqrt{1 - \frac{1}{2^n}}\,|t\rangle \tag{6.3}$$

$$|t\rangle = \frac{1}{\sqrt{2^n - 1}} \sum_{x\neq s} |x\rangle \tag{6.4}$$

Note that $|s\rangle$ and $|t\rangle$ are orthogonal states with magnitude 1, so they are an orthonormal basis for the dimension 2 subspace they span. In general, suppose that $|\chi\rangle = \alpha|s\rangle + \beta|t\rangle$ for some coefficients α and β. Applying $\tilde{\mathcal{O}}_f$ will flip the sign in front of α giving $-\alpha|s\rangle + \beta|t\rangle$. Finally, applying $2|\psi\rangle\langle\psi| - I$ takes this state to $2C|\psi\rangle + \alpha|s\rangle - \beta|t\rangle = \alpha'|s\rangle + \beta'|t\rangle = |\chi'\rangle$, for some constant C. In particular, since $|\chi\rangle$ is initially in the subspace spanned by $|s\rangle$ and $|t\rangle$, it will always remain such under the operations of Grover's algorithm.

To gain some intuition for the geometric process underlying Grover's algorithm, let us follow the initial steps carefully. After Step 2 of the algorithm, the state $|\psi\rangle$ becomes:

$$|\psi\rangle \xrightarrow{\text{Step 2}} -\frac{1}{\sqrt{2^n}}\,|s\rangle + \sqrt{1 - \frac{1}{2^n}}\,|t\rangle$$

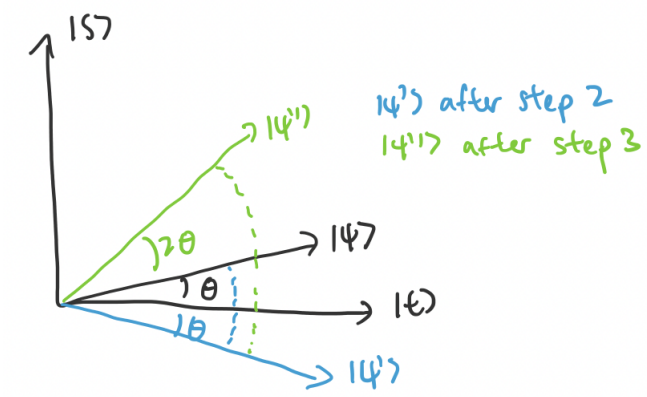


Figure 6.1: A schematic depiction of the quantum state during Grover's algorithm.

Step 3 implements a reflection $2|\psi\rangle\langle\psi| - I$ about the original state $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$, which as mentioned before gives some linear combination of $|s\rangle$ and $|t\rangle$:

$$|\psi\rangle \xrightarrow{\text{Steps 2 \& 3}} \ldots|s\rangle + \ldots|t\rangle \equiv |\chi\rangle$$

We can draw the subspace spanned by $|s\rangle$ and $|t\rangle$, as shown in Fig. 6.1. Initially, our state $|\psi\rangle$ has

$$\sin\theta = \frac{1}{\sqrt{2^n}} \approx \theta$$

where θ is the angle of $|\psi\rangle$ away from $|t\rangle$. Note that since θ is very small, the small angle approximation applies. Step 2 takes $|s\rangle \to -|s\rangle$ (a reflection about $|t\rangle$), so $|\psi\rangle \xrightarrow{\text{Step 2}} |\psi'\rangle$, originally θ above the $|t\rangle$ axis, becomes rotated to θ below the $|t\rangle$ axis. Step 3 negates anything orthogonal to $|\psi\rangle$ (a reflection about $|\psi\rangle$), so $|\psi'\rangle \xrightarrow{\text{Step 3}} |\psi''\rangle$ becomes rotated to 3θ above the $|t\rangle$ axis.

With this in mind, let θ_i denote the angle between vectors $|\chi\rangle$ and $|t\rangle$ at time step i. Now note that application of the oracle reflects $|\chi\rangle$ about the x-axis, so $\theta_i \to -\theta_i$. This reflected vector has angle $-(\theta_i + \theta)$ relative to $|\psi\rangle$, so applying the ψ-reflection operator $2|\psi\rangle\langle\psi| - I$ maps $|\chi\rangle$ to a state with relative angle $\theta_i + \theta$ to $|\psi\rangle$, and hence to angle $\theta_i + 2\theta$ from the $|t\rangle$ axis. In other words, $\theta_{i+1} = \theta_i + 2\theta$.

Since $\theta_0 = \theta$, the recursive relation implies that after T time steps we will have $\theta_T = (2T+1)\theta$. We want θ_T to be as close to $\frac{\pi}{2}$ as possible, since at the end of the repeated oracle-and-reflection process, we want a state that with high probability is measured in the state $|s\rangle$. Hence

$$(2T+1)\theta = (2T+1)\frac{1}{\sqrt{2^n}} = \frac{\pi}{2} \implies T \approx \frac{\pi}{4}\sqrt{2^n} = O(\sqrt{2^n}).$$

This completes our analysis of Grover's search algorithm.

Note: Throughout this analysis we assumed that the number of $s \in \{0,1\}^n$ such that $f(s) = 1$ denoted by M satisfied $M = 1$. If $M > 1$ and M is known, then θ is actually roughly $\sqrt{\frac{M}{2^n}}$ so we

can run for $T = O\left(\sqrt{\frac{2^n}{M}}\right)$ time steps. If M is unknown, we can binary search on M, which in worst case scenario gives runtime $O(n\sqrt{2^n})$ if $M = 0$.

Note: We can also try to factor in the time spent by the oracle in calculating f. For example, if we suppose f is a 3-SAT instance, i.e. $f = (x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5) \wedge (x_6 \vee x_7 ...)$, then classically, applying the oracle $\mathcal{O}_f$ takes $t_{oracle} \approx n^3$ steps. We then have total steps $T_{classical} = O(n^3 2^n)$, $T_{quantum} = O(n^3\sqrt{2^n})$. Often for f this oracle time is dominated by the search time from Grover's.

Note: In the context of a database search, assuming no other structure, we can see how Grover's algorithm gives an $O(\sqrt{N})$ time to find a marked element. If there are N database elements, we can encode them in roughly $\log_2 N$ bits, so Grover's search gives $O\left(2^{\frac{\log_2 N}{2}}\right) = O(\sqrt{N})$.

Does Grover's algorithm prove that quantum computation has a definite advantage over classical computation? Unfortunately, the lower big-O complexity of search queries in Grover's algorithm is **not** evidence that the quantum case is better than classical the classical case: to make this statement, we need to prove a tight lower bound on the classical runtime. We can, however, say that Grover search achieves the best possible query complexity amongst quantum algorithms to solve the search problem by employing a highly useful technique known as the *polynomial method*.

6.1.3 Optimality

Here, we discuss why Grover's algorithm is an optimal quantum algorithm. To do so, we must consider how many queries to the oracle are required to solve the problem. This is answered by the following theorem:

Theorem 6.1.1. *Given an unknown $f : \{0,1\}^n \to \{0,1\}$ and $\mathcal{O}_f$, any quantum algorithm needs $\Omega(\sqrt{2^n})$ queries to $\tilde{\mathcal{O}}_f$ to determine whether*

Case 1. $\nexists\ s \in \{0,1\}^n$ st. $f(s) = 1$

Case 2. $\exists$ at least one $s \in \{0,1\}^n$ st. $f(s) = 1$

with success probability $\geq \frac{2}{3}$.

The proof of this theorem involves the polynomial method (introduced in [BBC$^+$01]). In the following discussion, we give an overview of the main ideas that go into the proof.

Main concepts of proof: The primary observation of the proof is that quantum algorithms give multivariate polynomials in $y_1, y_2, y_3 ... y_{2^n}$; each variable is a bit. Here we define

$$y_x = f(x), \quad \forall x \in \{0,1\}^n.$$

Note that x is a bit string, whereas, with some abuse of notation, we treat the subscripts to y as the decimal version of x. Now, we can write:

$$\tilde{\mathcal{O}}_f = \sum_x (-1)^{y_x} |x\rangle\langle x| = \sum_x (1 - 2y_x) |x\rangle\langle x| \tag{6.5}$$

Thus, when we apply $\tilde{\mathcal{O}}_f$, where $|\psi\rangle$ is any initial state, we have:

$$|\psi\rangle \to \tilde{\mathcal{O}}_f |\psi\rangle = \sum_x (-1)^{y_x} |x\rangle \langle x|\psi\rangle = |\psi_1\rangle$$

Measuring 0 (1) can be denoted by the projective operator Π_0 (Π_1), corresponding to Case 1 (Case 2). So, the probability of getting 0, for example, is given by

$$\langle\psi_1| \Pi_0 |\psi_1\rangle = \sum_{x,x'} (1-2y_{x'})(1-2y_x) \langle\psi|x'\rangle \langle x|\psi\rangle \langle x'| \Pi_0 |x\rangle$$

In general, after T queries let the state of the algorithm be $|\psi_T\rangle$. Then the probability of measuring outcome 0 is given by $P_{2T}(\{y_x\}_{x\in\{0,1\}^n}) = \langle\psi_T|\Pi_0|\psi_T\rangle$, which is a degree $2T$ polynomial. If we define a series of unitaries U_i for each $i \leq T$, then $|\psi_T\rangle$ is obtained by a general expression

$$U_T\tilde{\mathcal{O}}_f \cdot \cdots\cdots \tilde{\mathcal{O}}_f U_2 \tilde{\mathcal{O}}_f U_1 \tilde{\mathcal{O}}_f |\psi_0\rangle.$$

This will generate amplitudes that are degree T polynomials in the y_x's, so the degree $2T$ result on expectations follows.

A formal argument can be made as follows.

Lemma 6.1.2. *Consider an algorithm after T queries on input $x \in \{0,1\}^n$. The final state can be written:*

$$\sum_{x\in\{0,1\}^n} \alpha_x(y_1, ..., y_{2^n}) |x\rangle \tag{6.6}$$

Where each $\alpha_x(y_1, ..., y_{2^n})$ is a multilinear polynomial in $y_1, y_2, ..., y_{2^n}$ with degree at most T.

Proof. We can prove this by induction.

Base case: $T = 0$

The algorithm's state after 0 queries is $U_0 |0\rangle^{\otimes n}$, where U_i is the unitary applied by the algorithm following the ith query. This is independent of y, so the coefficients are constants of degree 0.

Assumption: Assume that the state after T queries is

$$\sum_{x\in\{0,1\}^n} \alpha_x(y_1, ..., y_{2^n}) |x\rangle ,$$

where $\alpha_x(y_1, ..., y_{2^n})$ is a polynomial in $y_1, y_2, ..., y_{2^n}$ with degree at most T.

Induction: After the $T + 1^{\text{th}}$ query, the state will look like:

$$\begin{aligned} U_{T+1} \cdot \tilde{O}_f \sum_{x\in\{0,1\}^n} \alpha_x(y_1, ..., y_{2^n}) |x\rangle &= \sum_{x\in\{0,1\}^n} \alpha_x(y_1, ..., y_{2^n}) U_{T+1} \cdot \tilde{O}_f |x\rangle \\ &= \sum_{x\in\{0,1\}^n} \alpha_x(y_1, ..., y_{2^n})(1-2y_x) U_{T+1} |x\rangle \end{aligned}$$

Since $\alpha_x(y_1, ..., y_{2^n})$ is no more than degree T in $y_1, y_2, ..., y_{2^n}$, $\alpha_x(y_1, ..., y_{2^n})(1-2y_x)$ is no more than degree $T+1$ in $y_1, y_2, ..., y_{2^n}$. As U_{T+1} does not depend on the y variables, we see

that the final state after $T+1$ queries can be written

$$\sum_{x\in\{0,1\}^n} \alpha_x(y_1,...,y_{2^n})(1-2y_x)U_{T+1}\,|x\rangle$$
$$= \sum_{x\in\{0,1\}^n} \beta_x(y_1,...,y_{2^n})\,|x\rangle$$

where $\beta_x(y_1,...,y_{2^n})$ is a multivariate polynomial in $y_1,...,y_{2^n}$ with no more than degree $T+1$. We have thus shown Lemma 6.1.2 by induction. □

Now, the probability to see a projective measurement Π_0 on the final state after T queries will be:

$$\sum_{x'\in\{0,1\}^n}\sum_{x\in\{0,1\}^n} \alpha_x(y_1,...,y_{2^n})\alpha_{x'}(y_1,...,y_{2^n})\left\langle x'\middle|\Pi_0|x\right\rangle$$

By Lemma 6.1.2, since $\alpha_x, \alpha_{x'}$ are no more than degree T, this probability will be no more than degree $2T$ in $y_1,...,y_{2^n}$.

Now, if f is in Case 1, all the y_x are zero, and we want

$$P_{2T}(\{y_x\}_{x\in\{0,1\}^n}) = P_{2T}(0,0,...) \geq \frac{2}{3}. \tag{6.7}$$

If even one of the inputs change, we need

$$P_{2T}(0,\cdots,1,0,...) \leq \frac{1}{3}. \tag{6.8}$$

Thus, the polynomial needs to change very fast in its variables. The final step of the proof is that the polynomial must be of very high degree to accommodate this change, namely,

$$2T = \Omega(\sqrt{2^n}).$$

For example, consider a particularly symmetric case where

$$P_{2T}(\{y_x\}_{x\in\{0,1\}^n}) = Q_{2T}\left(\frac{y_1+y_2+\ldots+y_{2^n}}{2^n}\right),$$

for a polynomial Q_{2T} of degree $2T$. In other words, P_{2T} only depends on the hamming weight of its input. By the conditions in Equations 6.7,6.8 on P_{2T}, we have $Q_{2T}(0) \geq \frac{2}{3}$ and $Q_{2T}\left(\frac{1}{2^n}\right) \leq \frac{1}{3}$. This change is too fast, unless $T = \Omega(\sqrt{2^n})$ (see Markov's inequality for formal proof; [SV14, Page 29, Theorem 5.1]). In general, $P_{2T}(y_1, y_2, \ldots y_{2^n})$ may not be in this nice form. However, it is possible to reduce to this form via a symmetry argument. The idea is that the conditions in Equations 6.7,6.8 on P_{2T} stay the same if the input bits are permuted arbitrarily.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 7

February 16, 2022 *Scribe: Andrew Lu*

7.1 Quantum Fourier transform over $\{0,1\}^n$

Recall the Hadamard gate $H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$. Then we refer to the unitary operator $H^{\otimes n}$ as the *quantum Fourier transform*; note that

$$H^{\otimes n} |0\rangle^{\otimes n} = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} |z\rangle \text{ where } |z\rangle = |z_1\rangle \otimes |z_2\rangle \otimes \cdots \otimes |z_n\rangle ,$$

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

where $x.z$ is the *Boolean inner product* of strings x and z, defined by $x.z = \bigoplus_{i=1}^n x_i z_i$. Moreover, if over all $x \in \{0,1\}^n$ we have some amplitude function $a : \{0,1\}^n \to \{-1,1\}$, we have the more general formula representing the Fourier transform of the amplitudes:

$$\begin{aligned} H^{\otimes n} \sum_{x \in \{0,1\}^n} a(x) |x\rangle &= \frac{1}{\sqrt{2^n}} \sum_x a(x) H^{\otimes n} |x\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_x a(x) \sum_z (-1)^{x.z} |z\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_z \overbrace{\left(\frac{1}{\sqrt{2^n}} \sum_x a(x)(-1)^{x.z} \right)}^{\tilde{a}(z)} |z\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_z \tilde{a}(z) |z\rangle . \end{aligned}$$

Thus, performing a Fourier transform over $\{0,1\}^n$ takes n steps and circuit depth 1.

7.2 Simon's Problem

Last week: We looked at Grover's algorithm and how it gives a quadratic speedup to a search problem. But the problem is *unstructured*; we know nothing about the underlying nature of the function being queried. Simon's problem is an example of a function with more structure such that quantum algorithms can achieve a higher degree of speedup over classical algorithms.

Like in Lecture 6, define an oracle O_f for function $f : \{0,1\}^n \to \{0,1\}^n$ as follows:

$$O_f |x\rangle \otimes |b\rangle = |x\rangle \otimes |b \oplus f(x)\rangle .$$

Problem: Given a function $f : \{0,1\}^n \to \{0,1\}^n$ which has the property that there exists nonzero $s \in \{0,1\}^n$ satisfying $f(x \oplus s) = f(x)$ for all $x \in \{0,1\}^n$, find s with minimal querying. While classical (randomized) algorithms require $\Theta(\sqrt{2^n})$ queries, but there exists a quantum algorithm that makes only $O(n)$ queries.

Simon's algorithm:

1. Prepare $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |0\rangle^{\otimes n}$.

2. Query O_f to obtain $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in Range(f)} (|x\rangle + |x \oplus s\rangle) |y\rangle$, where for each y, the x term is one of the two n-bit strings that maps to y under f.

3. Measure the latter n qubits in the standard computational basis. We get a y with probability $\frac{1}{2^{n-1}}$, and the quantum state of the first n qubits is now $\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)$.

4. Perform QFT:

$$\begin{aligned} H^{\otimes n}(|x\rangle + |x \oplus s\rangle) &= \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} \left((-1)^{x.z} + (-1)^{x.z \oplus s.z}\right) |z\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_z (-1)^{x.z} \left(1 + (-1)^{s.z}\right) |z\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n : s.z = 0} (-1)^{x.z} |z\rangle . \end{aligned}$$

   Note that if we measure in the standard computational basis, we observe a random z satisfying $s.z = 0$.

5. Repeat the entire above procedure n times to obtain $z_1, z_2, \ldots, z_{n-1}$, where $s.z_i = 0$ for all i. With high probability, the $n-1$ vectors will be linearly independent, so in $O(n)$ queries we can calculate s via solving a system of linear equations.

Classical version analysis: In the classical model, note that we need a collision (two bit-strings x, x' satisfying $f(x) = f(x')$) to be able to find s. If we keep choosing distinct values of $x \in \{0,1\}^n$ at random T times without replacement, the probability of no collisions is roughly $\left(1 - \frac{1}{2^{n-1}}\right)\left(1 - \frac{2}{2^{n-1}}\right) \cdots \left(1 - \frac{T-1}{2^{n-1}}\right)$. The failure probability is on the order of $e^{-O(T^2/2^n)}$, so we must have $T = \Omega(\sqrt{2^n})$ in order to achieve error at most $1/3$.

A alternative sketch of the lower bound is as follows: we can re-interpret our problem as distinguishing between a 1-to-1 function from a 2-to-1 function when we guess s. When we query $x_1, \ldots, x_{T-1}$ without seeing a collision, we have eliminated $\binom{T-1}{2}$ candidates for s, and querying a Tth input x_T can only eliminate a $\frac{T-1}{(2^n-1)-\binom{T-1}{2}}$ fraction of the remaining candidates. After T steps, the probability that no collisions occur is

$$\prod_{k=1}^{T} \left(1 - \frac{k}{(2^n - 1) - \binom{k}{2}}\right) \geq \left(1 - \frac{T}{(2^n - 1) - \binom{T-1}{2}}\right)^T > 0.99 \text{ for } T = o(\sqrt{2^n}).$$

7.3 Forrelation

Given two functions $f, g : \{0,1\}^n \to \{0,1\}$ and their corresponding oracles O_f, O_g, define the *forrelation* (Fourier + correlation) between f and g

$$\mathcal{F}(f,g) = \frac{1}{2^{3n/2}} \sum_{x,y} (-1)^{x.y} a(x) b(y) = \frac{1}{2^n} \sum_x a(x) \tilde{b}(x),$$

where $a, b : \{0,1\}^n \to \{-1,1\}$ are amplitude functions defined by $a(x) = (-1)^{f(x)}$, $b(x) = (-1)^{g(x)}$. Similarly, define the correlation function

$$\zeta(x,y) = \frac{1}{2^n} \sum_x a(x) b(x).$$

There exists a randomized algorithm using $O(1/\epsilon^2)$ queries which involves sampling points and returning the empirical average (which is within ϵ additive error with high probability by Chernoff bound).

Theorem 7.3.1. *There exists a quantum algorithm to evaluate $\mathcal{F}(f,g) \pm \epsilon$ using $O(1/\epsilon)$ queries (as opposed to $\Omega(\sqrt{2^n}/n)$ queries in the classical randomized setting).*

A quick sketch of the algorithm: we can write

$$\mathcal{F}(f,g) = \langle 0^n | \, H^{\otimes n} O_f H^{\otimes n} O_g H^{\otimes n} \, |0^n\rangle \,,$$

where we define unitary operators $O_f |x\rangle = f(x) |x\rangle$ and $O_g |x\rangle = g(x) |x\rangle$. So the idea is to apply $H^{\otimes n} O_f H^{\otimes n} O_g H^{\otimes n}$ on $|0\rangle^{\otimes n}$ and then conduct a projective measurement onto $\{|0^n\rangle \langle 0^n| \,, I_n - |0^n\rangle \langle 0^n|\}$. Then $\mathcal{F}(f,g)$ is linearly related to the probability of observing a $|0\rangle$ in the ancilla register. By a Chernoff bound argument, if we iterate and perform $O(\epsilon^{-1})$ measurements, we will be able to approximate $\mathcal{F}(f,g)$ within an additive error of ϵ with probability at least $\frac{2}{3}$.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 08

February 18, 2022 *Scribes: Richard Qiu, Matthew Nazari*

8.1 Introduction to Phase Estimation

We will only briefly and informally discuss the phase estimation procedure today. Next lecture and in the future, we will return and add formalism. Nevertheless, phase estimation is an important technique.

Recall that a unitary operator V has complex eigenvalues of unit norm, so we can write them as $\{e^{2\pi i q_j}\}$, where $q_j \in (0,1)$ is the phase. Likewise, we can write the eigenvectors as $\{|\phi_j\rangle\}$. By the spectral theorem, we can then write V in its eigenbasis:

$$V = \sum_J e^{2\pi i q_k} |\phi_j\rangle \langle\phi_j|$$

8.2 Informal Definition

Given error parameters of ϵ, δ, phase estimation with respect to a unitary V (which we will denote PE_V) is a quantum algorithm which takes as input an eigenvector $|\phi_J\rangle$ and outputs $O \in \mathbb{R}$ such that $\Pr(O \in q_j \pm \epsilon) \geq 1 - \delta$ with $\mathcal{O}(\frac{1}{\epsilon\delta})$ calls to V.

Remark: phase estimation works in superposition, so $\frac{1}{2}|\phi_j\rangle + \frac{\sqrt{3}}{2}|\phi_{j'}\rangle$ works as an input and PE_V will return either $q_j \pm \epsilon$ or $q_{j'} \pm \epsilon$, with probability $1 - \delta$.

8.3 An Example

Let's look at a simple example of what a phase estimation could look like. Suppose V has eigenvalues drawn from $\{-1, 1\}$ (so $q_j \in \{0, 1/2\}$) so we can write $V = \sum_j (-1)^{x_j} |\phi_j\rangle \langle\phi_j|$. Then the algorithm might look like:

Step 1: Start with input state $|\psi\rangle$. This can be an eigenvector or a superposition of eigenvectors. Add an ancilla register $|+\rangle_z$ so that our state is now $|+\rangle_z \otimes |\psi\rangle$.

Step 2: Apply the controlled gate $|0\rangle\langle 0|_z \otimes I + |1\rangle\langle 1|_z \otimes V$. If our ancilla register is measured as 0, apply V. We can verify that this gate is indeed a unitary.

Step 3: Apply the Hadamard to our ancilla register: $H_z \otimes I$.

Step 4: Measure the ancilla register in the computational basis.

If our input state $|\psi\rangle = |\phi_j\rangle$, then our algorithm does:

$$\textbf{Step 1:} \quad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\phi_j\rangle$$

$$\textbf{Step 2:} \quad \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_j}|1\rangle) \otimes |\phi_j\rangle$$

$$\textbf{Step 3:} \quad |x_j\rangle \otimes |\phi\rangle$$

And after measurement, we would get x_j (without error in this special case, where our eigenvalues are ± 1). Similarly (ignoring normalization), if our input was a superposition of eigenvectors $|\phi_j\rangle + |\phi_{j'}\rangle$, our algorithm would return $|x_j\rangle_Z \otimes |\phi_j\rangle + |x_{j'}\rangle_Z \otimes |\phi_{j'}\rangle$ before measurement.

The algorithm for a more complex V follows this same idea, and we will return to this in a later lecture.

8.4 Importance

Why is phase estimation so important? Historically, phase estimation was introduced by Kitaev[1] as an alternate approach to Shor's algorithm. We'll see that many algorithms including Grover's search algorithm can be viewed in terms of phase estimation.

8.5 Phase Estimation for Search

Recall the search problem: suppose we have a function $f : \{0,1\}^n \to \{0,1\}$ with solution set $S = \{s \in \{0,1\}^n \mid f(s) = 1\}$. In the search problem, we want to output any $s \in S$.

Grover's search algorithm uses a phase oracle $\tilde{O}_f$:

$$\tilde{O}_f |x\rangle = \begin{cases} |x\rangle & x \notin S \\ -|x\rangle & x \in S \end{cases}$$

By successively applying $\tilde{O}_f$ and $W = H^{\otimes n}(2|0\rangle\langle 0|^{\otimes n} - I)H^{\otimes n}$ a precise number of times, we can find $s \in S$ with high probability.

Also recall that we can turn the search problem into a decision problem with the following 2 cases:

Case 1: S is empty

Case 2: $|S| = 1$

Now, we will show that we can use phase estimation to solve the decision problem. First, we'll need to find the spectrum of $W\tilde{O}_f$. In case 1, $\tilde{O}_f = I$ and W can be written in the Hadamard basis as:

$$W = \begin{bmatrix} 1 & & & & \\ & -1 & & & \\ & & -1 & & \\ & & & \ddots & \\ & & & & -1 \end{bmatrix}$$

[1] https://arxiv.org/abs/quant-ph/9511026

In case 2, we have $S = \{s\}$ and $\tilde{O}_f = I - 2|s\rangle\langle s|$. We also have $W = 2|\psi\rangle\langle\psi| - I$, where $|\psi\rangle = H^{\otimes n}|0\rangle^{\otimes n}$. Then, as in Grover's algorithm, we can view $W\tilde{O}_f$ as a rotation in the 2-dimensional subspace spanned by $|s\rangle$ and $|s'\rangle$ (the component of $|\psi\rangle$ orthogonal to $|s\rangle$):

$$W\tilde{O}_f = \begin{bmatrix} -1 & & & & \\ & \ddots & & & \\ & & -1 & & \\ & & & \cos 2\theta & -\sin 2\theta \\ & & & \sin 2\theta & \cos 2\theta \end{bmatrix}$$

The eigenvalues and eigenvectors of $\begin{bmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{bmatrix}$ are $e^{\pm 2i\theta}$ and $\begin{bmatrix} 1 \\ \pm i \end{bmatrix}$, respectively.

Now, we'll make the connection to phase estimation. We'll use unitary $V = W\tilde{O}_f$ and input vector $|\psi\rangle = H^{\otimes n}|0\rangle^{\otimes n}$. We'll call PE_V with $\epsilon = \frac{\theta}{3\pi}$ and $\delta = 0.01$. In case 1, $V|\psi\rangle = |\psi\rangle$ so PE_V returns a phase in $0 \pm \epsilon = (-\frac{\theta}{3\pi}, \frac{\theta}{3\pi})$. In case 2, $W\tilde{O}_f$ applied any number of times lives in the subspace spanned by $|s\rangle$ and $|s'\rangle$, so PE_V returns a phase in either $\frac{\theta}{\pi} \pm \epsilon = (\frac{2\theta}{3\pi}, \frac{4\theta}{3\pi})$ or $-\frac{\theta}{\pi} \pm \epsilon = (-\frac{4\theta}{3\pi}, -\frac{2\theta}{3\pi})$. Thus, since these are disjoin intervals, we can determine which case our problem lives in based on the output PE_V.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 09

February 23, 2022 *Scribes: Sabina Dragoi, Kevin Mao*

Today: Continuing Phase Estimation, Shor's Algorithm, and Quantum Fourier Transform

9.1 Recap of Quantum Phase Estimation

Last time, we looked at the Quantum Phase Estimation algorithm. As a recap, this algorithm's goal is to estimates the phase of a unitary operator U, for which we we know one of its eigenstates $|\psi\rangle$. Since this operator is unitary, then the norm of all its eigenvalues is 1, and hence we can denote the eigenvalue corresponding to $|\psi\rangle$ as $e^{2\pi i\theta}$. Here, θ is is the phase that we want to estimate. Recall that for this algorithm we use 2 registers: the counting register, and the qubit register for $|\psi\rangle$. The phase will be encoded in the counting register at the end of this algorithm, after we perform the final measurement on the counting register. Throughout this calculation, we assume that the counting register has n qubits, so there are 2^n possible binary representation, or, equivalently, we can measure the phase with precision $2\pi/2^n$.

Concretely, we looked at an example of a unitary whose eigenvalues were $\{1, -1\}$. Since these are just 2 possible values for the phase, this means that we only required $\lceil \log_2 2 \rceil = 1$ qubit in the counting register.

The first step in the algorithm is to apply a Hadamard gate on all the qubits in the counting register after initializing all of them in state $|0\rangle$. Then, we apply the **Controlled Unitary Gate** depending on the state of the qubits in the counting register, and then apply again Hadamard gates on all the qubits in the counting register. In the end, the state of the qubits in this register will give the phase we were looking for after we perform a measurement.

Step 1: Initialize	$\|0\rangle \|\psi\rangle$
Step 2: Apply H on counting qubits	$\|+\rangle \|\psi\rangle$
Step 3: Apply $\Big(\|0\rangle\langle 0\| \otimes I + \|1\rangle\langle 1\| \otimes U \Big)$	
Step 4: Apply H on counting qubits again	$\|\theta\rangle \|\psi\rangle$

Note: This also works for when $|\psi\rangle$ is not an eigenstate of U, but rather a superposition of its eigenstates.

9.2 Quantum Fourier Transform

One way of introducing the Quantum Fourier Transform is to view it as natural step of Quantum Phase Estimation. Let us now see how this works concretely.

Consider the general form of a unitary $V = \sum_J e^{2\pi i q_J} |\phi_J\rangle \langle\phi_J|$, where $|\phi_J\rangle$ are its eigenstates and q_J are the corresponding eigenvalues. For now also assume that $q_J = \frac{J}{2^n}$, for some $J, n \in N$, $J \in \{0, \ldots, 2^n - 1\}$, where n is the number of qubits in the counting register. After Step 1 of the previous section, we have our system in the total state:

$$\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} |y\rangle_Z \otimes |\phi_J\rangle$$

In Step 2, we apply V^y if you see $|y\rangle \langle y|_Z$, or equivalently, the global operator:

$$\sum_y |y\rangle \langle y|_Z \otimes V^y$$

Doing so gives the following final state:

$$\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} |y\rangle \otimes \Big(|\phi_J\rangle e^{2\pi i y \cdot J/2^n} \Big) \text{ because } V^y |\phi_J\rangle = e^{2\pi i y \cdot J/2^n} |\phi_J\rangle$$

By moving the exponential, since it is just a number (not a ket), this is further equivalent to:

$$\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \Big(e^{2\pi i y J/2^n} |y\rangle \Big) \otimes |\phi_J\rangle$$

Now we know the final state after measuring should give the phase in the counting register, i.e. $|\theta_J\rangle \otimes |\phi_J\rangle$. We would therefore want to have an operator that takes the superposition of the first register and maps it to this phase. This is the Inverse Quantum Fourier Transform:

$$|\theta_J\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i y J/2^n} |y\rangle$$

As mentioned before, we call this transformation, the **Inverse** Quantum Fourier Transform:

$$\boxed{(QFT)^{-1} |\theta_J\rangle = |J\rangle}, \text{ for } J \in \{0, \ldots, 2^n - 1\}$$

By analogy, we can then also define the Quantum Fourier Transform, which should take basis vectors from the initial basis $\{|0\rangle, \ldots, |2^n - 1\rangle\}$ to the Fourier basis $\{|\theta_0\rangle, \ldots, |\theta_{2^n-1}\rangle\}$:

$$\boxed{(QFT) |J\rangle = |\theta_J\rangle}, \text{ for } J \in \{0, \ldots, 2^n - 1\}$$

We made a bold statement above, which is that the QFT and IQFT are basically operators that switch bases. This implies the existence of what is known as the **Fourier basis**, which is defined by applying QFT on the basis vectors of our initial space, i.e. $\{|\theta_0\rangle, \ldots, |\theta_{2^n-1}\rangle\}$. Here is a short proof that these resulting vectors indeed form an orthonormal basis:

$$\begin{aligned}
\langle\theta_k|\theta_j\rangle &= \Big(\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{-2\pi i y k/2^n} \langle y| \Big) \Big(\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i y j/2^n} |y\rangle \Big) \\
&= \frac{1}{2^n} \sum_{y=0}^{2^n-1} e^{2\pi i y (j-k)/2^n} = \delta_{kj}
\end{aligned}$$

Although we only used the QFT for basis vectors in this example, this operator obviously applies to states in superposition as well, just as the Phase Estimation also does:

$$|\phi_1\rangle + |\phi_2\rangle \mapsto |\theta_1\rangle |\phi_1\rangle + |\theta_2\rangle |\phi_2\rangle$$

In particular, the Quantum Fourier Transform takes a vector $|v\rangle = \sum_{j=0}^{N=2^n-1} v_j |j\rangle$ and maps it to a vector $|\tilde{v}\rangle = \sum_{k=0}^{N=2^n-1} \tilde{v}_k |k\rangle$ using the transformation:

$$\tilde{v}_k = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i k y/N} v_y, \quad y \in \{0, 1, \ldots, N\}$$

9.3 Shor's algorithm

Shor's algorithm is motivated by the following problem:

Goal: Given N, find an output m that divides N (in polynomial time)

One potential way to approach this problem is through order/period finding, that is, given a, N, we want to output the smallest nonzero $r \in \mathbb{Z}$ such that $a^r \equiv 1 \mod N$. Let $n = \lceil \log_2 N \rceil$. Note that we want a to be co-prime with N, or else the operator we are about to define is not unitary. Further note that the probability of this condition being met is incredibility close to 1, especially for numbers that are the product of 2 large primes, such as in RSA encryption.

For simplicity, let us focus on the setting where $N = pq$ for two even primes and see how this can reduced to order/period finding. This is also the setting used in RSA encryption.

First, note that if you find an a with its period r such that (i) $a^r \equiv 1 \mod N$ and (ii) r is even, then you immediately have $(a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \mod N$. As both $a^{r/2} - 1$ and $a^{r/2} + 1$ are not multiples of N, they both share a non-trivial prime factor with N. In the setting where $N = pq$, we then know that $\{p, q\}$ will be $\{\mathsf{gcd}(a^{r/2} - 1, N), \mathsf{gcd}(a^{r/2} + 1, N)\}$.

Next, how to find (a, r) with property (i) and (ii)? it turns out that from number theory, we know that when $N = pq$, there's $\delta = \Omega(1)$ chance to get such an a with period r when a is randomly sampled from $1 < a < N$.

In summary, to reduce factoring $N = pq$ to period finding, we randomly samples lots of $a_1, a_2, \ldots, a_k$ and run the order/period finding for each of them to get $r_1, r_2, \ldots, r_k$. We are done as long as one of r_i is even and this happens with probability at least $1 - (1 - \delta)^k$.

Shor's algorithm approaches this through quantum phase estimation on the following unitary operator

$$V_a |y\rangle = |ya \mod N\rangle$$

where $|y\rangle$ is in the $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$ basis. We note that

$$V_a^z |y\rangle = |ya^z \mod N\rangle$$

Letting $\omega = e^{2i\pi/r}$, we have the following observations for Shor's algorithm. Firstly, we observe that the following state is an eigenstate of V_a:

$$\frac{1}{\sqrt{r}}(|1\rangle + \omega |a\rangle + \omega^2 |a^2\rangle + \cdots + \omega^{r-1} |a^{r-1}\rangle$$

Next, applying unitary V, we get

$$\begin{aligned}&\frac{1}{\sqrt{r}}(|a\rangle + \omega \left|a^2\right\rangle + \omega^2 \left|a^3\right\rangle + \cdots + \omega^{r-1} |1\rangle \\ &= \frac{w^{r-1}}{\sqrt{r}}(\omega |a\rangle + \omega^2 \left|a^2\right\rangle + \cdots + |1\rangle)\end{aligned}$$

This shows that w^{r-1} is an eigenvalue of V_a. To show it in general, we would need more eigenstates. These take the following form:

$$|\lambda_k\rangle = \frac{1}{\sqrt{2^n}}\Big(|1\rangle + e^{-\pi i k/2^n} |y\rangle + e^{-\pi i 2k/2^n} \left|y^2\right\rangle + \dots \Big)$$

More generally, we can define $|\lambda_k\rangle$ as

$$|\lambda_k\rangle := \frac{1}{\sqrt{r}}(|1\rangle + \omega^k |a\rangle + \omega^{2k} \left|a^2\right\rangle + \cdots + \omega^{(r-1)k} \left|a^{r-1}\right\rangle$$

for $k = 1, 2, \dots, r-1$, and as a result, we would have

$$|1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\lambda_k\rangle \ .$$

It is not hard to prove that the corresponding eigenvalues are w^k, where k is an integer. Details can be found on Page 4 of this note.

Another crucial observation is that $|1\rangle$ can be written as a linear combination of the eigenvectors above:

$$|1\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |\lambda_k\rangle$$

Then, from this setup, there are four main facts we can observe:

1. The phases of V_a are of the form $\frac{k}{r}, k \in \mathbb{Z}$
2. Phase estimation on V_a, PE_{V_a}, works in superposition. In particular, phase estimation works on the state $|1\rangle$, which is the uniform superposition of $|\lambda_0\rangle, \dots, |\lambda_{r-1}\rangle$, as established above.
3. From this we will get k/r where k is uniformly random from $k \in \{0, 1, \dots, r-1\}$. When k and r are co-prime, the we can use the "continued fraction" to estimate r.
4. V_a^z can be phase estimated in poly(n) steps for all $z \in \{0, 1, \dots, 2^n - 1\}$

Main take-away: Shor's algorithm works in poly(n) time versus the classical $2^{n^{1/3}}$! (though this has not been proven to be classically optimal bound)

9.4 Quantum Fourier Transform Revisited

Recall that

$$(QFT)\,|k\rangle = |\theta_k\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n} |y\rangle\, e^{2\pi i k y/2^n}, \text{ for } k \in \{0, \ldots, 2^n - 1\}$$

In order to understand the Quantum Fourier Transform, it helps to look at the classical equivalent, namely the Discrete Fourier Transform, which changes the basis $\{v_0, v_1, \ldots, v_{N=2^n-1}\}$ to another basis $\{\tilde{v}_0, \tilde{v}_1, \ldots, \tilde{v}_{N=2^n-1}\}$ (which might be more helpful for one's purposes. For example, it is already helpful in integer multiplication).

$$\tilde{v}_k = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n} e^{2\pi i k y/2^n} v_y$$

where $\tilde{v}_k$ is the Fourier transform of v_k. The Discrete Fourier Transform takes $O(N^2) = O(2^{2n})$ steps, but the Fast Fourier Transform is a way that implements this same operator in $O(N \log N) = O(n2^n)$.

Let's discuss FFT first: Expressing numbers in binary, which we write in the form $\ell_1 \ell_2 \ldots \ell_n$ and $y_1 y_2 \ldots y_n$, we have:

$$\tilde{v}_{\ell_1,\ldots,\ell_n} = \frac{1}{\sqrt{2^n}} \sum_{y_1,\ldots y_n} e^{2\pi i (y_1 y_2 \ldots y_n)(\ell_1 \ell_2 \ldots \ell_n)/2^n} v_{y_1,\ldots,y_n}$$

We can split this sum into cases, where the two cases are $y_n = 0$ and $y_n = 1$. In the first case of $y_n = 0$, we know that $y_1 \ldots y_n$ is even, so we can divide the denominator and numerator of $2\pi i (y_1 y_2 \ldots y_n)(\ell_1 \ell_2 \ldots \ell_n)/2^n$ by 2, giving us the first term in the following sum. In the $y_n = 1$ case, we can "factor out" the $y_n = 1$, which gives us the second term in the following sum.

$$\begin{aligned} &= \frac{1}{\sqrt{2^n}} \sum_{y_1,\ldots y_{n-1}} e^{2\pi i (y_1 y_2 \ldots y_{n-1})(\ell_1 \ell_2 \ldots \ell_n)/2^{n-1}} v_{y_1,\ldots,y_{n-1},0} \\ &+ e^{2\pi i \ell_1/2^n} \frac{1}{\sqrt{2^n}} \sum_{y_1,\ldots y_{n-1}} e^{2\pi i (y_1 y_2 \ldots y_{n-1})(\ell_2 \ldots \ell_n)/2^{n-1}} v_{y_1,\ldots,y_n} \end{aligned}$$

QFT is a quantization of the above, and works in a similar recursive manner. Concretely, QFT takes a vector $|v\rangle = \sum_{j=0}^{N=2^n-1} v_j\, |j\rangle$ and maps it to a vector $|\tilde{v}\rangle = \sum_{k=0}^{N=2^n-1} \tilde{v}_k\, |k\rangle$ using the transformation:

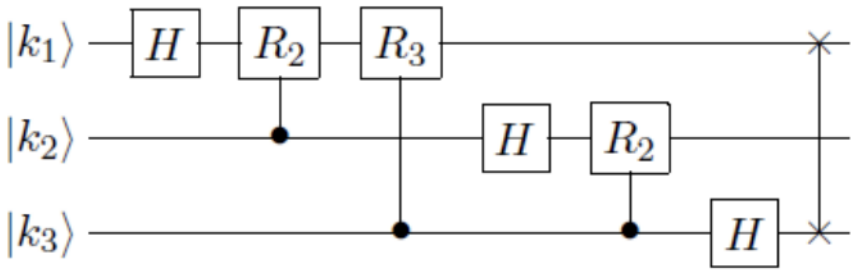
$$\tilde{v}_k = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i k y/N} v_y, \quad y \in \{0, 1, \ldots, N\}$$

Hence, for basis states we have the following transformation:

$$(QFT)\,|k\rangle = |\theta_k\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n} |y\rangle\, e^{2\pi i k y/2^n}, \text{ for } k \in \{0, \ldots, 2^n - 1\}$$

Since multi-qubit gates are physically hard to implement, then we want to achieve this operator by only using 2-qubit gates. The gates that we will in the quantum circuit will be Hadamard gates and controlled phase gates $R_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-2\pi i/2^k} \end{pmatrix}$.

The following circuit is an example of a QFT on 3 qubits (Ronald de Wolf's "Quantum Computing: Lecture Notes"). Notice that we can draw clear parallels with FFT: first, the Hadamard gate is analogous to the 'divide' step and then the R_k gates are used to introduce the $e^{2\pi i \ell_1}$ factor. Then the circuit recurses on smaller number of qubits, i.e. just on $|k_2\rangle$ and $|k_3\rangle$.



CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 10

February 25, 2022 *Scribes: Amir Shanehsazzadeh*

Today: Quantum Walks

1. Classical random walk over a graph
2. Definition of quantum walk
3. Jordan's Lemma

10.1 Motivation

Quantum walks are useful for solving many problems, just like classical random walks. For an immediate example, note that quantum walks generalize Grover's search algorithm.

Applications:

1. Generalize Grover's
2. Element distinctness problem
3. Hamiltonian simulation

10.2 Random Walk over a Graph

Suppose we have a graph $G = (V, E)$. For connected nodes x, y let $P_{xy} = \frac{1}{d_x}$ where d_x is the degree of x (for unconnected nodes $P_{xy} = 0$). The matrix $P \in \mathbb{R}^{|V|\times|V|}$ is the transition matrix corresponding to random walk on the graph G with transition probabilities between nodes x and y given by P_{xy}. The stationary distribution corresponding to P is $\nu \in \mathbb{R}^{|V|}$ where $\nu_x = \frac{d_x}{|E|}$. To see this note that

$$\sum_{x:y\sim x} P_{xy}\nu_x = \sum_{x:y\sim x} \frac{1}{d_x}\cdot\frac{d_x}{|E|} = \sum_{x:y\sim x}\frac{1}{|E|} = \frac{d_y}{|E|} = \nu_y.$$

We can do an eigenvalue decomposition: $P = B^{-1}\cdot \mathrm{diag}(\lambda_1, \lambda_2, ...)\cdot B$ with $\lambda_1 = 1$ and $\lambda_1 > \lambda_2 > \lambda_3 > \cdots$. The spectral gap is defined as $\gamma = \lambda_1 - \lambda_2$. It is well-known that convergence time to the stationary distribution scales inversely with γ.

Detailed balance condition: $\nu_x P_{xy} = \frac{1}{|E|} = \nu_y P_{yx}$. Define the matrix D such that $D_{xy} = \sqrt{P_{xy}P_{yx}}$. Then under detailed balance we have

$$D_{xy} = \sqrt{P_{xy}P_{yx}} = P_{xy}\sqrt{\frac{\nu_x}{\nu_y}} \implies D = \mathrm{diag}(\sqrt{\nu_1}, \sqrt{\nu_2}, ...)\cdot P\cdot \mathrm{diag}\left(\frac{1}{\sqrt{\nu_1}}, \frac{1}{\sqrt{\nu_2}}, ...\right).$$

Note that D is Hermitian and has the same eigenvalues as P. We will come back to D in our analysis of the quantum walk.

Random walk without forgetting: Quantum walks will be unitary and unitaries are reversible. We want a reversible analog of the classical random walk so that we can quantize it. Some approaches:

1. **Coins:** $(x, c) = (x, \text{equal-weighted coin with } d_x \text{ sides})$. At x flip c to get value $i \in [1, 2, ..., d_x]$ and move to the ith neighbor of x. The information in the coin is sufficient for reversing the process.

2. **Edge Process:** Take left and right copies of the vertices V_L and V_R, respectively. Connect $x \in V_L$ with $y \in V_R$ if $(x, y) \in E$. This forms a bipartite graph $G' = (V', E')$.

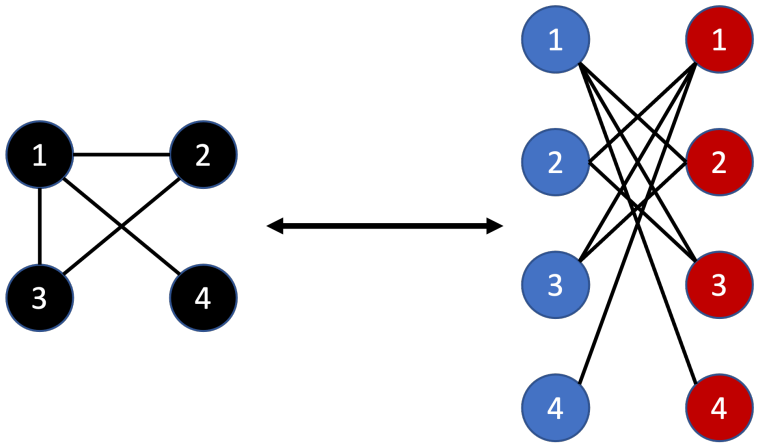


Figure 10.1: Edge Process for Random Walk

Now we perform a random walk over edges going from V_L to V_R and back: start at $e = (x, y) \in E'$ in a "left-looking" manner (corresponding to starting at $x \in V$) the next step is to uniformly randomly select a $(y, x') \in E'$ that is "right-looking" (i.e. uniformly randomly select an edge originating at y). Repeat this process of selecting "right-looking" nodes from "left-looking" nodes and vice-versa. There is a clear correspondence between edges in G' and edges in G so this random walk simulates a random walk on G. This is reversible because you move from left to right and back and so if you are at V_R you can look to the left to find the corresponding node in V_L and vice-versa.

Viewing this as a random walk over the edges we see that the stationary distribution μ is uniform over the edges: any edge e has probability $\mu(e) = \frac{1}{|E|}$. This implies $\nu_x = \frac{dx}{|E|}$ for the random walk over G. To see the former claim let e, e' be edges in the bipartite graph G' and note that

$$P(e') = \sum_e P(e'|e)P(e) = \sum_e P(e|e')P(e) = \frac{1}{|E|}\sum_e P(e|e') = \frac{1}{|E|}.$$

We use above the fact that $P(e'|e) = P(e|e')$. To see this, note that $P(e|e')$ is non-zero if and only if the edges e and e' share a vertex in V_L (left-looking step) and V_R (right-looking step), in which case $\frac{1}{P(e|e')}$ is the number of edges incident on that shared vertex. By a similar argument, we get the same value for $P(e'|e)$.

10.3 Quantum Walk

Properties:

1. A unitary W instead of a transition matrix P
2. Stationary state $|\mu\rangle$ similar to the stationary distribution of the edge random walk μ.
3. "Spectral gap" $\approx \sqrt{\gamma}$ with γ the spectral gap of P (source of speed-up)
4. Reflects about $|\mu\rangle$ (operator $2|\mu\rangle\langle\mu| - \mathbb{1}$), does not prepare $|\mu\rangle$ (source of speed-up)

An example of usefulness is search. Classically, we prepare μ, check it for a match, and then resample μ if there is no match. Quantum search gives a quadratic speed-up and involves us preparing $|\mu\rangle$, checking our state with a quantum oracle, and if necessary reflecting about $|\mu\rangle$. For Grover search we used $|\mu\rangle = |+\rangle^{\otimes n}$ and the quantum oracle $\tilde{O}_f = \mathbb{1} - 2|s\rangle\langle s|$.

Constructing the quantum walk is similar to the edge process construction of the classical random walk. Consider the quantum states:

$$|\psi_x^1\rangle = \sum_y \sqrt{P_{xy}}\,|x\rangle\otimes|y\rangle = |x\rangle\otimes\left(\sum_y \sqrt{P_{xy}}\,|y\rangle\right),\ |\psi_y^2\rangle = \sum_x \sqrt{P_{yx}}\,|x\rangle\otimes|y\rangle = \left(\sum_x \sqrt{P_{yx}}\,|x\rangle\right)\otimes|y\rangle$$

and define the unitary

$$W = \left(\left(\sum_x 2|\psi_x^1\rangle\langle\psi_x^1|\right) - \mathbb{1}\right)\cdot\left(\left(\sum_y 2|\psi_y^2\rangle\langle\psi_y^2|\right) - \mathbb{1}\right).$$

W consists of two steps: conditioning on states in the left qubit and reflecting about a superposition of those edges (left operator) and then a similar reflection for states in the right qubit (right operator).

Theorem: Let P the transition matrix of the classical walk corresponding to W. Denote the eigenvalues of P as $\{\lambda_J\}$ and note that $\lambda_1 = 1$. The eigenvalues of W are $\{e^{\pm 2i\theta_J}\} = \{1, e^{\pm 2i\theta_2}, e^{\pm 2i\theta_3}, ...\}$ where $\cos\theta_J = \lambda_J$. Note that

$$\cos\theta_2 = \lambda_2 = \lambda_1 - (\lambda_1 - \lambda_2) = 1 - \gamma,\ \cos\theta_2 \approx 1 - \frac{\theta_2^2}{2} \implies \theta_2 \approx \sqrt{2\gamma}.$$

Proof Sketch: We will use Jordan's lemma, which allows us to decompose projector matrices into block diagonal form where the blocks are 1×1 or 2×2 matrices (corresponding to 1-D and 2-D subspaces, respectively). We can write

$$\Pi_1 = \sum_x |\psi_x^1\rangle\langle\psi_x^1|,\ \Pi_2 = \sum_y |\psi_y^2\rangle\langle\psi_y^2| \implies W = (2\Pi_1 - \mathbb{1})\cdot(2\Pi_2 - \mathbb{1}).$$

Note that Π_1 and Π_2 are projectors so $\Pi_1^2 = \Pi_1$ and $\Pi_2^2 = \Pi_2$. Jordan's lemma lets us decompose Π_1 and Π_2 into matrices with 1×1 or 2×2 blocks along the diagonal.

Take the matrix D such that $D_{xy} = \sqrt{P_{xy}P_{yx}}$. Detailed balance gives us $D_{xy} = \sqrt{\mu_x} \cdot P_{xy} \cdot \frac{1}{\sqrt{\mu_y}}$ and so the eigenvalues of D and P are the same. Now note that $D_{xy} = \sqrt{P_{xy}P_{yx}} = \langle \psi_x^1 | \psi_y^2 \rangle$, which gives

$$\Pi_1\Pi_2 = \sum_{x,y} |\psi_x^1\rangle \langle \psi_x^1| \cdot |\psi_y^2\rangle \langle \psi_y^2| = \sum_{x,y} D_{xy} |\psi_x^1\rangle \langle \psi_y^2| .$$

The sets $\{|\psi_x^1\rangle\}_{x \in V}$ and $\{|\psi_y^2\rangle\}_{y \in V}$ are orthonormal bases which implies that the singular values of $\Pi_1\Pi_2$ are the eigenvalues of D. We now apply Jordan's lemma to relate the singular values of $\Pi_1\Pi_2$ with the eigenvalues of W, which gives the desired relationship between the eigenvalues of W and the eigenvalues of P.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 11

March 2, 2022 *Scribes: Liam McInroy*

Today: revisiting quantum walks, collision problem, Hamiltonians.

11.1 Revisiting Quantum Walks

Recall from Section 10.2 that we introduce random walks.

Revisiting Classical Random Walks

Let $G = (V, E)$ be an (undirected[1]) graph, with the *transition matrix* $P \in \mathbb{R}^{|V|\times|V|}$ given by

$$P_{xy} = \begin{cases} \frac{1}{d_x} & \text{if } (x, y) \in E \\ 0 & \text{else,} \end{cases} \tag{11.1}$$

where d_x is the degree of vertex $x \in E$. Then we can interpret P_{xy} as the probability of walking from vertex x to y, and we can form a *markov chain* over the state space of vertices V with the transition matrix P. For this particular example, the markov chain has a *stationary distribution* $\mu \in \mathbb{R}^{|V|}$ satisfying $\sum \mu_x P_{xy} = \mu_y$.

A common question in the study of markov chains is the *mixing time*, or the number of iterations of P that must be applied to converge to the stationary distribution. One example of why this might be useful, is that if we wanted to generate random samples of the vertices from the stationary distribution then we only need to randomly walk from any starting vertice given by the mixing time to result in a new randomly sampled vertice from the stationary distribution. So we would only need to store the local edges of the graph.

So the complexity measure we are interested in calculating is the mixing time, and this is possible for the random walk we've constructed by analyzing the *spectral gap* γ of P. Specifically, we may find an eigenvalue decomposition $P = B^{-1} \cdot \text{diag}(\lambda_1, \ldots) \cdot B$ for some invertible matrix $B \in \mathbb{R}^{|V|\times|V|}$ and eigenvalues $\lambda_1 > \lambda_2 > \lambda_3 > \ldots$. It's a general result of stochastic matrices that $\lambda_1 = 1$, and we define the spectral gap as

$$\gamma = \lambda_1 - \lambda_2 = 1 - \lambda_2. \tag{11.2}$$

It's also a general result that the mixing time is then inversely proportionally related to γ. For the exact statement, see Theorem 3 of this document.

[1] $(x, y) \in E$ if and only if $(y, x) \in E$

The Edge Process

One classical technique of simulating a classical random walk is the *edge process random walk*. This construction is useful for introducing quantum random walks, as it is a reversible analog of the classical random walk and thus more suitable for quantization.

We start with the original $G = (V, E)$ as given in the classical random walk scenario, but will perform a modified random walk on a newly constructed bipartite graph $G' = (V', E')$. Concretely, we have $V' = V_L \cup V_R$ where V_L and V_R are two distinct copies of V, one on the left and right side of the bipartite matching. The edges E' are defined by choice of $x \in V_L$ and $y \in V_R$ and create the edge $(x, y) \in E'$ if and only if $(x, y) \in E$ as considering x, y as elements of V. We end up with two edges of G' for each edge of G, if $(x, y) \in E$ then there is the "left-looking" edge (x, y) and the "right-looking" edge (y, x). Importantly, for two edges $(x, y), (x', y') \in E'$, then the edge (x', y') may be selected by "right-looking" from the edge (x, y) only if $x' = y$, and vice versa the edge (x, y) may be selected by "left-looking" from the edge (x', y') only if $y = x'$.

Then we may perform a random walk over the edges, rather than vertices, by alternating between randomly selecting the "left-looking" and "right-looking" edges. From the property noted above, then we end up with the identity that the conditional probability of selecting any e' from e is equal to the conditional probability of selecting any e from e'. This makes the process reversible (unlike the classical random walk on vertices where $P_{xy} \neq P_{yx}$ in general). We also obtain that the stationary distribution $\mu' \in \mathbb{R}^{|E'|}$ is uniform, and as there are twice as many edges in E' as E, then we result in the equivalent stationary distribuion over vertices of G as in the classical vertex random walk.

Revisiting Quantum Walks

The construction of the graph used in quantum walk is analogous to the construction of the edge process in classical random walks. However, naively quantization and phase estimation would suggest that we ought to reflect around reflect around some $|\mu\rangle$ of the stationary distribution and naively preparing $|\mu\rangle$ by computing μ would result in no speedup. Instead our aim is to construct an operator that behaves like reflection around $|\mu\rangle$ without computing $|\mu\rangle$ explicitly.

So let $G = (V, E)$ be a graph again, with P given as before. We introduce bases $\{|x\rangle : x \in V\}$ of a $|V|$-dimensional space and will operate in the space given by the bases $\{|x\rangle \otimes |y\rangle : x, y \in V\}$ and there is naturally a subspace $\mathcal{H} = \text{span}\{|x\rangle \otimes |y\rangle : (x, y) \in E\}$ which we will aim to constrain our operations to lie within.

Then we construct our unitary matrix W to represent the operator $2|\mu\rangle\langle\mu| - \mathbb{1}$ without preparing $|\mu\rangle$ via

$$W = \Big[\sum_x 2\left|\psi_x^1\right\rangle\!\left\langle\psi_x^1\right| - \mathbb{1}\Big]\Big[\sum_y 2\left|\psi_y^2\right\rangle\!\left\langle\psi_y^2\right| - \mathbb{1}\Big], \tag{11.3}$$

for the states $(|\psi_x^1\rangle)_{x\in V}$ and $(|\psi_y^2\rangle)_{y\in V}$ given by

$$\begin{aligned} |\psi_x^1\rangle &= \sum_y \sqrt{P_{xy}}\,|x\rangle \otimes |y\rangle\,, \\ |\psi_y^2\rangle &= \sum_x \sqrt{P_{yx}}\,|x\rangle \otimes |y\rangle\,. \end{aligned} \tag{11.4}$$

By construction, then we note that $\Pi_1 = \left|\psi_x^1\right\rangle\!\left\langle\psi_x^1\right|$ and $\Pi_2 = \left|\psi_y^2\right\rangle\!\left\langle\psi_y^2\right|$ satisfy

$$\Pi_1\Pi_2 = \sum_{x,y} \sqrt{P_{xy}P_{yx}} \left|\psi_x^1\right\rangle \left\langle\psi_y^2\right| = \sum_{x,y} \sqrt{\mu_x} P_{xy} \frac{1}{\sqrt{\mu_y}} \left|\psi_x^1\right\rangle \left\langle\psi_y^2\right|. \tag{11.5}$$

Thus we find that $\Pi_1\Pi_2$ shares the eigenvalues of P, and we claim these are also shared by the classical random walk on G induced by the iteration of W. Further, Jordan's lemma then allows us to relate the eigenvalues of W to those of $\Pi_1\Pi_2$ and P via the expression

$$\mathrm{Spec}(W) = \left\{e^{\pm 2i\theta_1}, e^{\pm 2i\theta_2}, \ldots, \right\}, \tag{11.6}$$

for the collection of (θ_i) such that $\cos\theta_i = \lambda_i$. In particular, we have that $\theta_1 = 0$ as λ_1 and thus

$$\cos\theta_2 = \lambda_2 = 1 - \gamma, \tag{11.7}$$

so the Taylor approximation $\cos\theta \approx 1 - \frac{\theta^2}{2}$ provides

$$\theta_2 \approx \sqrt{2\gamma}. \tag{11.8}$$

So if we consider the phase estimation PE_W of W such that

$$\mathrm{PE}_W \left|\phi\right\rangle \mapsto \begin{cases} \left|\phi\right\rangle & \text{if } \left|\phi\right\rangle = \left|\mu\right\rangle, \\ -\left|\phi\right\rangle & \text{else,} \end{cases} \tag{11.9}$$

then amplitude amplification tells us that we may approximate $2\left|\mu\right\rangle\!\left\langle\mu\right| - \mathbb{1}$ with $O\left(\frac{1}{\theta_2}\right) = O\left(\frac{1}{\sqrt{\gamma}}\right)$ applications of PE_W. Thus we end up with a quadratic speedup over a classical edge process.

11.2 Collision Problem

Let us now see a tangible example of random walks where quantum algorithms are superior. We'll look at the collision problem.

Recall in the Grover Problem that there is $f : \{0,1\}^n \to \{0,1\}^m$ for $m > n$, and we wish to determine if f is injective or not (if there is $x, y \in \{0,1\}^n$ with $f(x) = f(y)$, or a *collision*).

Classically, this requires $\theta\left(2^n\right)$ queries using a randomized algorithm.

In the quantum domain, Ambainis found $\theta\left(2^{\frac{2n}{3}}\right)$ queries of the oracle are necessary. Below is a brief description of both algorithms, and a back of the envelope calculation for the quantum improvement. This note contains more details.

To determine if there is a collision, we naively could check all pairs of inputs for collision. A simple generalization of this would be to compare all subsets of size R of the inputs, but we would be making many redundant queries. So instead, suppose we initially choose R inputs from $\{0,1\}^n$ and performing queries on this subset to check for collisions. Now, instead of re-sampling another randomly sampled R more inputs at the next iteration, we randomly choose only one of the already queried R inputs and replace it with a newly randomly sampled input.

This induces a random walk on the *Johnson graph* of $\{0,1\}^n$, the graph composed of vertices representing the various size R subsets of $\{0,1\}^n$ with between subsets if they differ in only one

element. We then mark subsets which contain distinct elements with non-unique images under f for search. We can characterize the expected number of steps to find a collision by the number of expected collisions, the expected time to reach a marked element (the mixing time), and so we may calculate the number of expected queries from these numbers.

It is a folklore result that the spectral gap of an R-Johnson graph is $\gamma = \frac{1}{R}$, and so we require $O\left(\frac{1}{\gamma}\right)$ steps to reach the stationary distribution. Similarly, the expected number of marked elements for search may be bounded by $\frac{N^2}{R^2}$, for $N = 2^n$. Classically, we may calculate the expected number of queries to find the collision,

$$R + 1\big(\frac{1}{\gamma}\big)\big(\frac{N^2}{R^2}\big) = R + \frac{N^2}{R} \to O(N), \tag{11.10}$$

Using the quantum speedup of the quantum random walk and search, we improve the mixing time,

$$R + 1\sqrt{\frac{1}{\gamma}}\sqrt{\frac{N^2}{R^2}} = R + \sqrt{R}\sqrt{\frac{N^2}{R^2}} = R + \frac{N}{\sqrt{R}} \to O\left(N^{\frac{2}{3}}\right). \tag{11.11}$$

11.3 Hamiltonian Complexity

We now will move onto Hamiltonian complexity.

For our purposes, think of a Hamiltonian as any Hermitian matrix. There is some more significant physical significance as representing the potential and kinetic energy of a system which motivates simulation of Hamiltonians.

Anyways, consider a Hamiltonian H in d-dimensions and consider its eigenvalue decomposition,

$$H = \sum_{i=1}^{d} E_i \left|\phi_i\right\rangle\!\left\langle\phi_i\right|, \tag{11.12}$$

for some non-descending eigenvalues $E_1 \leq E_2 \leq \ldots \leq E_d$ with corresponding eigenvectors $(\phi_i)_{i \leq d}$. There are some significant associations to a Hamiltonian,

- The *ground state* $|\phi_1\rangle\langle\phi_1|$,
- The *spectral gap* $\gamma = E_i - E_1$ for the least i such that $E_i \neq E_1$,
- The *Gibbs state* for the *inverse temperature* parameter β,

$$e^{-\beta H} = \frac{1}{\mathrm{Tr}\{e^{-\beta H}\}} \sum_{i=1}^{d} e^{-\beta E_i} \left|\phi_i\right\rangle\!\left\langle\phi_i\right|, \tag{11.13}$$

- The *expectation values* of the operators on the ground state of the Hamiltonian

$$\begin{aligned} \mu &= \mathrm{Tr}\{|\phi_1\rangle\langle\phi_1| \cdot \mu\}, \\ E_1 &= \mathrm{Tr}\{|\phi_1\rangle\langle\phi_1| \cdot H\}, \end{aligned} \tag{11.14}$$

- The *time evolution* of a state ρ via

$$\rho \mapsto e^{-iHt}\rho e^{iHt}, \tag{11.15}$$

Of some particular interest are *local Hamiltonians*, where we may decompose H as

$$H = \sum_{\alpha} b_\alpha P_\alpha, \tag{11.16}$$

where the collection of (P_α) are tensors of Pauli matrices. We denote a local Hamiltonian as k-local if each P_α has at most k non-identity Pauli matrices.

For instance, we can phrase 3SAT as a local Hamiltonian for which the ground state is a satisfying assignment.

We also may be interested in *Hamiltonian simulation*, for the computation of a matrix V which approximates the time evolution of a Hamiltonian, or precisely

$$V\rho V^\dagger \approx e^{-iHt}\rho e^{iHt}. \tag{11.17}$$

Hamiltonian simulation is in the class of QMA-hard complexity, and closely tied to the complexity of computing E_1. Which in turn connects to quantum walks to be covered in the next lecture.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 12

March 4, 2022 *Scribes: Tarun Prasad*

Today's lecture notes are on the following topics:

- Quantum proofs
- Recap of Cook-Levin theorem
- Quantum Cook-Levin theorem
- Feynman-Kitaev clock construction

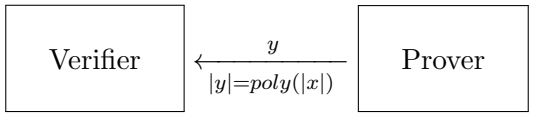
12.0 Classical Proof Systems and Quantum Proofs

Before we begin discussing quantum proofs, let us first recall some of the classical proof systems. In each of the following systems, L is a language, x is some string over the alphabet of L, and the goal is for a verifier to efficiently decide whether or not $x \in L$ with the help of a prover.

12.0.1 NP (Non-deterministic polynomial time)

In the NP protocol, the prover sends a proof y, whose length is polynomial in the length of x, and the verifier then runs a deterministic polynomial-time algorithm V that looks at both x and y, where V is such that

$$\begin{aligned} \text{(Completeness) } x \in L &\implies \exists y V(x,y) = 1 \\ \text{(Soundness) } x \notin L &\implies \forall y V(x,y) = 0. \end{aligned}$$



One example of a language L in NP is 3SAT, the set of all satisfiable boolean formulas consisting of collections of boolean variables arranged into clauses, where up to three variables (or their negations) are separated by ORs in each clause, and the clauses themselves are separated by ANDs, for instance,

$$(x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee \bar{x}_4 \vee x_5) \wedge \cdots .$$

The proof here is simply an assignment to the variables. If the formula is satisfiable, then there is an assignment that the prover can send to the verifier, who can then quickly check that it is valid. If it is not satisfiable, then no matter what proof the prover sends, it will be impossible for the verifier to satisfy all the clauses.

Not only is 3SAT *in* NP, it is also (informally) at least as hard as any problem in NP, as the Cook-Levin theorem states.

Theorem 12.0.1 (Cook-Levin)**.** *3SAT is NP-complete.*

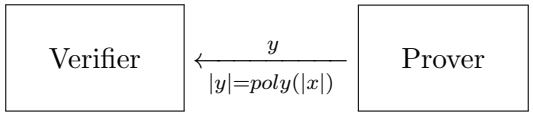
Proof. We have just shown that 3SAT is in NP; what remains to be shown is that 3SAT is NP-hard, i.e., any problem in NP can be reduced to it. Consider an arbitrary problem in NP: by definition, there must exist some polynomial-size verifier circuit V that takes the input $x = x_0x_1 \dots x_n$ and the proof $y = y_0y_1 \dots y_m$ and outputs 0 or 1 depending on whether the proof verifies the instance. To encode this circuit in a 3SAT instance, we will introduce the following boolean variables: $x_1, \dots, x_n, y_1 \dots, y_n$ representing (and enforced to be equal to) the inputs to the circuit, and z_i representing the output of each gate i. This can be enforced by adding constraints representing the computation of each gate; for example, if z_3 corresponds to the output of an OR gate that takes x_3 and x_4 as inputs, we add the constraint $(z_3 \leftrightarrow x_3 \vee x_4)$ (of course, after representing equivalences etc. in a manner compatible with 3SAT).

Adding such constraints for all gates, along with a final clause enforcing that the output bit is 1, will suffice to simulate the circuit: if there is a y that the circuit accepts, there must be a corresponding satisfying assignment to our 3SAT instance; if there is no such y, there can also be no satisfying assignment since its existence will enable us to backtrack through the computation and arrive at a valid y. □

12.0.2 MA (Merlin-Arthur)

The MA protocol is similar to the NP protocol, except in that the verifier is allowed to use randomness. As before, the prover sends a proof y, whose length is polynomial in the length of x, and the verifier then runs a *randomized* polynomial-time algorithm V that looks at both x and y. In other words, the verifier is allowed to toss coins and use the results of the coin tosses (say r) in its algorithm. It is important to note here that the proof y may depend on x but not on r, i.e., the prover does not get to see the private randomness of the verifier. The requirement in this case is that

$$\begin{aligned} \text{(Completeness) } x \in L &\implies \exists y \Pr_r[V(x,y,r) = 1] = 1^1 \\ \text{(Soundness) } x \notin L &\implies \forall y \Pr_r[V(x,y,r) = 1] \leq e^{-\Theta(|x|)}. \end{aligned}$$



What would be an example of an MA-complete problem (i.e. the analog of 3SAT for MA)? As it turns out, there *was* no known example until one was identified (Bravyi & Terhal 2009) within the field of quantum complexity! The Cook-Levin theorem fails here simply because we do not know of a way to enforce that the verifier started with a random coin in our reduction to 3SAT.

12.0.3 QMA (Quantum Merlin-Arthur)

By abuse of terminology, many refer to this class as Quantum NP because they think of it as a quantum analog of NP, but really it is a quantum analog of MA because we allow some small

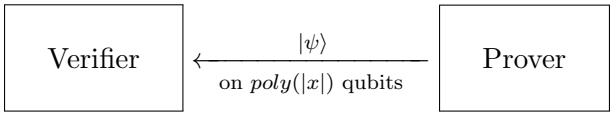
[1] This probability can equivalently be specified as being at least $1 - e^{-\Theta(|x|)}$.

probability of error. This time, the verifier has a quantum computer, i.e., a quantum circuit with polynomially many gates that takes in the input x along with a quantum state $|\psi\rangle$ (on polynomially many qubits) as proof from the prover. The verifier runs its quantum circuit and finally chooses one of its output qubits (say O) and performs a measurement in the $\{|0\rangle, |1\rangle\}$ basis: we shall call this measurement outcome the output of the algorithm. Here, we require that

$$\begin{aligned}\text{(Completeness) } x \in L &\implies \exists\, |\psi\rangle \Pr[V(x, |\psi\rangle) = |1\rangle] \geq 1 - e^{-\Theta(|x|)} \\ \text{(Soundness) } x \notin L &\implies \forall\, |\psi\rangle \Pr[V(x, |\psi\rangle) = |1\rangle] \leq e^{-\Theta(|x|)}.\end{aligned}$$

As a reminder, the mathematical expression for the above probability is:

$$\begin{aligned}\Pr[V(x, |\psi\rangle) = 1] &= \mathrm{Tr}\left[(|1\rangle\langle 1|_O \otimes \mathbb{1})V\,|x\rangle\langle x| \otimes |\psi\rangle\langle\psi|\,V^\dagger\right] \\ &= \langle x| \otimes \langle\psi|\,V^\dagger(|1\rangle\langle 1|_O \otimes \mathbb{1})V\,|x\rangle \otimes |\psi\rangle\,.\end{aligned}$$



12.1 Quantum Cook-Levin Theorem

Definition 12.1.1. The 5-local Hamiltonian problem 5-LHP$_{a,b}$ is defined as follows: given a 5-local Hamiltonian $H = \sum_\alpha q_\alpha P_\alpha$, decide whether

1. $E_1(H) \leq a$, or
2. $E_1(H) \geq b$,

where $E_1(H)$ is the ground state energy of H and $a \leq b$.

Recall from Section 11.3 that this problem is a generalization of 3SAT: setting $a = 0$ and $b = 1/2$ after constructing our LHP instance will suffice because a 3SAT instance will be satisfiable if and only if the corresponding instance of 3-LHP$_{0,1/2}$ has 0 as its minimum eigenvalue. The quantum analog of 3SAT being NP-complete is the following theorem.

Theorem 12.1.2 (Quantum Cook-Levin, Kitaev 1999)**.** *The 5-local Hamiltonian problem 5-LHP$_{a,b}$ is QMA-complete for inverse-polynomial gap $b - a$.*

Proof. First we will show that 5-LHP$_{a,b}$ is in QMA for $b - a = \Omega\left(\frac{1}{poly(|x|)}\right)$. A simple way to do this is for the prover to send the ground state $|\phi_1\rangle$ as proof. Recall that $H\,|\phi_1\rangle = E_1\,|\phi_1\rangle$, and so,

$$E_1 = \mathrm{Tr}[H\,|\phi_1\rangle\langle\phi_1|] = \sum_\alpha q_\alpha \,\mathrm{Tr}[P_\alpha\,|\phi_1\rangle\langle\phi_1|].$$

So all the verifier needs to do now is to compute the expectation values $\mathrm{Tr}[P_\alpha\,|\phi_1\rangle\langle\phi_1|]$ for all α, from which it can compute E_1 and check whether $E_1 \leq a$ or $E_1 \geq b$. How can it compute these expectation values? One way of doing so is to instead compute an estimate by having the prover send multiple copies of $|\phi_1\rangle$ and calculating the empirical average (using Chernoff bounds to show that this is close to the true expected value).

Another method is through a Hamiltonian simulation. Implementing and applying the unitary e^{iHt}, we notice that

$$e^{iHt} |\phi_1\rangle = e^{iE_1 t} |\phi_1\rangle ,$$

which follows from the fact that $H^k |\phi_1\rangle = E_1^k |\phi_1\rangle$. In other words, the phase of the unitary has information about the eigenvalue E_1 and we can obtain this to the required accuracy by running the phase estimation algorithm.

The second part of the quantum Cook-Levin proof is showing that 5-LHP$_{a,b}$ is QMA-hard for $a = e^{-\Theta(|x|)}$ and $b = \frac{1}{c|x|^3}$. This requires showing that any problem in QMA can be reduced to a local Hamiltonian problem. To do so, let us try to follow the same idea as in the classical Cook-Levin theorem (see proof of Theorem 12.0.1). Analogously define $|x\rangle = |x_0 x_1 \dots x_n\rangle$ to be the input, $|\phi\rangle = |\phi_0 \phi_1 \dots \phi_m\rangle$ to be the proof, and $|\theta_{i+1}\rangle$ to be the output after each "layer" of quantum gates U_i. We want to enforce

$$|\theta_{i+1}\rangle = U_i |\theta_i\rangle$$

for all i. But how do we enforce this constraint using local Hamiltonians?

Consider the following example: say after 100 steps of the computation, we arrive at the following state, called a *cat state*:

$$|\theta_{100}\rangle = \frac{1}{\sqrt{2}} |00\cdots0\rangle + \frac{1}{\sqrt{2}} |11\cdots1\rangle .$$

Supposing $U_{100} = \mathbb{1}$, we would also have

$$|\theta_{101}\rangle = |\theta_{100}\rangle .$$

Now suppose we compute the reduced density matrix of $|\theta_{101}\rangle$ on 2 qubits:

$$\begin{aligned}
&\mathrm{Tr}_{\text{all but 2 qubits}}[|\theta_{101}\rangle \langle\theta_{101}|] \\
&= \mathrm{Tr}_{\text{all but 2 qubits}} \left[\frac{1}{2} |0\cdots0\rangle\langle0\cdots0| + \frac{1}{2} |1\cdots1\rangle\langle1\cdots1| + \frac{1}{2} |0\cdots0\rangle \langle1\cdots1| + \frac{1}{2} |1\cdots1\rangle \langle0\cdots0| \right] \\
&= \frac{1}{2} |00\rangle\langle00| + \frac{1}{2} |11\rangle\langle11| .
\end{aligned}$$

What we end up with is two perfectly-correlated qubits (and the same applies for any constant number of qubits). Now consider a different quantum state $|\theta'_{101}\rangle$ (which can be obtained by simply applying a Z on $|\theta_{100}\rangle$),

$$|\theta'_{101}\rangle = \frac{1}{\sqrt{2}} |00\cdots0\rangle - \frac{1}{\sqrt{2}} |11\cdots1\rangle ,$$

and trace out all but 2 qubits:

$$\begin{aligned}
&\mathrm{Tr}_{\text{all but 2 qubits}}[|\theta'_{101}\rangle \langle\theta'_{101}|] \\
&= \mathrm{Tr}_{\text{all but 2 qubits}} \left[\frac{1}{2} |0\cdots0\rangle\langle0\cdots0| + \frac{1}{2} |1\cdots1\rangle\langle1\cdots1| - \frac{1}{2} |0\cdots0\rangle \langle1\cdots1| - \frac{1}{2} |1\cdots1\rangle \langle0\cdots0| \right] \\
&= \frac{1}{2} |00\rangle\langle00| + \frac{1}{2} |11\rangle\langle11| \\
&= \mathrm{Tr}_{\text{all but 2 qubits}}[|\theta_{101}\rangle \langle\theta_{101}|].
\end{aligned}$$

What this shows is that even though $|\theta_{101}\rangle$ and $|\theta'_{101}\rangle$ are different quantum states, *locally* they look exactly the same! So there is no local Hamiltonian instance that accepts one but rejects the other - this is called *local indistinguishability* and shows that we cannot directly adapt the proof of the classical Cook-Levin theorem to the quantum case. The way we can resolve this is using the Feynman-Kitaev clock construction.

□

12.2 Feynman-Kitaev Clock Construction

To solve our problem from before, introduce a new register C (the "clock" register) and define

$$|\zeta\rangle = \frac{1}{\sqrt{2}} \left(|\theta_{100}\rangle \otimes |1\rangle_C + |\theta_{101}\rangle \otimes |2\rangle_C \right).$$

If $|\theta_{100}\rangle = |\theta_{101}\rangle$, this simplifies to

$$|\zeta\rangle = \frac{1}{\sqrt{2}} |\theta_{100}\rangle \otimes (|1\rangle + |2\rangle)_C.$$

Also define a Hamiltonian

$$H = \mathbb{1} \otimes (|1\rangle - |2\rangle)(\langle 1| - \langle 2|)_C.$$

What is the energy of $|\zeta\rangle$ with respect to H? Notice that $(|1\rangle + |2\rangle)$ is orthogonal to $(|1\rangle - |2\rangle)$, so the energy is zero!

$$\langle \zeta | H | \zeta \rangle = 0.$$

But in general,

$$\langle \zeta | H | \zeta \rangle = \frac{1 - \langle \theta_{100} | \theta_{101} \rangle}{2},$$

which means that the energy of $|\zeta\rangle$ is 0 when $|\theta_{100}\rangle$ and $|\theta_{101}\rangle$ are the same, and higher when they are different. Importantly, the Hamiltonian H only acted non-trivially on the one qubit in the C register; this is how we can *locally* check that the quantum state did not change.

Now we are ready to describe the general Feynman-Kitaev clock construction. Given a circuit of T steps of computation, say,

$$U_T U_{T-1} \cdots U_1 |x\rangle \otimes |\psi\rangle,$$

the construction is the following history state as an analog of $|\zeta\rangle$:

$$|history\ state\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} U_t U_{t-1} \cdots U_1 |x\rangle \otimes |\psi\rangle \otimes |t\rangle_C.$$

The Hamiltonian can now be defined as:

$$H = H_{init} + H_{propagation} + H_{check},$$

where

$$H_{init} = |0\rangle\langle 0|_C \otimes \sum_i |\bar{x}_i\rangle\langle \bar{x}_i|$$

enforces that we have the right input at time step 0 (here, $\bar{x}_i = 1 - x_i$), and

$$H_{check} = |T\rangle\langle T|_C \otimes |0\rangle\langle 0|_O$$

enforces that the output qubit is *not* in the 0 state. We will discuss the $H_{propagation}$ Hamiltonian, along with a completeness and soundness proof, in the next lecture.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 13

March 9, 2022 *Scribes: Chi-Ning Chou*

Today's lecture notes are on the following topics:

- Clock construction.
- Applications.
- Examples.
- Proofs.

13.1 Clock Construction

Recall that in the last lecture we wanted to show that the 5-local Hamiltonian problem is QMA-hard. And we ended up with a glimpse into the *Feynman-Kitaev clock construction*, which is a key step for proving this theorem.

13.1.1 History states

Given a quantum circuits with gates $U_1, U_2, \ldots, U_T$ and an input state $|in\rangle$, the history state is defined as

$$|history\rangle := \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |t\rangle_C \otimes U_t U_{t-1} \cdots U_t |in\rangle \tag{13.1}$$

where $U_0 = I$ is used here for notational simplicity and $|\cdot\rangle_C$ stands for the "clock register"[1]. We are going to see how $|history\rangle$ can be treated as a quantum proof for faithfully executing the circuit. But before that, let's take a step back to see how QMA and other related complexity classes fit into the notations we are using here. In particular, we will see what are the corresponding circuits $U = U_T \cdots U_1$ and input states $|in\rangle$.

- (QMA) Given a QMA problem, there will be a verifier algorithm V which takes the input $x \in \{0,1\}^n$ of the problem, $m = \mathsf{poly}(n)$ ancilla qubits, a quantum state $|\psi\rangle$ (which serves as a proof/witness). So we naturally pick the circuit U to be the verifier circuit V and
$$|in\rangle = |x\rangle \otimes |\psi\rangle \otimes |0\rangle^{\otimes m} .$$
- (BQP) Given a BQP problem, there will be a quantum circuit C which takes the input $x \in \{0,1\}^n$ of the problem and $m = \mathsf{poly}(n)$ ancilla qubits. So we naturally pick U to be C and
$$|in\rangle = |x\rangle \otimes |0\rangle^{\otimes m} .$$

[1] There's a tricky issue about how to properly encode the time to make sure the Hamiltonian can be made local, but in this lecture for simplicity we will ignore that.

- (MA) Given a MA problem there will be a classical circuit C which takes the input $x \in \{0,1\}^n$ of the problem, some uniform random bits $r \in \{0,1\}^\ell$, and a classical proof $w \in \{0,1\}^k$. Here we can use multiple EPR pairs to prepare the random bits[2]. Concretely, we can pick U to be the quantum version of C and

$$|in\rangle = |x\rangle \otimes \left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right)^{\otimes \ell} \otimes |w\rangle \otimes |0\rangle^{\otimes m} .$$

  where the extra $|0\rangle^{\otimes m}$ is the ancilla qubits that might appear when we quantize C.

13.1.2 Clock Hamiltonian

Given a QMA problem with verifier circuit U, an input $x \in \{0,1\}^n$, and a proof state $|\psi\rangle$, our goal is to construct (in polynomial time) a local Hamiltonian H_{clock} with the following properties:

- (completeness) if $\Pr[U(x, |\psi\rangle)] > 1 - e^{-\Theta(n)}$ (i.e., x is a Yes instance), then

$$0 \leq E_1(H_{clock}) \leq \mathrm{tr}(|history\rangle\langle history| \, H_{clock}) < e^{-\Theta(n)}$$

  i.e., the ground state energy of H_{clock} is inverse exponentially small.

- (soundness) if $\Pr[U(x, |\psi\rangle)] < e^{-\Theta(n)}$ (i.e., x is a No instance), then

$$E_1(H_{clock}) = \Omega\left(\frac{1}{T^3}\right)$$

  where T is the running time of U. Namely, the ground state energy of H_{clock} is at least inverse polynomially large.

Note that if we can show the above two properties, we essentially show that local Hamiltonian is QMA-hard as desired.

In the rest of this lecture, we will first have a quick digression on the applications of the ideas used in the proof to other problems in quantum complexity and quantum computation. Then, we will present the full construction of H_{clock} and provide a proof sketch.

13.2 Applications

There are several applications of the clock construction beyond its original motivation in proving the QMA-hardness of local Hamiltonian problem:

- *Adiabatic quantum computing* is a model for realizing quantum computation. Instead of applying quantum gates, we let the quantum system to evolve by a given Hamiltonian specified according to the input of the computational problem. Through the clock construction, one can then transform any BQP algorithm into a Hamiltonian and set up an adiabatic quantum computing system.

[2]Concretely, we use the first qubit of an EPR pair as a classical random bit and leave the second qubit untouched. At the end when partial trace out the the second qubit, the first qubit behaves exactly as a uniform classical bit.

- *Verification of quantum computing* is an important and non-trivial task since a direct classical simulation of a quantum algorithm has exponential overhead in time. Recent breakthrough in classical verification of quantum computing by Mahadev [Mah18] used the clock construction.
- *Volume law entanglement:* Understanding the ground states of local hamiltonians is an old pursuit in physics. The amount of entanglement is a standard measure of the complexity of the ground state. QMA completeness local hamiltonians exhibit high amount of entanglement (called volume law entanglement), as shown in [GH10, Ira10, AHL$^+$14]
- *Zero-knowledge protocol:* Clock construction is crucial in the known zero-knowledge protocols for QMA [BG20].

13.3 Construction

Let's now take a look at the details of the clock construction. Recall that we want two things for our final Hamiltonian H_{clock}: (i) when the circuit U accepts $|in\rangle$, the history state $|history\rangle$ should have small energy w.r.t. H_{clock}; (ii) when the circuit U rejects $|in\rangle$, the ground state energy of H_{clock} should be large.

To achieve (i), we would like to construct some local Hamiltonian to guarantee a few things (i-a) check if $|history\rangle$ is indeed built for the input x; (i-b) check if $|history\rangle$ indeed encodes the all the intermediate computational steps on $U|in\rangle$; (i-c) check if U indeed accepts $|in\rangle$. As previewed in the previous lecture, we will do this by constructing three local Hamiltonian H_{init}, H_{prop}, H_{check} for (i-a), (i-b), and (i-c) respectively and let

$$H_{clock} = H_{init} + H_{prop} + H_{out}\,.$$

As a remark, the Hamiltonian H_{clock} can only depend on x and U and cannot use the proof $|\psi\rangle$. Nevertheless, the history state $|history\rangle$ can use $|\psi\rangle$.

The following table summarizes the high-level picture of the reduction from any QMA problem L to the local Hamiltonian problem.

	A QMA problem L with circuit U	Local Hamiltonian Problem
Input	x	H_{clock}
Proof	$\|\psi\rangle$	$\|history\rangle$
Yes	$\exists\,\|\psi\rangle\,,\ \Pr[U \text{ accepts } (\|x\rangle\,,\|\psi\rangle)] > 1 - e^{-\Theta(n)}$	$E_1(H_{clock}) < e^{-\Theta(n)}$
No	$\forall\,\|\psi\rangle\,,\ \Pr[U \text{ accepts } (\|x\rangle\,,\|\psi\rangle)] < e^{-\Theta(n)}$	$E_1(H_{clock}) = \Omega\left(\frac{1}{T^3}\right)$

13.3.1 H_{prop}

For (i-b), we would like to construct a local Hamiltonian H_{prop} so that $|history\rangle$ is a ground state. Note that we cal always easily construct a Hamiltonian using the whole circuit U so that $|history\rangle$ is a ground state, but such a straightforward construction won't be local. So the key is really about how to build several local Hamiltonians that locally verify the computation at time $t+1$ from the computation at time t.

Step 1: Trivial circuit U. But anyway let's take a step back and first think about the base case where $T = 1$ and $U = I$ is the identity circuit. How to build a Hamiltonian so that $|history\rangle = \frac{1}{\sqrt{2}}(|0\rangle_C \otimes |in\rangle + |1\rangle_C \otimes |in\rangle)$ is the ground state? The following local Hamiltonian turns out to be sufficient!

$$H_0 = (|0\rangle - |1\rangle)(\langle 0| - \langle 1|)_C \otimes I\,. \tag{13.2}$$

The reason is that the energy of H_0 on $\frac{1}{\sqrt{2}}(|0\rangle_C \otimes |in\rangle + |1\rangle_C \otimes |\theta\rangle)$ is $1 - \langle in|\theta\rangle$ for any state $|\theta\rangle$. It will be a good exercise for you to verify this claim and see the footnote for a proof [3].

Step 2: Single gate circuit U. Next, let's consider the case where U contains a single gate u, i.e., $U = u \times I$, so we have $|history\rangle = \frac{1}{\sqrt{2}}(|0\rangle_C \otimes |in\rangle + |1\rangle_C \otimes U\,|in\rangle)$. How to build a local Hamiltonian H_1 so that the above $|history\rangle$ is the ground state? Note that we can first rewrite $|history\rangle$ as follows.

$$|history\rangle = W\left[\frac{1}{\sqrt{2}}\left(|0\rangle_C \otimes |in\rangle + |1\rangle_C \otimes |in\rangle\right)\right]$$

where $W = |0\rangle\langle 0|_C \otimes I + |1\rangle\langle 1|_C \otimes U$ is a unitary matrix. Thus, the following choice of H_1 suffices!

$$H_1 = W H_0 W^\dagger$$

where H_0 is from Equation 13.2. It's a good exercise to check why $|history\rangle$ is the ground state of H_1 and see the footnote for a proof[4]. Let's further open up H_1 as follows.

$$\begin{aligned} H_1 &= W H_0 W^\dagger \\ &= W[(|0\rangle - |1\rangle)(\langle 0| - \langle 1|)_C \otimes I]W^\dagger \\ &= |0\rangle\langle 0|_C \otimes I + |1\rangle\langle 1|_C \otimes I - |0\rangle\langle 1|_C \otimes U^\dagger - |1\rangle\langle 0|_C \otimes U\,. \end{aligned} \tag{13.3}$$

Note that Equation 13.3 is quite "interpretable" in the sense that we can think of H_1 contains four "paths": $0 \to 0$, $1 \to 1$, $0 \to 1$, and $1 \to 0$ where the first two has "weight" I, the third one has weight $U^\dagger$, and the last one has weight U. In particular, each "weight" is a local Hamiltonian.

Step 3: General T gates circuit U. Finally, when U contains of gates $u_1, u_2, \ldots, u_T$, the history state becomes

$$|history\rangle = \frac{1}{\sqrt{T+1}}\left(|0\rangle_C \otimes |in\rangle + \sum_{t=1}^{T} |t\rangle_C \otimes (U_t U_{t-1} \cdots U_1)\,|in\rangle\right)$$

where $U_t = u_t \otimes I$. Now the question is, how to construct a local Hamiltonian H_T so that the above $|history\rangle$ is its ground state? Given Equation 13.3 and its "interpretation", it is natural to pick the following Hamiltonian.

$$H_T = \sum_{t=1}^{T}\left[|t-1\rangle\langle t-1|_C \otimes I + |t\rangle\langle t|_C \otimes I - |t-1\rangle\langle t|_C \otimes U_t^\dagger - |t\rangle\langle t-1|_C \otimes U_t\right]. \tag{13.4}$$

[3] Let $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle_C \otimes |in\rangle + |1\rangle_C \otimes |\theta\rangle)$, the energy of H on $|\psi\rangle$ is $\langle\psi|\,H\,|\psi\rangle = \langle\psi|\,(|0\rangle\langle 0| \otimes I)\,|\psi\rangle + \langle\psi|\,(|1\rangle\langle 1| \otimes I)\,|\psi\rangle - \langle\psi|\,(|0\rangle\langle 1| \otimes I)\,|\psi\rangle - \langle\psi|\,(|1\rangle\langle 0| \otimes I)\,|\psi\rangle = \frac{1}{2}(1 + 1 - \langle in|\theta\rangle - \langle in|\theta\rangle) = 1 - \langle in|\theta\rangle$.

[4] From Step 1, we know that $H_0\,|\psi\rangle = 0$ where $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle_C \otimes |in\rangle + |1\rangle_C \otimes |in\rangle)$. Note that $|history\rangle = W\,|\psi\rangle$. As W is unitary, we have $H_1\,|history\rangle = W H_0 W^\dagger\,|history\rangle = W H_0 W^\dagger W\,|\psi\rangle = W H_0\,|\psi\rangle = 0$.

Note that we literally replace $0 \to t-1$, $1 \to t$, and $U \to U_t$ from Equation 13.3 for each t and sum them up.

So now if we pick $H_{prop} = H_T$ where H_T is from Equation 13.4, we will have the history state $|history\rangle$ from Equation 13.1 being its ground state **for any** input state $|in\rangle$ (note that the choice of $|history\rangle$ depends on $|in\rangle$). This seems good but wait a second, if the verifier gives us a history state $|history'\rangle$ that does not have anything to do with our input state $|in\rangle$, $|history'\rangle$ will still be the ground state of H_{prop} and hence we will always accept it! Namely, we need to make sure the history state indeed encodes the computation history of $|in\rangle$ and this can be handled by H_{init} in the next step.

13.3.2 H_{init}

Recall that the input state for a QMA problem is of the form

$$|in\rangle = |x\rangle \otimes |\psi\rangle \otimes |0\rangle^{\otimes m}$$

where $x \in \{0,1\}^n$ is the classical input the the problem, $|\psi\rangle$ is a quantum proof, and $|0\rangle^{\otimes m}$ is the ancilla qubits. Note that $|x\rangle$ and $|0\rangle^{\otimes m}$ are the part that have to be fixed, i.e., we want to make sure that the $|in\rangle$ in the history state $|history\rangle$ sent by the quantum prover has its first part exactly being $|x\rangle$ and the third part exactly being $|0\rangle^{\otimes m}$.

Let's start with a simpler task: how to construct a local Hamiltonian H_x so that x is its ground state? The following naturally works!

$$H_x = \sum_{i=1}^{n} |\bar{x_i}\rangle\langle\bar{x_i}|$$

where $\bar{x_i} = 1 - x_i$. The reason why we have to negate/complement each bit is that here a ground state *minimizes* the energy.

So to check the validity of $|in\rangle$ via local Hamiltonian, we simply construct a local Hamiltonian

$$H_{init} = \sum_{i=1}^{n} |\bar{x_i}\rangle\langle\bar{x_i}| \otimes I + \sum_{j=1}^{m} |1\rangle\langle 1| \otimes I$$

where the first sum makes sure the classical input register is $|x\rangle$ and the second sum makes sure the ancilla register contains $|0\rangle^{\otimes m}$.

13.3.3 H_{out}

Note that $H_{input} + H_{prop}$ has history states of the following form as ground states

$$\frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |t\rangle_C \otimes U_t U_{t-1} \cdots U_t(|x\rangle \otimes |\psi\rangle \otimes |0\rangle^{\otimes m})$$

for every $|\psi\rangle$. Namely, the subspace of the ground states of $H_{input} + H_{prop}$ is captured by the following projective operator

$$\Pi_{gs} = \mathsf{span}\left\{ \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |t\rangle_C \otimes U_t U_{t-1} \cdots U_t(|x\rangle \otimes |\psi\rangle \otimes |0\rangle^{\otimes m}) \,:\, \forall\, |\rangle\, \psi \right\}.$$

Finally, we would like to construct the last local Hamiltonian H_{check} to check whether the QMA circuit U indeed accepts $|x\rangle$ and $|\psi\rangle$. As we already have $|T\rangle_C \otimes U_T \cdots U_1 |in\rangle$ inside $|history\rangle$, we can simply "read out" the output probability of U evaluating on $(|x\rangle, |\psi\rangle)$ using the following Hamiltonian.

$$H_{output} = |T\rangle\langle T|_C \otimes |0\rangle\langle 0|_O \otimes I$$

where O stands for the output register, i.e., without loss of generality being the first output qubit of U.

In summary, we have constructed local Hamiltonians H_{init}, H_{prop}, H_{output} and we will let

$$H_{clock} = H_{init} + H_{prop} + H_{output}$$

to be our final local Hamiltonian.

13.4 Proof Sketch

The completeness part of the theorem is straightforward: throughout the construction we have guaranteed $|history\rangle$ to be the ground state of H_{init}, H_{prop}, and H_{output} as long as x is a Yes instance and $|\psi\rangle$ is a valid proof. To see the ground state energy is exponentially small, observe that the only "contribution to the energy" comes from the error probability of U accepting $(|x\rangle, |\psi\rangle)$. As the error probability is exponentially small, the ground state energy is also exponentially small. We will go into more details in the next lecture.

The soundness of the theorem is the difficult part. The analysis involves quantum random walks and Jordan lemma. We will give a proof sketch in the next lecture.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 14

March 11, 2022 *Scribe: Xiaomin Li*

Today's lecture notes are on the following topics:

- Final analysis of clock construction
- Marriott Watrous protocol (Zombie story that ends well)

14.1 Final Analysis of Clock Construction

14.1.1 Review

Recall from last lecture that we defined the history state:

$$|history\rangle := \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |t\rangle_C \otimes U_t U_{t-1} \cdots U_0 |in\rangle \tag{14.1}$$

where $U_0 := I$. For a language L in the class QMA, write $|in\rangle = |x\rangle \otimes |\psi\rangle$, where x is the string whose membership in L is to be decided and $|\psi\rangle$ is the associated witness of membership. We also constructed the clock Hamiltonian:

$$H_{clock} = H_{in} + H_{prop} + H_{out},$$

where

$$H_{in} = |0\rangle\langle 0|_C \otimes \sum_{i=1}^{n} |\bar{x}_i\rangle\langle \bar{x}_i| \quad (\text{where } n = |x|) \tag{14.2}$$

$$H_{prop} = \sum_{t=1}^{T} \left[|t\rangle\langle t|_C \otimes I + |t-1\rangle\langle t-1|_C \otimes I - |t\rangle |t-1\rangle_C \otimes U_t - |t-1\rangle \langle t|_C \otimes U_t^\dagger \right] \tag{14.3}$$

$$H_{out} = |T\rangle\langle T|_C \otimes |0\rangle\langle 0|_O \,. \tag{14.4}$$

and the ground states of $H_{in} + H_{prop}$ is the following projective operator

$$\Pi_{gs} \text{ projects on } \mathsf{span} \left\{ \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} |t\rangle_C \otimes U_t U_{t-1} \cdots U_t (|x\rangle \otimes |\psi\rangle) \,:\, \forall\, |\psi\rangle \right\}. \tag{14.5}$$

For the details of the construction above, please refer back to the scribe notes for Lecture 13. Recall that we want to show that local Hamiltonian is QMA-hard. Hence we would like to prove the following theorem.

Theorem 14.1.1. *If there exists $|\psi\rangle$ such that $\mathbb{P}[accept] \geq 1 - e^{-n}$ (in this case, $x \in L$), then $E_1(H_{clock}) \leq O(\frac{e^{-n}}{T+1})$ (and the history state achieves this). If for all $|\psi\rangle$ we have $\mathbb{P}[accept] \leq e^{-n}$ (in this case, $x \notin L$), then $E_1(H_{clock}) = \Omega(\frac{1}{T^3})$.*

14.1.2 Prove Completeness

In the case $x \in L$, we focus on the first claim (completeness) in the theorem. Note that in our construction, $|history\rangle$ is the ground state of $H_{in} + H_{prop}$. When we apply H_{out} on $|history\rangle$, we get the projection on the part of $|history\rangle$ which rejects. In the assumption of the first claim of the theorem, we have that the acceptance probability is at least $1 - e^{-n}$, so the rejection probability is at most $1 - (1 - e^{-n}) = e^{-n}$. This implies that

$$\begin{aligned}\langle history|\, H_{clock}\, |history\rangle &= \langle history|\, H_{in} + H_{prop}\, |history\rangle + \langle history|\, H_{out}\, |history\rangle \\ &\leq 0 + \frac{1}{(\sqrt{T+1})^2} \cdot e^{-n} \\ &= O(\frac{e^{-n}}{T+1}).\end{aligned}$$

This finishes the proof for completeness. Next, we focus on the proof of the second claim (soundness).

14.1.3 Prove Soundness

The proof of soundness consists of two parts. We show in Lemma 14.1.2 below that $H_{in} + H_{prop} \geq \Omega(\frac{1}{T^2})(I - \Pi_{gs})$. (Recall from (14.5) that the Hamiltonian Π_{gs} corresponds to the ground space (the zero eigenvalue subspace), so $I - \Pi_{gs}$ corresponds to the nonzero eigenvalues subspace.) Then in Lemma 14.1.3 below, we claim that $H_{out} + (I - \Pi_{gs}) \succeq \Omega(\frac{1}{T})I$. Combining these two lemmas implies that

$$\begin{aligned}H_{clock} = H_{in} + H_{prop} + H_{out} &\succeq \Omega(\frac{1}{T^2})(I - \Pi_{gs}) + H_{out} \quad \text{(by Lemma 14.1.2)} \\ &\succeq \Omega(\frac{1}{T^2})(I - \Pi_{gs} + H_{out}) \quad (\text{since } 1 \geq \frac{1}{T^2}) \\ &\succeq \Omega(\frac{1}{T^3})I \quad \text{(by Lemma 14.1.3).}\end{aligned}$$

This implies $E_1(H_{clock}) = \Omega(\frac{1}{T^3})$, which completes the proof of the theorem. Therefore, it suffcies to prove the two lemmas below. We only provide a brief sketch of the proofs. One can find the details in the survey by Aharonov and Naveh [AN02].

Lemma 14.1.2. $H_{in} + H_{prop} \succeq \Omega(\frac{1}{T^2})(I - \Pi_{gs})$

Proof sketch. Let $H_{in}+H_{prop}$ have eigen-decomposition $Q\Lambda_1 Q^{-1}$ and $I-\Pi_{gs}$ have eigen-decomposition $Q\Lambda_2 Q^{-1}$. Denote the eigenvalues of $H_{in} + H_{prop}$ in ascending order: $\lambda_0, \lambda_1, \lambda_2, \ldots$ ($\lambda_0 = 0$ and λ_1 is the second smallest eigenvalue). In matrix forms, we have

$$\Lambda_1 = \begin{bmatrix} \ddots & & & & \\ & \lambda_3 & & & \\ & & \lambda_2 & & \\ & & & \lambda_1 & \\ & & & & 0 \end{bmatrix}, \quad \Lambda_2 = \begin{bmatrix} \ddots & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix}.$$

Then

$$\Lambda_1 \succeq \begin{bmatrix} \ddots & & & & \\ & \lambda_1 & & & \\ & & \lambda_1 & & \\ & & & \lambda_1 & \\ & & & & 0 \end{bmatrix} = \lambda_1 \begin{bmatrix} \ddots & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix} = \lambda_1 (I - \Pi_{gs})$$

Now it suffices to show that $\lambda_1 = \Omega(\frac{1}{T^2})$, where λ_1 is the second eigenvalue of $H_{in} + H_{prop}$.

The hard part of the remainder of the proof is to study the eigenvalues of H_{prop}, where we will use the theory of random walks. We apply rotations on H_{prop} because the eigenvalues will not change if we look at H_{prop} in a rotated basis. That is, the eigenvalues $R^\dagger H_{prop} R$ has the same eigenvalues with H_{prop}, for a rotation matrix R. Particularly, we choose to define R as

$$R := \sum_{t=0}^{T} U_t \cdots U_1 U_0 \otimes |t\rangle\langle t| \,,$$

then it gives

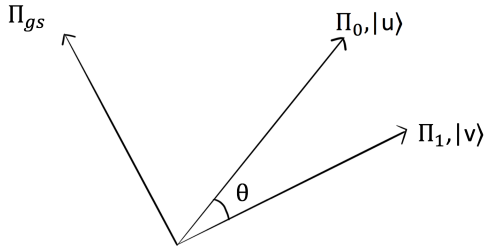
$$R^\dagger H_{prop} R = \sum_{t=1}^{T} (|t\rangle\langle t|_C + |t-1\rangle\langle t-1|_C - |t\rangle\langle t-1|_C - |t-1\rangle\langle t|_C) \otimes I,$$

and the terms inside the sum are related to the stochastic matrix corresponding to a simple random walk from 0 to T with loops on the start and end points. Then we can apply the related theory to finish the analysis of the eigenvalues of H_{prop}. □

Lemma 14.1.3. $H_{out} + (I - \Pi_{gs}) \succeq \Omega(\frac{1}{T}) I$

Proof sketch. Use Π_0 to denote H_{out} and use Π_1 to denote $I - \Pi_{gs}$. Both of them are projectors by definition. We will derive a lower bound $\Omega(\frac{1}{T})$ for all the eigenvalues of $\Pi_0 + \Pi_1$, by using the Jordan's Lemma.

Recall that by Jordan's Lemma, we can decompose the entire space into 1-dimensional or 2-dimensional subspaces which are invariant under both projectors Π_0 and Π_1 (so in the eigendecomposition we have a block-diagonal form where each block has shape 1×1 or 2×2). In each 2-dimensional subspace, Π_0 and Π_1 are rank-1 projectors. More precisely, we can find vectors $|u\rangle$ and $|v\rangle$ such that Π_0 projects on $|u\rangle$ and Π_1 projects on $|v\rangle$. Suppose the angle between those two vectors is θ. Please see the graph below:



Note that θ is a nonzero angle. Why? If we draw the orthogonal vector of $|v\rangle$, then it is just corresponding to the projector Π_{gs}, because we defined $\Pi_1 := I - \Pi_{gs}$. Suppose $\theta = 0$, then $|u\rangle$ aligns with $|v\rangle$ and the vectors Π_{gs} projects on are orthogonal to $|u\rangle$. From (14.5) we know that all the vectors Π_{gs} projects on have the form of the history states. If they are orthogonal to $|u\rangle$, then it means they have 0 energy on $\Pi_0 = \Pi_{out}$. This is a contradiction because

$$\langle history| H_{out} |history\rangle \geq \frac{1}{(\sqrt{T+1})^2} \cdot (1 - e^{-n}) > 0$$

by the assumption in the second claim of Theorem 14.1.1.

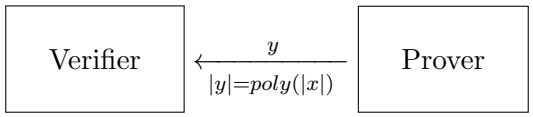
Then we can deduce a desired lower bound on the nonzero angle θ and this further gives a lower bound on the eigenvalues of $\Pi_0 + \Pi_1$. We will not present the details here, but one can refer to the survey by Aharonov and Naveh [AN02]. □

14.2 Marriott Watrous Protocol

14.2.1 Amplification of QMA

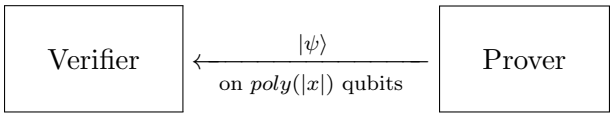
The $MA_{c,s}$ protocol is described as follows.

$$\begin{aligned} \text{(Completeness) } x \in L &\implies \exists y \Pr[V(x,y) = 1] = c \\ \text{(Soundness) } x \notin L &\implies \forall y \Pr[V(x,y) = 1] \leq s. \end{aligned}$$



Now we want amplify the completeness-soundness gap $c - s$ such that $c \to 0.99$ and $s \to 0.01$. One strategy is to run the algorithm N times and accept only if we get the output 1 from the verifier more than $N\frac{c+s}{2}$ times. We expect $N = \Theta(\frac{1}{(c-s)^2})$ by the Chernoff bound. For QMA protocol, we define c_x and s_x (as functions of x) as follows.

$$\begin{aligned} \text{(Completeness) } x \in L &\implies c_x := \max_{|\psi\rangle} \Pr[V(x, |\psi\rangle) = 1] \\ \text{(Soundness) } x \notin L &\implies s_x := \max_{|\psi\rangle} \Pr[V(x, |\psi\rangle) = 1]. \end{aligned}$$



We call the algorithm above $\mathcal{A}_x$. Pondering about the interpretation for c_x and s_x will convince us that the definitions above are correct (thinking in the example of 3SAT might be helpful). When $x \in L$, there exists some proof $|\psi\rangle$ such that $V(x, |\psi\rangle) = 1$. For completeness, we only care about the valid proofs, so we defined $c_x := \max_{|\psi\rangle} \Pr[V(x, |\psi\rangle) = 1]$ and we want c_x to be as large as possible. On the other hand, when $x \notin L$, we want that for all the proofs given to the verifier, the

probability of accepting is low. Hence we defined $s_x := \max_{|\psi\rangle} \Pr[V(x, |\psi\rangle) = 1]$ and want s_x to be as low as possible.

For the amplification of QMA (for example, from $\mathrm{QMA}_{c,s}$ to $\mathrm{QMA}_{0.99,0.01}$), we can use the same approach as for the amplification of MA, by running the verification N times. However, after measurement, our quantum state become useless, so we need N copies of $|\psi\rangle$ for this approach. Instead, we will introduce a protocol by Marriott and Watrous, which only uses one copy of $|\psi\rangle$ to accomplish the amplification.

14.3 Marriott-Watrous Protocol

Theorem 14.3.1 (MW05)**.** *There is a procedure that uses one copy of $|\psi\rangle$ and makes $N = O(\frac{1}{(c-s)^2})$ calls to $\mathcal{A}_x$, $\mathcal{A}_x^{-1}$ (with measurements), such that if $x \in L$, then it accepts with probability 0.99 and if $x \notin L$, then it accepts with probability 0.01.*

We will discuss this protocol in details in the next lecture. Here are some applications of the Marriott-Watrous protocol:

- Quantum Cryptography [Wat06].
- Quantum Signal Processing.
- Quantum Walks.
- Black Holes [YK17].

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 15

March 23, 2022 *Scribe: Justin Mack*

15.1 Lecture Plan

Goals

1. Finish Marriott-Watrous Discussion
2. Discuss Hamiltonian Simulation

15.2 Marriot-Watrous Protocol

15.2.1 Overview

Recall that the setting for Marriott-Watrous is amplification. Quantum Merlin-Arthur is a class where a prover sends a proof x to a verifier. The verifier then performs some quantum algorithm for measurement and accepts the proof with probability $\geq c$ if x belongs to a language L – this is called completeness – and accepts the proof with probability $\leq s$ if x does not belong to L – this is called soundness.

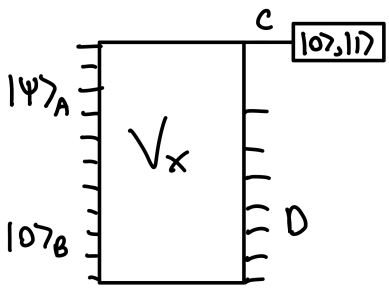
$$P_{\text{accept}} = \begin{cases} \geq c, & x \in L \\ \leq s, & x \notin L \end{cases}$$

The goal of Marriot-Watrous is to increase the difference between c and s. For a long time, people thought the only way to do this was to have the prover send multiple copies of the witness, because the information was lost when it was measured by the verifier. However, the Marriott-Watrous protocol uses just one copy of the witness to perform this amplification.

15.2.2 The Verifier Circuit

Specification

We have a circuit V that depends on x, so we write V_x – this is the verifier circuit. It takes as input in register A the witness state $|\psi\rangle_A$. It also takes the ancillary input $|0\rangle_B$ in register B. After performing some quantum computation on these inputs, the circuit then measures a certain output qubit in register X in the $\{|0\rangle, |1\rangle\}$ basis, where $|0\rangle$ and $|1\rangle$ correspond to the reject and accept states, respectively. The remaining output is stored in a register D.



The verifier circuit has the property that if we have $x \in L$ there exists a witness such that it accepts with probability at least c. It also has the property that if we have $x \notin L$ for all witnesses it accepts with probability at most s.

$$x \in L \implies \exists\, |\psi\rangle\, P[V_x(|\psi\rangle) = |0\rangle] = c.$$
$$x \in L \implies \forall\, |\psi\rangle\, P[V_x(|\psi\rangle) = |0\rangle] \leq s.$$

Ancillary Registers

Claim
Without ancillaries, there is always a state $|\psi\rangle$ such that the verifier circuit is guaranteed to accept.
Argument
Imagine we have a circuit V_x with only the input $|\psi\rangle$. Our goal is to set the output register C to $|1\rangle$ so the circuit always accepts, and we don't care about the outputs in D, so the desired input state lives in the subspace

$$|1\rangle\langle 1|_C \otimes \mathbb{1}_D.$$

On any such output state, we can apply the circuit in reverse

$$|\psi\rangle = V_x^{-1}\left(|1\rangle\langle 1|_C \otimes \mathbb{1}_D\right) V_x.$$

Basically, we can find an input that always outputs $|1\rangle$ by setting C to $|1\rangle$, D to anything, and reversing the circuit. In this case, it doesn't matter what x is because c and s will both always be 1.
To get around this issue, we add an ancilliary $|0\rangle$. The difference is that if we try to apply the circuit in reverse we might get a different value for the ancillary, which will make that state an invalid input to the circuit.

The Input Sets

We are interested in the set of states that are always accepted by the circuit. Let's call this set Π_1, which is a subspace of the Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$.

$$\Pi_1 \subseteq \mathcal{H}_A \otimes \mathcal{H}_B.$$

Π_1 is defined as the following projector for the same reasons discussed in the ancillary registers section. We start with our desired output and run the circuit in reverse to define the set of accepted inputs.

$$\Pi_1 = V_x^{-1}(|1\rangle\langle 1|_C \otimes \mathbb{1}_D)V_x.$$

We can see that Π_1 is a projector because if you square it you get back the same value

$$\begin{aligned}
&V_x^{-1}(|1\rangle\langle 1|_C \otimes \mathbb{1}_D)V_x V_x^{-1}(|1\rangle\langle 1|_C \otimes \mathbb{1}_D)V_x. \\
&= V_x^{-1}(|1\rangle\langle 1|_C \otimes \mathbb{1}_D)(|1\rangle\langle 1|_C \otimes \mathbb{1}_D)V_x. \\
&= V_x^{-1}(|1\rangle\langle 1|_C |1\rangle\langle 1|_C \otimes \mathbb{1}_D \mathbb{1}_D)V_x. \\
&= V_x^{-1}(|1\rangle\langle 1|_C \otimes \mathbb{1}_D)V_x.
\end{aligned}$$

We are also interested in the set of states that are considered valid inputs to the circuit. Let's call this set Π_2, which is also a subspace of the Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$.

$$\Pi_2 \subseteq \mathcal{H}_A \otimes \mathcal{H}_B.$$

Π_2 is defined as the set of states where register A can be any state and register B is $|0\rangle$, as these define all valid inputs to the circuit.

$$\Pi_2 = \mathbb{1}_A \otimes |0\rangle\langle 0|_B .$$

15.2.3 The Marriott-Watrous Theorem

Remember that our goal is to amplify the difference between c and s. The natural approach requires an increase in the length of the message due to the no-cloning theorem, but the remarkable aspect of Marriott-Watrous is that it showed there is a way to perform the amplification without this additional input.

Theorem 15.2.1. *There is a protocol which takes one witness $|\psi\rangle$ and makes $N = \mathcal{O}\left(\frac{1}{(c-s)^2}\right)$ calls to the circuits V_x and V_x^{-1}, such that if $x \in L$ the circuit accepts with probability 0.99 and if $x \notin L$ the circuit accepts with probability 0.01.*

Comparison with Classical

In the classical case, the prover sends a proof string y to the verifier. The verifier then runs the verification circuit $N = \mathcal{O}\left(\frac{1}{(c-s)^2}\right)$ times, and checks if the number of accepts is at least $N \cdot \frac{(c+s)}{2}$. If so, they accept, and if not they reject.

15.2.4 Specifying the Protocol

The protocol input is $|\psi\rangle_A \otimes |0\rangle_B$.

Step 1

Measure the current state in the basis $\{\Pi_1, \mathbb{1} - \Pi_1\}$. If the outcome is Π_1 we output the bit 1, and if the outcome is $\mathbb{1} - \Pi_1$ we output the bit 0.

Step 2

Measure the current state in the basis $\{\Pi_2, \mathbb{1} - \Pi_2\}$. If the outcome is Π_2 we output the bit 1, and if the outcome is $\mathbb{1} - \Pi_2$ we output the bit 1.

Step 3

Repeat the previous steps N times.

Step 4

Look at the $2N$ output bits. Let's say for instance we got the output string

$$0011101011.$$

To decide whether to accept or reject, we scan the string from left to right and count the number of times that the bit does not change value. To be more specific, you count the changes between the $(2i-1)$-th and the $2i$-th bit. If this number is at least $N(c+s)$ we accept, but if it is less than $N(c+s)$, then we reject.

15.2.5 Understanding the Protocol

Claim 1

Given a state $|\psi\rangle_A \otimes |0\rangle_B$, the probability of acceptance is given by

$$P_{\text{accept}} = \text{tr}\left(\left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right)\Pi_1\right).$$

To show that this is true, let's plug in for Π_1.

$$\text{tr}\left(\left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right)V_x^{-1}\left(|1\rangle\langle 1|_A \otimes \mathbb{1}_B\right)V_x\right).$$

We then use the property that the trace keeps matrices cyclic to write

$$\text{tr}\left(V_x\left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right)V_x^{-1}\left(|1\rangle\langle 1|_A \otimes \mathbb{1}_B\right)\right).$$

The first term $V_x\left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right)V_x^{-1}$ is the output state of the circuit when the input is $|\psi\rangle_A \otimes |0\rangle_B$, and the second term $\left(|1\rangle\langle 1|_A \otimes \mathbb{1}_B\right)$ is the projector onto the accepting inputs. Phrased differently, we are taking the input state, getting the corresponding output state, and measuring the desired register – giving us the acceptance probability.
The other way to interpret this is to look at the input side rather than the output side.

$$\left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right)\Pi_1.$$

We can interpret this entire expression as the overlap between the set of valid inputs and the set of accepting inputs – the higher this overlap the higher the probability of acceptance on an arbitrary input state.

Claim 2

The maximum acceptance probability is the largest eigenvalue of the operator

$$\Pi_2\Pi_1\Pi_2.$$

We can first show mathematically why this is the case

$$\max_{|\psi\rangle} P_{\text{accept}} = \max_{|\psi\rangle} \operatorname{tr}\left(\left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right) \Pi_2 \Pi_1 \Pi_2\right).$$
$$\max_{|\psi\rangle} P_{\text{accept}} = \max_{|\psi\rangle} \operatorname{tr}\left(\left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right) \left(\mathbb{1}_A \otimes |0\rangle\langle 0|_B\right) \Pi_1 \Pi_2\right).$$
$$\max_{|\psi\rangle} P_{\text{accept}} = \max_{|\psi\rangle} \operatorname{tr}\left(\left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right) \Pi_1 \Pi_2\right).$$

Applying the cyclic property

$$\max_{|\psi\rangle} P_{\text{accept}} = \max_{|\psi\rangle} \operatorname{tr}\left(\Pi_2 \left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right) \Pi_1\right).$$
$$\max_{|\psi\rangle} P_{\text{accept}} = \max_{|\psi\rangle} \operatorname{tr}\left(\left(\mathbb{1}_A \otimes |0\rangle\langle 0|_B\right) \left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right) \Pi_1\right).$$
$$\max_{|\psi\rangle} P_{\text{accept}} = \max_{|\psi\rangle} \operatorname{tr}\left(\left(|\psi\rangle\langle\psi|_A \otimes |0\rangle\langle 0|_B\right) \Pi_1\right).$$

We can see that this result is the same as the probability of acceptance expression we proved in Claim 1, except we are taking the maximum over all valid input states.
Application of Jordan's Lemma
Let's now talk about the operator

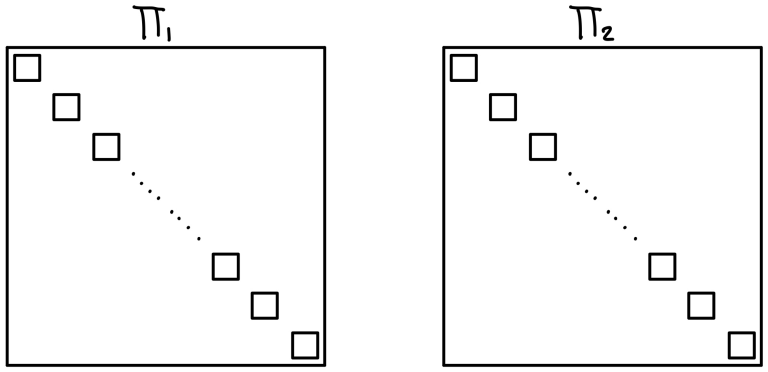
$$\Pi_1 \Pi_2.$$

We can use Jordan's Lemma to decompose this operator. This gives us an orthonormal basis with the following properties.

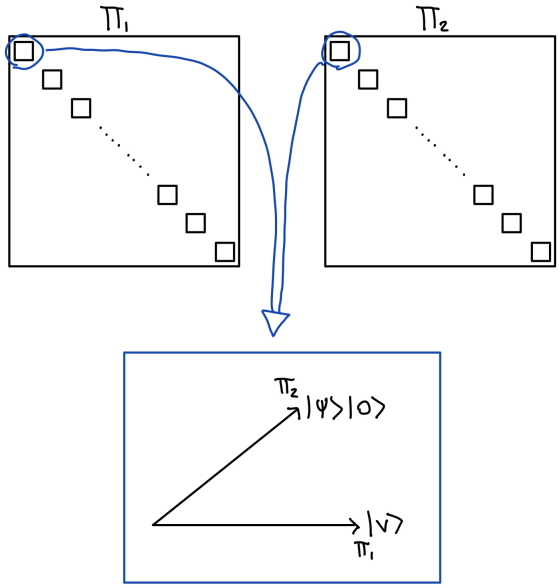
1. For both Π_1 and Π_2, the matrix form under this basis is the block diagonal with either 1-D or 2-D blocks.

2. In each 2-D block, both Π_1 and Π_2 have dimension 1. In other words, in that block we know that Π_1 is $|w\rangle\langle w|$ and Π_2 is $|v\rangle\langle v|$.

Thus, by choosing $\left|w^{\perp}\right\rangle$ (resp. $\left|v^{\perp}\right\rangle$) that is orthonormal to $|w\rangle$ (resp. $|v\rangle$), we get an orthonormal basis $\left\{|w\rangle, \left|w^{\perp}\right\rangle\right\}$ (resp. $\left\{|v\rangle, \left|v^{\perp}\right\rangle\right\}$) for this block.



Zoom in on these blocks, we have a component $|v\rangle \in \Pi_1$, and a component $|\psi\rangle_A \otimes |0\rangle_B \subseteq \Pi_2$. An important note is that $|\psi\rangle_A \otimes |0\rangle_B$ – the component of Π_2 within the Jordan blocks – is an eigenvector of $\Pi_2\Pi_1\Pi_2$.



The point of this is that if we have some valid input $|\psi\rangle_A \otimes |0\rangle_B \subseteq \Pi_2$, then its overlap with $|v\rangle$ is its projection onto $|v\rangle$. Let's say the two components are at an angle θ. In this case, we have that

$$P_{\text{accept}}(|\psi\rangle_A) = \cos^2(\theta).$$

Focusing on the first block, we can project $|\psi\rangle_A \otimes |0\rangle_B$ onto $|v\rangle$ as follows:

$$(\Pi_2\Pi_1\Pi_2)_{\text{within the block}} = (|\psi\rangle \otimes |0\rangle\langle\psi| \otimes \langle 0|)\,(|v\rangle\langle v|)\,(|\psi\rangle \otimes |0\rangle\langle\psi| \otimes \langle 0|)\,.$$
$$(\Pi_2\Pi_1\Pi_2)_{\text{within the block}} = |\langle v|\psi\rangle\,|0\rangle|^2\,|\psi\rangle \otimes |0\rangle\,\langle\psi| \otimes \langle 0|\,.$$

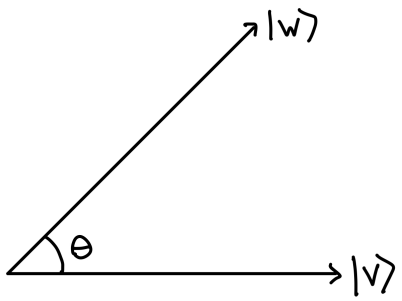
The resulting constant $|\langle v|\psi\rangle\,|0\rangle|^2$ is the eigenvalue of the $\Pi_2\Pi_1\Pi_2$ operator on this subspace. This tells us that the block component overlaps ($\cos^2(\theta)$) are equivalent to the eigenvalues of $\Pi_2\Pi_1\Pi_2$. To specify the eigendecompsition we have been discussing in math, we have

$$\Pi_2\Pi_1\Pi_2 = \sum_i P_{\psi_i}\,|\psi_i\rangle\,|0\rangle\langle\psi_i|\,\langle 0|$$

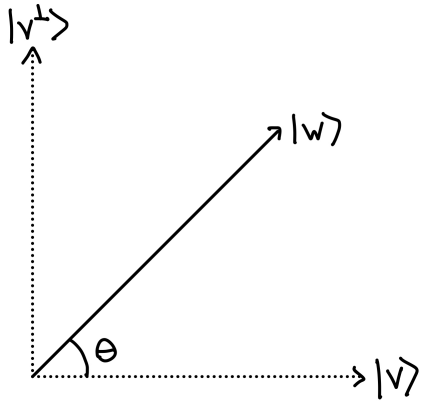
where $|\psi_i\rangle$ is the accept state after measuring in the $\{\Pi_1, \mathbb{1} - \Pi_1\}$ basis. We now narrow our focus to some specific value of i.

Bringing it Home

We have proven our claims, and now we want to use these reach a final understanding of the protocol. Let's focus on just one of these blocks from the Jordan's Lemma decomposition and see what the protocol looks like. We start with our two components $|w\rangle$ and $|v\rangle$ at an angle θ from each other.



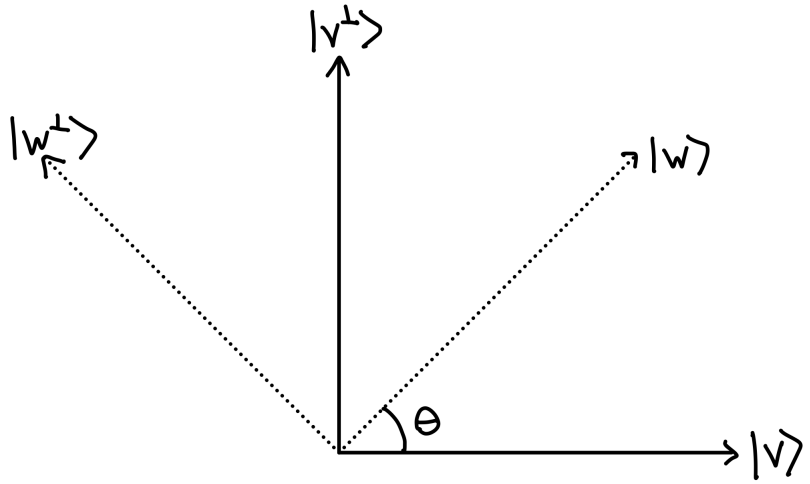
In the first step of the protocol, the input is $|w\rangle = |\psi\rangle_A \otimes |0\rangle_B$, and we measure in the $\{\Pi_1, \mathbb{1} - \Pi_1\}$ basis – which is the same as the $\{|v\rangle, |v^\perp\rangle\}$ basis.



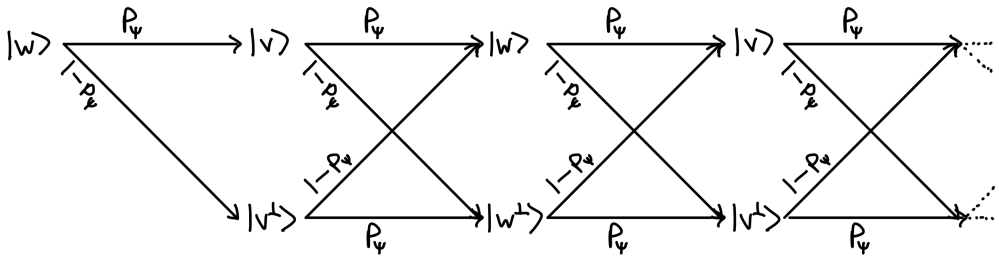
When we perform the measurement, we get the accept state $|v\rangle$ with probability $\cos^2(\theta)$. We'll call this P_ψ, which means we reject with probability $1 - P_\psi$.

$$P_\psi = P_{\text{accept}}(|\psi\rangle) = \cos^2(\theta).$$

After this first measurement, we are either in the state $|v\rangle$ or $|v^\perp\rangle$, and we then measure in the $\{|w\rangle, |w^\perp\rangle\}$ basis.



We then continually repeat this process, so we basically just have the current state vector bouncing randomly between the four states $|v\rangle, |w\rangle, |v^\perp\rangle, |w^\perp\rangle$. We can express this process using the following diagram.



It become more apparent now why in step 4 of the protocol we analyze the output bit string by looking at instances in which the bits did not change. From the diagram we can see that we don't change bits with probability P_ψ, and we do change bits with probability $1 - P_\psi$. Therefore, by counting the number of times the bit does not change we can estimate P_ψ.

15.2.6 Conclusion

The main takeaway from this lecture is that the Marriott-Watrous protocol performs coin tosses with heads and tails probabilities $(P_\psi, 1-P_\psi)$ on an input $|\psi\rangle|0\rangle$ which is an eigenstate of $\Pi_2\Pi_1\Pi_2$. In our homework, we will show that this still works for linear combinations of eigenstates.

15.3 Hamiltonian Simulation

15.3.1 Overview

Given a Hamiltonian H, a quantum state evolves as follows

$$H : \rho \to e^{iHt}\rho e^{-iHt}$$

You generally don't have access to the Hamiltonian of matter, but you still want to be able to perform that quantum evolution so you can simulate the properties of that matter. This is where Hamiltonian simulation comes in.

15.3.2 Applications

- One application of Hamiltonian simulation is simulating the properties of exotic matter.
- Another application of Hamiltonian simulation is solving linear systems, with some caveats.

15.3.3 Goal

The goal of Hamiltonian simulations is, given H as input, to find a quantum circuit V that with size polynomial in t (and some other factors) such that

$$V\rho V^\dagger \approx e^{iHt}\rho e^{-iHt}.$$

For our purposes, we can focus on pure states, because if we can approximate pure states then we can always also approximate mixed states by decomposing them and approximating each of those pure states individually.

$$V|\psi\rangle \approx e^{iHt}|\psi\rangle .$$

Sometimes ancillary registers are necessary, in which case we rewrite as

$$V\rho \otimes |0\rangle\langle 0|_B V^\dagger \approx e^{iHt}\rho_A e^{-iHt} \otimes |\ldots\rangle\langle\ldots|_B .$$

15.3.4 Method 1: Trotter Method

This method applies only to a subset of Hamiltonians known as local Hamiltonians. It also has a relatively high computation cost with an asymptotic runtime of $\mathcal{O}(t^2)$ (there are also some other factors). However, this is the more "physical" method, and physicists have been using it for many years. We will discuss this method in greater detail in a future lecture.

15.3.5 Method 2: Phase Estimation with Quantum Walks

This method applied to a broader subset of Hamiltonians known as sparse Hamiltonians. It also has the optimal runtime of $\mathcal{O}(t)$ (there are also some other factors). The method is inspired by computer science and quantum walks, not physical sciences. Let's say we have a Hamiltonian H with the eigendecomposition

$$H = \sum_i E_i |\phi_i\rangle\langle\phi_i| .$$

In this case, if we apply e^{iHt} to a state $|\psi\rangle$, we can write $|\psi\rangle$ in the eigenbasis of H and then apply e^{iHt} to get

$$e^{iHt}|\psi\rangle = \sum_j c_j e^{iE_j t}|\phi_j\rangle .$$

Given this, Hamiltonian simulation works as follows. We first treat $|\phi_i\rangle$ as an input and perform phase estimation to get the value E_i. We then apply the number $e^{iE_i t}$ to the output state. Finally, we uncompute to get the desired output, which is allowed because everything is unitary thus allowing us to undo. The process mathematically looks like this

$$|\phi_i\rangle \to |\phi_i\rangle|E_i\rangle \to e^{iE_i t}|\phi_i\rangle|E_i\rangle \to e^{iE_i t}|\phi_i\rangle .$$

However, we don't know how to actually compute the phase E_i, because phase estimation requires a unitary. It turns out that there exists a unitary called the quantum walk operator that contains the necessary information about the eigenvalues of the Hamiltonian – allowing us to perform this phase estimation. We will discuss this in more detail in a future lecture. We'll also see that just like we can transform any random walk process into a quantum walk, we can also transform a Hamiltonian into a quantum walk operator.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 16

March 25, 2022 *Scribes: Vassilios Kaxiras*

16.1 Today

1. Trotter method (Hamiltonian simulation)
2. Quantum Linear Systems

16.1.1 Recap

What is Hamiltonian simulation? Given a Hamiltonian H on n qubits, find

$$V \approx e^{iHt} \tag{16.1}$$

We want the size (number of gates) of the quantum circuit V to be polynomial in n and t. Two methods of getting V:

1. Trotter: for local Hamiltonians: requires $O(t^2)$ 2 qubit gates
2. Quantum walks: for sparse hamiltonians: requires $O(t)$ circuit depth

16.2 Trotter method

Consider a 2-local Hamiltonian:

$$H = \sum_{\alpha=1}^{m} b_\alpha P_\alpha, \quad b_\alpha \in [-1, 1] \tag{16.2}$$

H is 2-local if each P_α is a 2-qubit Pauli operator, which means it performs a Pauli operation on at most 2 qubits, and the identity on the rest. For example, a possible H with $m = 2(n-1)$ (n being the number of qubits) is

$$\begin{aligned} H &= \sum_{i=1}^{n-1} (X_i \otimes X_{i+1} + Z_i \otimes Z_{i+1}) \\ X_i &= \mathbb{1} \otimes \mathbb{1} \otimes \cdots X \otimes \mathbb{1} \otimes \cdots \mathbb{1} \end{aligned} \tag{16.3}$$

Theorem 16.2.1. *Given $\{b_\alpha, P_\alpha\}_{\alpha=1}^m$ and ϵ allowed error, you can efficiently construct a quantum circuit V:*

$$\|V - e^{-iHt}\|_\infty \leq \epsilon \tag{16.4}$$

The size (in number of 2 qubit gates) is

$$O\left(\frac{m^3t^2}{\epsilon}\right) \tag{16.5}$$

with depth

$$O\left(\frac{m^2t^2}{\epsilon}\right) \tag{16.6}$$

Proof. Use the Suzuki-Trotter method:

1. Group $\{P_\alpha\}_{\alpha=1}^m$ into mutually commuting terms.
2. Evolve for tiny time $e^{i\delta H}$, δ small.
3. Repeat

In general, we know

$$e^{iA+iB} \neq e^{iA}e^{iB} \tag{16.7}$$

if A, B don't commute. This means

$$e^{iHt} \neq \Pi_\alpha e^{ib_\alpha P_\alpha t} \tag{16.8}$$

Now we are specifically interested in

$$\begin{aligned} H &= \sum_i (X_i \otimes X_{i+1} + Z_i \otimes Z_{i+1}) = H_1 + H_2 \\ H_1 &\equiv \sum_i^m X_i \otimes X_{i+1}, \quad H_2 \equiv \sum_i^m Z_i \otimes Z_{i+1} \end{aligned} \tag{16.9}$$

The question is, how does e^{iHt} compare to $e^{iH_1t}e^{iH_2t}$? We want to find this because all the terms in H_1 commute and all the terms commute in H_2, so we can write e^{iH_1t} and e^{iH_2t} with an efficient quantum circuit. However, H_1 and H_2 do not commute, so we cannot simply write $e^{iH} = e^{iH_1}e^{iH_2}$. We can solve this with the Suzuki-Trotter method, which estimates e^{iH} by performing incremental steps as described below.

Suzuki-Trotter method:

$$e^{i\delta H} \approx e^{i\delta H_1}e^{i\delta H_2} \tag{16.10}$$

Claim 16.2.2.

$$\|e^{i\delta H} - e^{i\delta H_1}e^{i\delta H_2}\|_\infty \leq O(m^2\delta^2 e^{\delta m}) \tag{16.11}$$

Proof. Idea: Note

$$e^{i\delta H} = I + i\delta H + \sum_{l=2}^{\infty} \frac{(i\delta H)^l}{l!} \tag{16.12}$$

So

$$\|e^{i\delta H} - I - i\delta H\|_\infty = \left\|\sum_{l=2}^{\infty} \frac{(i\delta H)^l}{l!}\right\|_\infty \leq \sum_{l=2}^{\infty} \frac{\delta^l}{l!}\|H\|_\infty^l \leq \sum_{l=1}^{\infty} \frac{\delta^l}{l!}(2m)^l \leq \delta^2(2m)^2 e^{2\delta m} \tag{16.13}$$

where $\|H\|_\infty \leq 2m$ since there are $2m$ terms with max eigenvalue 1 in H ($\|\cdot\|_\infty$ denotes the maximum eigenvalue of the operator). □

So,

$$\begin{aligned} e^{i\delta H} &\approx I + i\delta H \\ e^{i\delta H_1}e^{i\delta H_2} = (I + i\delta H_1)(I + i\delta H_2) &= I + i\delta(H_1 + H_2) - \delta^2 H_1 H_2 \end{aligned} \tag{16.14}$$

Now by the same argument as above for $\|H\|_\infty \leq 2m$, we have

$$\delta^2 \|H_1 H_2\|_\infty \leq \delta^2 m^2 \tag{16.15}$$

which is on the order of the error between $e^{i\delta H}$ and $I + i\delta H$, we get

$$e^{i\delta H} \approx I + i\delta H \approx e^{i\delta H_1} e^{i\delta H_2} \tag{16.16}$$

□

In summary, the Trotter method uses the fact that

$$e^{iHt} = \left(e^{iH\delta}\right)^{\frac{t}{\delta}} \tag{16.17}$$

and, for δ small,

$$e^{iH\delta} \approx e^{i\delta H_1} e^{i\delta H_2} \tag{16.18}$$

Lemma 16.2.3. *Given matrices P, Q, R with $\|P\|_\infty, \|Q\|_\infty, \|R\|_\infty \leq 1$, then*

$$\|P - QR\|_\infty \leq \epsilon \implies \|P^k - (QR)^k\|_\infty \leq k\epsilon \tag{16.19}$$

So using this Lemma,

$$\begin{aligned} \|e^{i\delta H} - e^{i\delta H_1} e^{i\delta H_2}\|_\infty \leq O(m^2\delta^2 e^{\delta m}) \implies \\ \left\| e^{iHt} - (e^{i\delta H_1} e^{i\delta H_2})^{\frac{t}{\delta}} \right\|_\infty \leq O(m^2 t\delta e^{2m\delta}) \end{aligned} \tag{16.20}$$

So if we want our error to be ϵ, we need

$$\delta = \frac{\epsilon}{m^2 t} \tag{16.21}$$

Thus the depth of the circuit needed to simulate this Hamiltonian is

$$O\left(\frac{t}{\delta}\right) = O\left(\frac{m^2 t^2}{\epsilon}\right) \tag{16.22}$$

Which proves equation (16.6).

16.3 Quantum Walk

H is a $2^n \times 2^n$ matrix, and H is s-sparse, which means each row and column has at most s non-zero elements. Compared to locality, this effectively means there are s distinct P_α's, but each P_α can be arbitrarily non-local (it can be up to n-local).
Furthermore, each entry of H is $\leq \alpha$.
The depth is $O\left(\frac{st\alpha}{\epsilon}\right)$
If H is local, $s \leq m$ and $\alpha \leq m$, which gives a better result than Trotter method.
More on this next week!

16.4 Linear Systems

Given an $N \times N$ matrix A and a matrix b that is $N \times 1$, we want to find an $N \times 1$ matrix x such that

$$Ax = b \tag{16.23}$$

It is know that if A is an s-sparse matrix, then for k being the ratio of the highest to lowest eigenvalue of A (called the condition number of A), you can figure out x with some error in time $O(Nks)$.

16.4.1 Quantum Linear Systems (Harrow, Hassidim, and Lloyd)

Given a s-sparse matrix A and a vector b,

$$|b\rangle = \frac{1}{\|b\|} \sum_i b_i |i\rangle \tag{16.24}$$

over a Hilbert space of dimension N with basis $\{|1\rangle, |2\rangle, \ldots, |N\rangle\}$.
$|b\rangle$ can be generated by a unitary oracle U:

$$|b\rangle = U|1\rangle \tag{16.25}$$

Suppose we have oracle access to A, which gives you access to two data sources:

1. Given x, y we can know A_{xy} in 1 query.
2. Given x, we can know the non-zero columns of A in row x with $s \ll N$ queries.

Our goal is to output

$$|x\rangle = \frac{1}{\|x\|} \sum_i x_i |i\rangle \tag{16.26}$$

such that $Ax = b$.
The HHL algorithm takes

$$O\left(\frac{k^2 s}{\epsilon}\right) \tag{16.27}$$

queries to O and U to output $|x\rangle$ with error ϵ. This is independent of N! Thus, this may be potentially useful for solving linear systems where only a small part of the output x is needed.

Proof. Fact: Can assume that A is Hermitian. Suppose A is not Hermitian. Then define a new problem

$$\begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix} \begin{bmatrix} 0 \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{16.28}$$

whose solution will give solutions to $Ax = b$.
Sketch of proof:
Think of A as Hamiltonian. Then

$$A = \sum_i E_i |\phi_i\rangle \langle\phi_i| \tag{16.29}$$

We want

$$|x\rangle \propto A^{-1} |b\rangle \tag{16.30}$$

Decompose $|b\rangle$ into the eigenvalues of A:

$$|b\rangle = \sum_i \mu_i |\phi_i\rangle \tag{16.31}$$

Then

$$A^{-1} |b\rangle = \sum_i \frac{\mu_i}{E_i} |\phi_i\rangle \tag{16.32}$$

Then perform the following steps:

1. Use phase estimation to figure out E_i.
2. Try to change $|\phi_i\rangle$ to $\frac{1}{E_i} |\phi_i\rangle$
3. Repeat

The advantage comes from the fact that PE occurs in superposition. More on this next week! □

Furthermore, for any problem in BQP, you can find an observable M, a matrix A, and a vector b such that for the solution x to $Ax = b$, determining $\langle x| M |x\rangle$ will solve that problem. In other words, the problem of solving quantum linear systems is BQP-complete.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 17

March 30, 2022 *Scribes: Ethan Lee*

Today's lectures notes are on the following topics:

- Quantum Linear Systems
- Quantum Walks (revisited)

17.1 Recap: Hamiltonian Simulation, Quantum Linear Systems

17.1.1 Hamiltonian Simulation

Recall from the last lecture that the goal of Hamiltonian simulation is, given the s-sparse matrix H, to simulate

$$e^{iHt}. \tag{17.1}$$

We found that the cost of this simulation is $O(s \cdot \frac{t}{\epsilon})$, where ϵ is the error term.

Why do we consider e^{iHt} in this process and not e^{-iHt}? Refer to Harrow's Criteria: a computer scientist will wish to simulate e^{iHt} while a physicist will wish to simulate e^{-iHt}.

H is specified by the following oracles (by "specified", we mean that we access H through these oracles):

$$x, y \longrightarrow \boxed{\text{O}} \longrightarrow H_{x,y}$$

$$x \longrightarrow \boxed{\text{O}} \longrightarrow \frac{1}{\sqrt{s}} \sum_{y:H_{x,y}\neq 0} |x, y\rangle$$

The first oracle accesses elements in H. The second oracle takes input x and outputs the superposition over all columns in H with non-zero entries for x.

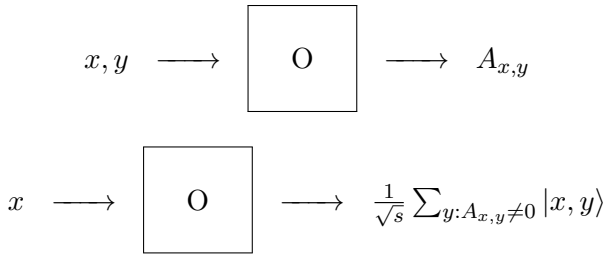
17.1.2 Quantum Linear Systems

In the Quantum Linear Systems problem, we have a hermitian, s-sparse, $N \times N$ matrix A, where s-sparse means we have at most s non-zero entries per row of A. Note that if A is not hermitian to start with, we can use the trick mentioned in the last lecture to construct the problem

$$\begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix} \begin{bmatrix} 0 \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}. \tag{17.2}$$

.

Then, similarly to the oracles defined above, we access the matrix A through the following oracles:



Now, given $|b\rangle$ written as

$$|b\rangle = \frac{1}{\|b\|} \sum_i b_i |i\rangle ,$$

we encode a certain column matrix

$$\bar{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} . \tag{17.3}$$

Note that above, being "given" $|b\rangle$ means that we have a nice unitary U such that

$$U |1\rangle = |b\rangle . \tag{17.4}$$

The choice of $|1\rangle$ is unimportant here, just that it is a simple state.

Then, the solution of the Quantum Linear System (QLS) problem is to find a quantum version of $\bar{x}$, such that

$$\bar{x} = A^{-1} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} . \tag{17.5}$$

More precisely, the goal of the QLS problem then is to output a state $\overset{\epsilon}{\approx} |x\rangle = \frac{1}{\|x\|_2} \sum_i x_i |i\rangle$. Note that this is analogous to the typical linear systems problem of finding $x = A^{-1}b$.

17.2 Quantum Linear Systems

17.2.1 QLS Solution Theorem

Theorem 17.2.1. *We can find $|x\rangle$ in $O(\kappa^2 \frac{s}{\epsilon})$ queries to oracles and uses of U, where κ is the condition number of A, the ratio of the largest eigenvalue and smallest eigenvalue of A.*

We prove this theorem using the information in the following subsections.

17.2.2 Rewriting the QLS Problem

Before getting into the theorem proof, note that the technique described for solving the QLS problem can be used to solve any problem of the form

$$x = f(A) \cdot b, \tag{17.6}$$

where f is a 'nice' matrix-valued function. Then, let us consider applying f to a scalar, which we can think of as being the eigenvalue of a matrix. Within this form, we assume f satisfies

$$f(z) \leq 1, f(z) \geq \theta, \tag{17.7}$$

for some θ and for any z in some range I. Note that here, z is not a matrix. Then, we also assume f is continuous in the range I; that is,

$$f(z+y) \subseteq (1 \pm \eta y) f(z). \tag{17.8}$$

This form helps with solving more general problems, and we can write QLS in this form below.

Assume the spectrum of A is such that

$$spec(A) \in \left[-1, -\frac{1}{\kappa}\right] \cup \left[\frac{1}{\kappa}, 1\right]. \tag{17.9}$$

We pick our f in QLS to be

$$f(z) = \frac{1}{\kappa z}. \tag{17.10}$$

We also can see that our range I is

$$I = \left[-1, -\frac{1}{\kappa}\right] \cup \left[\frac{1}{\kappa}, 1\right], \tag{17.11}$$

as specified before. Then, this means that $f(z)$ will be between θ and 1, where $\theta = \frac{1}{\kappa}$; additionally, we can see by simple calculation (not shown here) that $\eta = \kappa$. The intuition for this equality is that the function f will change a lot at the smallest value of z.

Now, suppose we can write A in the "Hamiltonian form"

$$A = \sum_i E_i \left|\phi_i\right\rangle\!\left\langle\phi_i\right|, \tag{17.12}$$

where $|E_i| \in [\frac{1}{\kappa}, 1]$. Then, f acts on the terms in this form of A such that

$$f(A) = \sum_i f(E_i) \left|\phi_i\right\rangle\!\left\langle\phi_i\right|. \tag{17.13}$$

Now, given $|b\rangle = U|0\rangle$ for some unitary, we wish to output a state $\overset{\epsilon}{\approx} |x\rangle$ such that $x = f(A) \cdot b$. In QLS,

$$f(A) = \frac{1}{\kappa} \sum_i \frac{1}{E_i} \left|\phi_i\right\rangle\!\left\langle\phi_i\right| = \frac{1}{\kappa} A^{-1}. \tag{17.14}$$

17.2.3 Harrow-Hassidim-Lloyd Algorithm

The HHL Algorithm provides a solution to QLS and proceeds as follows:

1. Start with $|b\rangle$ from QLS, where $|b\rangle$ can be expanded in the eigenbasis of A, so that $|b\rangle = \sum_i \mu_i |\phi_i\rangle$. Then, add a register starting at $|0\rangle$ which we will use in the next step. At this point, our state is $\sum_i \mu_i |\phi_i\rangle |0\rangle$.

2. Perform phase estimation on a unitary derived from A using some unitary U (more info on what this unitary could be is provided below). Applying U changes each $|\phi_i\rangle|0\rangle$ in our state to $|\phi_i\rangle\left|\tilde{E}_i\right\rangle$, where $\tilde{E}$ is the eigenvalue estimate. At this point, our state is $\sum_i \mu_i |\phi_i\rangle\left|\tilde{E}_i\right\rangle$.

3. This step is called Rejection Sampling. Add a register $|0\rangle_Q$, and take each $|\phi_i\rangle\left|\tilde{E}_i\right\rangle|0\rangle_Q$ to $|\phi_i\rangle\left|\tilde{E}_i\right\rangle(f(\tilde{E}_i)|0\rangle_Q + \sqrt{1-f^2(\tilde{E}_i)}|1\rangle_Q)$. At this step, our state is $\sum_i \mu_i |\phi_i\rangle\left|\tilde{E}_i\right\rangle(f(\tilde{E}_i)|0\rangle_Q + \sqrt{1-f^2(\tilde{E}_i)}|1\rangle_Q)$.

4. Now, we undo the phase estimation and erase each $\left|\tilde{E}_i\right\rangle$ by applying the inverse of the phase estimation unitary. Then, our state now is $\sum_i \mu_i |\phi_i\rangle|0\rangle(f(\tilde{E}_i)|0\rangle_Q + \sqrt{1-f^2(\tilde{E}_i)}|1\rangle_Q)$

Revisiting Step 2, the phase estimation unitary must be some unitary with information about the eigenvalues of A. An example could be

$$V = e^{iAt}, \tag{17.15}$$

as this V has phases e^{iE_it}. We can use Hamiltonian Simulation to implement this unitary V.

Looking to the result of the HHL Algorithm, we have final state

$$\sum_i \mu_i |\phi_i\rangle|0\rangle(f(\tilde{E}_i)|0\rangle_Q + \sqrt{1-f^2(\tilde{E}_i)}|1\rangle_Q) = \sum_i \mu_i f(\tilde{E}_i)|\phi_i\rangle|0\rangle|0\rangle_Q + \mu_i\sqrt{1-f^2(\tilde{E}_i)}|\phi_i\rangle|0\rangle|1\rangle_Q.$$

Then, when we measure the Q register and get a 0, we have our solution, since our goal was to find $|x\rangle$ such that

$$|x\rangle \propto \sum_i \mu_i f(E_i)|\phi_i\rangle.$$

If we get a 1 upon measuring Q, we have an undesired term, suggesting that we should repeat the algorithm some number of times, which we will find below. Alternatively, we can use amplitude amplification to get further improvement on the number of repeats.

17.2.4 HHL Algorithm Analysis

We can analyze the cost of the algorithm as follows. Our resulting state from the HHL Algorithm if Q is measured to be 0 is

$$\sum_i \mu_i f(\tilde{E}_i)|\phi_i\rangle = \sum_i \mu_i(1 \pm \eta\delta) f(E_i)|\phi_i\rangle,$$

if we invoke the properties of f that we found before. Then, we can have the error term $(1 \pm \eta\delta)$ be set to $(1 \pm \epsilon)$, where ϵ is the allowed error, by setting

$$\delta = \frac{\epsilon}{\eta}.$$

This sets the cost of phase estimation to be $O(\frac{s\eta}{\epsilon})$ if we use the quantum walk operator. We can also see that the probability of getting 0 when measuring Q is $\sum_i \mu_i^2 f^2(\tilde{E}_i)$, which is bounded below

by $\theta^2 = \frac{1}{\kappa^2}$. Then, we would repeat this algorithm $O(\frac{1}{\theta^2})$ times, giving us the cost of the algorithm to be

$$O(\frac{s}{\theta^2} \cdot \frac{\eta}{\epsilon}).$$

We can improve this runtime by performing amplitude amplification. Using this, only $O(\frac{1}{\theta})$ calls to the algorithm are needed and the improved runtime is

$$O(\frac{s}{\theta} \cdot \frac{\eta}{\epsilon}) = O(\frac{\kappa s \eta}{\epsilon}),$$

which if we set $\eta = \kappa$, is equal to $O(\kappa^2 \frac{s}{\epsilon})$, the runtime from Theorem 17.2.1. Thus, using amplitude amplification is the final step towards showing that the HHL Algorithm proves Theorem 17.2.1.

17.3 Quantum Walks

17.3.1 Recap

Recall the setup of quantum walks: we have a graph $G = (V, E)$ and a transition matrix P where

$$P_{xy} = \frac{1}{d_x},$$

and such that

$$\sum_y P_{xy} = 1.$$

We also assumed Detailed Balance, the property such that, given the stationary μ,

$$\mu_x P_{xy} = \mu_y P_{yx}.$$

Recall also the edge process (covered more in scribe notes 10), where we turned a graph into a bipartite graph with a full set of vertices on the left and a full set of vertices on the right, and edges between the sets if those edges already existed in the original graph. The edge process is such that we take a random walk over all of the edges by using this bipartite version of the graph.

Recall that we defined the following states for the quantum walk:

$$\left|\psi_x^1\right\rangle = \sum_y \sqrt{P_{xy}} \left|x\right\rangle \otimes \left|y\right\rangle,$$

$$\left|\psi_y^2\right\rangle = \sum_x \sqrt{P_{yx}} \left|x\right\rangle \otimes \left|y\right\rangle.$$

We also defined the quantum walk operator,

$$W = ((2\sum_x \left|\psi_x^1\right\rangle\!\left\langle\psi_x^1\right| - \mathbb{1})((2\sum_y \left|\psi_y^2\right\rangle\!\left\langle\psi_y^2\right| - \mathbb{1}).$$

Then, we defined

$$\Pi_1 = \sum_x \left|\psi_x^1\right\rangle\!\left\langle\psi_x^1\right|, \Pi_2 = \sum_y \left|\psi_y^2\right\rangle\!\left\langle\psi_y^2\right|,$$

so that

$$W = (2\Pi_1 - \mathbb{1})(2\Pi_2 - \mathbb{1}).$$

We also defined our stationary

$$\left|\mu\right\rangle = \frac{1}{\sqrt{2|E|}} \sum_{x,y} \left|x\right\rangle \otimes \left|y\right\rangle.$$

17.3.2 Remaining Proof in Quantum Walks

Recall that when discussing quantum walks, we had the conclusion where if we had

$$spec(W) = \{e^{\pm 2i\theta_1}, e^{\pm 2i\theta_2}, \dots\},$$

and $\{\lambda_J\}$ are the eigenvalues of the classical transition matrix P, then

$$\cos\theta_J = \lambda_J.$$

We now prove this result. In the vector space, there exists a basis where Π_1 and Π_2 become block-diagonal. Then, we can write

$$\Pi_1 = \sum_b |u_b\rangle\langle u_b|, \Pi_2 = \sum_b |v_b\rangle\langle v_b|,$$

where b is the block number, and the $|u_b\rangle$ and $|v_b\rangle$ values are unit vectors or 0 by assumption.

By Jordan's Lemma, W must also respect the Jordan block structure since it is the product of reflections across Π_1 and Π_2. Then, we can write W in the form

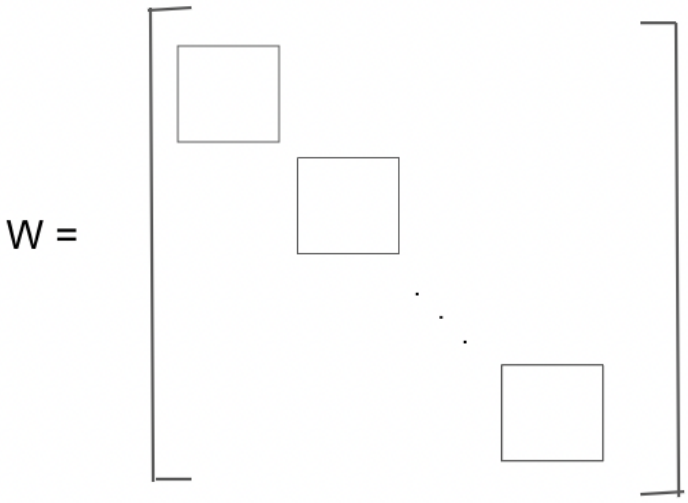


Figure 17.1: Block-Diagonalized W

If we zoom in on one of the blocks, say block b, then it contains

$$(2|u_b\rangle\langle u_b| - \mathbb{1}_b)(2|v_b\rangle\langle v_b| - \mathbb{1}_b).$$

Thus, we can reduce this problem to a 2×2 context. Since W is a product of 2 reflections in 2 dimensions, it is a rotation, and we can find the angle of this rotation by looking to an example of the effect of applying W on one specific vector, in particular, on v_b.

Figure 17.2 shows that if θ_b is the original angle between v_b and u_b, then W will move v_b by $2\theta_b$ since it simply reflects v_b about u_b, meaning the angle of rotation is $2\theta_b$. Then, we have

$$\cos\theta_b = \langle u_b|v_b\rangle.$$

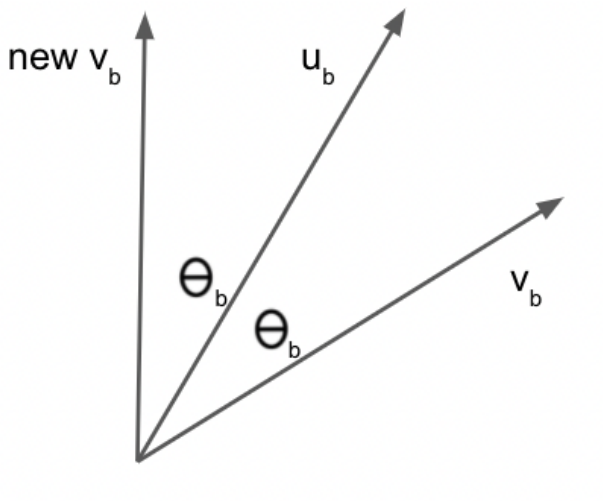


Figure 17.2: Rotating v_b about u_b

Additionally, we can write the SVD of $\Pi_1\Pi_2$ as

$$\sum_b |u_b\rangle \langle u_b|v_b\rangle \langle v_b| .$$

Thus, to find $\cos\theta_b$, we compute the singular values of $\pi_1\pi_2$, and those singular values are equal to the λ_J values from before, which are the eigenvalues of W.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 18 (Guest Lecture by Robert Huang)

April 1, 2022 *Scribes: Michael Yu and Ashley Zhuang*

18.1 Predicting Properties in a Quantum State

Suppose there is a physical source that generates the n-qubit state ρ, where $\rho \in \mathbb{C}^{2^n \times 2^n}$, $\rho \succeq 0$ (positive semi-definite) and $\mathrm{Tr}(\rho) = 1$. Quantum state tomography is the process of reconstructing ρ from the outcomes of measurements performed on identical copies of the state. In general, quantum state tomography can require exponentially-many such measurements. Instead of reconstructing the entire quantum state, which can be costly, we can utilize shadow tomography to predict properties of ρ such as $\mathrm{Tr}(O_1\rho), \mathrm{Tr}(O_2, \rho), \ldots, \mathrm{Tr}(O_M\rho)$ for observables O_i up to ϵ error.

18.2 How can we predict properties of ρ efficiently?

Task: Perform measurements on ρ to obtain a classical description S_ρ of the data which can then be used to predict $\mathrm{Tr}(O_1\rho), \mathrm{Tr}(O_2, \rho), \ldots, \mathrm{Tr}(O_M\rho)$ up to ϵ error. The idea is that after a sufficient number of measurements N, we can uniquely determine S_ρ for the quantum state and its list of observables O_i. Here we assume that $\|O_i\|_\infty \leq 1$.

18.2.1 Naive Case

Naively, we can determine $\mathrm{Tr}(O_1\rho), \mathrm{Tr}(O_2, \rho), \ldots, \mathrm{Tr}(O_M\rho)$ by measuring ρ under the eigenbasis of O_k for $k = 1, 2, \ldots, M$. To ensure that we are within ϵ error, each observable requires $L = O\left(\frac{\log M}{\epsilon^2}\right)$ measurements. Since quantum states collapse after a measurement, we need

$$O\left(\frac{M \log M}{\epsilon^2}\right)$$

copies and measurements of the quantum state. By the Chernoff Bound,

$$\Pr\Big[|\hat{O}_k - \mathrm{Tr}(O_k\rho)| < \epsilon\Big] \geq 1 - \frac{.01}{M}$$

such that $Pr[|\hat{O}_k - \mathrm{Tr}(O_k\rho)| < \epsilon, \forall k = 1, 2, \ldots M] \geq .99$ under the union bound where $\hat{O}_k$ is our reconstructed O_k under the algorithm. Naturally, it is not hard to see that the naive implementation is inefficient and quite costly, especially for large M. In the classical case where ρ is diagonal, we can get a linear speedup.

18.2.2 Classical Case

In the classical case, ρ is diagonal in the computational basis and its diagonal entries follow some probability distribution:

$$\rho = \begin{bmatrix} p_1 & & \\ & \ddots & \\ & & p_n \end{bmatrix}$$

For $i = 1, 2, \ldots, N$, we can then measure ρ in the computational basis $\{|b\rangle \langle b|\}_{b\in\{0,1\}^n}$ to obtain $b_i \in \{0,1\}^n$. Given observable O_k, we can predict $\mathrm{Tr}(O_k\rho)$ by using these b_i measurements as

$$\hat{O}_k = \frac{1}{N}\sum_{i=1}^{N} \langle b_i| O_k |b_i\rangle$$

Since $\mathbb{E}_{|b\rangle\sim\rho}[\langle b| O_k |b\rangle] = \mathrm{Tr}(O_k\rho)$ and $|\langle b| O_k |b\rangle| \leq 1$, we need $N = O(\frac{logM}{\epsilon^2})$ measurements to ensure that $Pr[|\hat{O}_k - \mathrm{Tr}(O_k\rho)| < \epsilon] \geq 1 - \frac{.01}{M}$, or equivalently under the union bound that $Pr[|\hat{O}_k - \mathrm{Tr}(O_k\rho)| < \epsilon, \forall k = 1, 2, \ldots M] \geq .99$. Therefore in the case where ρ is diagonal in the computational basis, we see a significant improvement in both efficiently and cost. However, in the general case where ρ may not be diagonal we would not be able to utilize this algorithm. Instead, we can follow a similar procedure using shadow tomography to get similar efficiency to the classical case.

18.3 Shadow Tomography

The procedure for shadow tomography is similar to that of the classical case, but instead of assuming that ρ is diagonal and measuring in the computational basis directly we first evolve ρ under some unitary U. More precisely, for $i = 1, \ldots, N$, sample U_i under some distribution over $SU(2^n)$[1] then evolve $\rho \to U_i\rho U_i^\dagger$ and measure in the computational basis to obtain b_i. We then store (U_i, b_i) classically. Since ρ cannot be assumed to be diagonal, it is crucial to first evolve ρ under U_i in order for the diagonal terms of $U_i\rho U_i^\dagger$ to have contributions from the off-diagonal terms of ρ.

For concreteness, let us see an example of shadow tomography where the unitaries are sampled over the uniform distribution of $Cl(2^n)$[2] which can be efficiently simulated classically. Given O_k, for $i = 1, \ldots, N$ compute

$$X_i = (2^n + 1)\,\mathrm{Tr}\Big(O_k U_i^\dagger |b_i\rangle\langle b_i| U_i\Big) - \mathrm{Tr}(O_k)$$

Using these X_i, our reconstruction of $\mathrm{Tr}(O_k\rho)$ is

$$\hat{O}_k = \text{Median-of-Means}(X_1, \ldots, X_N)$$

Median-of-Means: Suppose X_i is a random variable with variance $\mathrm{Var}[X_i] = \sigma^2$. By the Chebyshev, $\Pr\left[|\frac{1}{L}\sum_{i=1}^{L} X_i - \mathbb{E}[X]| < \epsilon\right] \geq 1 - \frac{\sigma^2}{L\epsilon^2}$. Equivalently, for $Y = \frac{1}{L}\sum_{i=1}^{L} X_i$ and

[1] $SU(2^n)$ is the set of unitaries that can be generated by any combination of Hadamard, $\pi/8$ or CNOT gates. Recall these set of gates achieve universality and thus it is an infinite set. Practical applications of shadow tomography typically use smaller, finite sets such as $Cl(2^n)$.

[2] $Cl(2^n)$ is the Clifford group whose unitaries are generated by combinations of Hadamard, phase and CNOT gates. The Clifford group has the property that any circuit with these gates can be reduced to one with linear depth, hence the set is finite.

$L = \frac{10\sigma^2}{\epsilon^2}$, $\Pr[|Y - \mathbb{E}[Y]| < \epsilon] \geq .9$. The Median-of-Means of the X_i is then defined as $Z = \text{Median}(Y_1, \ldots, Y_{N/L})$.

To see that this construction of $\hat{O}_k$ is indeed within ϵ error of $\text{Tr}(O_k\rho)$, define

$$G_s = \begin{cases} 1 & |Y_s - \mathbb{E}[Y]| < \epsilon \\ 0 & \text{else} \end{cases}$$

Now so long as at least half the Y_s are such that $|Y_s - \mathbb{E}[Y]| \leq \epsilon$, then our Median-of-Means Z is close to $\mathbb{E}[Y]$. Specifically, we have

$$\Pr[|Z - \mathbb{E}[Y]| < \epsilon] \geq \Pr\left[\frac{1}{N/L}\sum_{s=1}^{N/L} G_s > .5\right] \geq 1 - \delta$$

for $\frac{N}{L} = O(\log\frac{1}{\delta})$. All together, this construction of $\hat{O}_k$ accurately predicts $\mathbb{E}[Y] = \mathbb{E}[X]$ with $N = L \cdot O(\log\frac{1}{\delta}) = O(\frac{\sigma^2}{\epsilon^2}\log\frac{1}{\delta})$ measurements. Lastly, we want to show that $\mathbb{E}[X] = \text{Tr}(O_k\rho)$ and this can be seen through state design.

18.3.1 State Design

For a uniform distribution over pure states, μ_H, we have the following properties:

1. First moment: $\int d\mu_H \, |\psi\rangle\langle\psi| = \frac{\mathbb{1}}{2^n}$.
2. Second moment: $\int d\mu_H \, |\psi\rangle\langle\psi|^{\otimes 2} = \frac{\mathbb{1}\otimes\mathbb{1}+\text{SWAP}}{2^n(2^n+1)}$.
3. Third moment: $\int d\mu_H \, |\psi\rangle\langle\psi|^{\otimes 3} = \frac{1}{2^n(2^n+1)(2^n+2)}\sum_{\pi\in S_3} W_\pi$.

where $|\psi\rangle$ are pure states, $\text{SWAP}(A\otimes B) = B\otimes A$ and S_3 is the symmetric group over 3 elements. W_π for the third moment is then the 6 different ways of permuting 3 elements. We can use the second moment to show that $\mathbb{E}[X] = \text{Tr}(O_k\rho)$, and we can use the third moment to show that $\text{Var}[X] \leq 3\,\text{Tr}\big(O_k^2\big)$. Thus, the variance is bounded and does not scale with the system size.

Remarks:

1. If $N = O\left(\max_k \frac{\text{Tr}(O_k^2)\log M}{\epsilon^2}\right)$, we can predict $\text{Tr}(O_1\rho), \text{Tr}(O_2,\rho), \ldots, \text{Tr}(O_M\rho)$ within ϵ error.
2. $\mathbb{E}[X] = \mathbb{E}_U\left[\sum_{b\in\{0,1\}^n}\langle b|\,U\rho U^\dagger\,|b\rangle\,[(2^n+1)\,\text{Tr}\big(O_kU^\dagger\,|b\rangle\,\langle b|\,U\big) - \text{Tr}(O_k)]\right]$.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 19

March 30, 2022 *Scribes: Richard Allen, Raluca Vlad*

Today's lectures notes are on Hamiltonian simulation using quantum walks.

19.1 Hamiltonian simulation using quantum walks

19.1.1 Setup

We discussed how to do Hamiltonian simulation for local Hamiltonians using the Trotter method, but only with simulation time $O(t^2)$ for a time evolution of length t. We should be able to do this simulation in linear time, since this is how nature does it. How can we do so?

$$e^{iHt} \xrightarrow{\text{Trotter}} O(t^2).$$
$$?? \to O(t).$$

In this and the following lecture, we will prove we can achieve linear time simulation using the quantum walks framework. Essentially, we interpret a Hamiltonian as a generalization of a symmetric random walk. The method in fact applies to sparse Hamiltonians (that include local Hamiltonians) with oracle access as detailed below (see also Lecture 17).

We will assume that it takes unit cost to make oracle calls. Let's say we are given a Hamiltonian H_A on some N-dimensional space A, and say that H_A is s-sparse. Also, say we are given an oracle O_e that, on input $|x,y\rangle_{AB}$ and $|00\ldots\rangle$, outputs $|x,y\rangle_{AB}$, as well as the (x,y) entry $|H_{xy}\rangle$ of the Hamiltonian.

$$|x,y\rangle_{AB} \otimes |00\cdots\rangle_{\text{ancillae}} \longrightarrow \boxed{O_e} \longrightarrow |x,y\rangle_{AB} \otimes |H_{x,y}\rangle$$

Using the bonus part of homework 7, we construct the oracle O_s that, on input $|x\rangle_A |0\rangle_B$, outputs

$$\frac{1}{\sqrt{s}} \cdot \sum_{y:H_{xy}\neq 0} |x,y\rangle_{AB}.$$

That is, we have the oracle:

$$|x\rangle_A |0\rangle_B \longrightarrow \boxed{O_s} \longrightarrow \tfrac{1}{\sqrt{s}} \textstyle\sum_{y:H_{x,y}\neq 0} |x,y\rangle_{AB}$$

It will be useful to utilize the eigendecomposition of H:

$$H = \sum_J E_J \cdot |\phi_J\rangle\langle\phi_J|.$$

Finally, define $\alpha = \max_{xy}(|H_{xy}|)$.

19.1.2 Recall: Quantum Walks

Say we are given a Hermitian stochastic matrix with entries P_{xy}. This matrix must be real-valued, so Hermiticity is equivalent to symmetry, ie. $P_{xy} = P_{yx}$. We defined the following quantum states:

$$\left|\psi_x^1\right\rangle = \sum_y \sqrt{P_{xy}} \left|x\right\rangle \left|y\right\rangle$$

$$\left|\psi_y^2\right\rangle = \sum_x \sqrt{P_{yx}} \left|x\right\rangle \left|y\right\rangle$$

We also considered the projections:

$$\Pi_1 = \sum_x \left|\psi_x^1\right\rangle\!\left\langle\psi_x^1\right|$$

$$\Pi_2 = \sum_y \left|\psi_y^2\right\rangle\!\left\langle\psi_y^2\right|$$

The quantum walk operation was $(2\Pi_1 - \mathbb{1})(2\Pi_2 - \mathbb{1})$. And we could compute $\Pi_1\Pi_2$ using the fact that

$$\left\langle\psi_x^1|\psi_y^2\right\rangle = \sqrt{P_{xy}P_{yx}} = P_{xy}.$$

19.1.3 Hamiltonian Simulation: High Level

Today, we want to prove the following theorem:

Theorem In the setting from above, there exists a quantum walk Q_H which

(1) makes $O(\frac{s\alpha t}{\epsilon})$ calls to the oracles O_e, O_s;

(2) for any input state $\left|\omega\right\rangle_A$ satisfies:

$$\|Q_H \left|\omega\right\rangle_A \otimes \left|0\right\rangle_{\text{ancillary}} - e^{iHt} \left|\omega\right\rangle_A \otimes \left|\cdots\right\rangle\| \leq \epsilon.$$

At a high level, we will be thinking of our Hamiltonian H_{xy} as a symmetric stochastic matrix P_{xy} and designing a quantum walk operator V that has information about the eigenvalues E_j of H. This procedure has some issues:

1. $\sum_y H_{xy} \neq 1$, so H is not truly stochastic.
2. H_{xy} may have complex or negative entries.

Ignoring these complications for the moment, the simulation algorithm acting on an eigenstate $\left|\phi_j\right\rangle$ of H would act as follows:

$$\left|\phi_J\right\rangle_A \xrightarrow{PE_V + \text{controlled rotation}} e^{iE_jt} \left|\phi_J\right\rangle_A \left|\cdots\right\rangle \xrightarrow{\text{undo } PE_V} e^{iE_jt} \left|\phi_J\right\rangle_A$$

Since phase estimation works in superposition, we can decompose an arbitrary state in the eigenbasis of H and run the same algorithm:

$$\left|\omega\right\rangle_A = \sum_J \beta_J \left|\phi_J\right\rangle_A$$

$$\left|\omega\right\rangle_A \longrightarrow \sum_J \beta_J e^{iE_Jt} \left|\phi_J\right\rangle_A = e^{iHt} \left|\omega\right\rangle_A$$

19.1.4 Hamiltonian Simulation: Details

Having seen a high level picture of Hamiltonian simulation using random walks, we now describe the details of the implementation.

Consider the following quantum state (where A,B are registers of size 2^N for an $N \times N$ Hamiltonian and C,D are qubit registers)

$$\left|\psi_x^1\right\rangle = \frac{1}{\sqrt{s\alpha}} \cdot \sum_y \sqrt{H_{yx}} \left|0,x\right\rangle_{CA} \left|y,0\right\rangle_{BD} + \left|0\right\rangle_C \left|\xi_x\right\rangle_{AB} \left|1\right\rangle_D ,$$

where the second term inside the summation takes care that the state is normalized. Similarly, define

$$\left|\psi_y^2\right\rangle = \frac{1}{\sqrt{s\alpha}} \cdot \sum_x \sqrt{H_{xy}} \left|0,x\right\rangle_{CA} \left|y,0\right\rangle_{BD} + \left|1\right\rangle_C \left|\xi_y'\right\rangle_{AB} \left|0\right\rangle_D .$$

We will leave $\left|\xi_x\right\rangle, \left|\xi_y'\right\rangle$ unspecified, as their exact form will not matter. In the following claim and its proof, we characterize the complexity of creating $\left|\psi_x^1\right\rangle, \left|\psi_y^2\right\rangle$.

Claim 19.1.1. There exist unitaries T^1 and T^2 such that

$$\left|\psi_x^1\right\rangle = T^1 \left|0\right\rangle_C \left|x\right\rangle_A \left|0\right\rangle_B \left|0\right\rangle_D$$

$$\left|\psi_y^2\right\rangle = T^2 \left|0\right\rangle_C \left|y\right\rangle_A \left|0\right\rangle_B \left|0\right\rangle_D$$

with T^1, T^2 each making one call to each of $O_s, O_e, O_e^\dagger$.

To prove this claim, start with $\left|x\right\rangle_A \left|0\right\rangle_B$ and perform the following steps (adding ancillary registers as needed):

$$\begin{aligned}
\left|x\right\rangle_A \left|0\right\rangle_B &\xrightarrow{O_s} \frac{1}{\sqrt{s}} \cdot \sum_{y:H_{xy}\neq 0} \left|x,y\right\rangle_{AB} \otimes \left|00\cdots 0\right\rangle \\
&\xrightarrow{\text{SWAP } O_e \text{ SWAP}} \frac{1}{\sqrt{s}} \cdot \sum_{y:H_{xy}\neq 0} \left|x,y\right\rangle_{AB} \otimes \left|H_{yx}\right\rangle \left|0\right\rangle_D \\
&\xrightarrow{\text{Rejection sampling}} \frac{1}{\sqrt{s}} \cdot \sum_{y:H_{xy}\neq 0} \left|x,y\right\rangle_{AB} \otimes \left|H_{yx}\right\rangle \left(\sqrt{\frac{H_{yx}}{\alpha}} \left|0\right\rangle_D + \sqrt{1-\frac{|H_{yx}|}{\alpha}} \left|1\right\rangle_D\right) \\
&\xrightarrow{O_e^\dagger} \frac{1}{\sqrt{s}} \cdot \sum_{y:H_{xy}\neq 0} \left|x,y\right\rangle_{AB} \otimes \left|00\cdots 0\right\rangle_{\text{ancillas}} \left(\sqrt{\frac{H_{yx}}{\alpha}} \left|0\right\rangle_D + \sqrt{1-\frac{|H_{yx}|}{\alpha}} \left|1\right\rangle_D\right) \\
&= \frac{1}{\sqrt{s\alpha}} \cdot \sum_y \sqrt{H_{yx}} \left|x,y\right\rangle_{AB} \left|0\right\rangle_D + \left|\xi_x\right\rangle_{AB} \left|1\right\rangle_D ,
\end{aligned}$$

where to get the last line we dropped the extra ancillary registers and we defined

$$\left|\xi_x\right\rangle_{AB} := \sqrt{\alpha - |H_{yx}|} \left|x,y\right\rangle_{AB} .$$

This gives us a "quantum walk on $(\mathbb{C}^2 \otimes \mathbb{C}^N) \otimes (\mathbb{C}^2 \otimes \mathbb{C}^N)$".

By adding a qubit register C, we recover $\left|\psi_x^1\right\rangle$. The procedure for generating $\left|\psi_y^2\right\rangle$ is similar - the only difference is we use the register C for rejection sampling and add D in the final step.

We will now define our quantum walk operator. Let

$$\Pi_1 = \sum_x \left|\psi_x^1\right\rangle\!\left\langle\psi_x^1\right|$$
$$\Pi_2 = \sum_x \left|\psi_y^2\right\rangle\!\left\langle\psi_y^2\right|$$

With these definitions, the product of the projectors $\Pi_1\Pi_2$ is

$$\Pi_1\Pi_2 = \sum_{xy} \left|\psi_x^1\right\rangle \left\langle\psi_x^1|\psi_y^2\right\rangle \left\langle\psi_y^2\right|$$

where we can quickly check that the inner product is

$$\left\langle\psi_x^1|\psi_y^2\right\rangle = \frac{1}{s\alpha}(\sqrt{H_{yx}})^*\sqrt{H_{xy}}.$$

Note that despite the presence of the additional states $\left|\xi_x\right\rangle, \left|\xi_y'\right\rangle$, this inner product correctly captures the Hamiltonian entries, due to orthogonality of the qubit registers.

In general, H_{yx} may be complex. However, if we choose our square roots carefully, this equals $\frac{H_{xy}}{s\alpha}$, so $\Pi_1\Pi_2$ is

$$\Pi_1\Pi_2 = \frac{1}{s\alpha}\sum_{xy} H_{xy}\left|\psi_x^1\right\rangle\left\langle\psi_y^2\right|$$

which is just the Hamiltonian (times $1/(s\alpha)$) in another basis! Hence, the singular values of $\Pi_1\Pi_2$ are the same as the eigenvalues of H (again, times $1/(s\alpha)$). Using our definitions of the unitaries T^1 and T^2, we can alternatively write this as

$$\Pi_1\Pi_2 = T^1\left(\left|0\right\rangle\!\left\langle 0\right|_C \otimes \frac{H_A}{s\alpha} \otimes \left|0\right\rangle\!\left\langle 0\right|_B \otimes \left|0\right\rangle\!\left\langle 0\right|_D\right)(T^2)^\dagger$$

Now define the quantum walk operator by

$$W = (2\Pi_1 - I)(2\Pi_2 - I).$$

The eigenvalues of W are $\{e^{\pm 2i\theta_J}\}$, for $\cos\theta_J = \frac{E_J}{s\alpha}$ the singular values of $\Pi_1\Pi_2$. Furthermore, the Jordan blocks of the decomposition of Π_1 and Π_2 are defined by the following vectors:

$$T_1\left|0\right\rangle\left|\phi_J\right\rangle\left|00\right\rangle \;\;,\;\; T_2\left|0\right\rangle\left|\phi_J\right\rangle\left|00\right\rangle$$

for $\left|\phi_J\right\rangle$ an eigenstate of H, since these are the left and right singular vectors of $\Pi_1\Pi_2$, as the decomposition above shows.

Now we consider the full quantum walk algorithm applied to an eigenstate of H:

1. Apply T^1 to map from $\left|\phi_J\right\rangle$ to one of the Jordan blocks:

$$\left|0\right\rangle_C\left|\phi_J\right\rangle_A\left|0\right\rangle_B\left|0\right\rangle_D \xrightarrow{T^1} T^1\left|0\right\rangle_C\left|\phi_J\right\rangle_A\left|0\right\rangle_B\left|0\right\rangle_D.$$

2. Perform phase estimation of the unitary W:

$$T^1 |0\rangle_C |\phi_J\rangle_A |0\rangle_B |0\rangle_D \xrightarrow{PE_W} a |\mu_+\rangle |2\theta_J\rangle + b |\mu_-\rangle |-2\theta_J\rangle$$

   where $|\mu_\pm\rangle$ is the eigenstate of W in the Jth Jordan block with eigenvalue $e^{\pm 2i\theta_J}$.

3. Apply a controlled phase conditioned on the second register:

$$a |\mu_+\rangle |2\theta_J\rangle + b |\mu_-\rangle |-2\theta_J\rangle \xrightarrow{\text{CTRL Phase}} e^{is\alpha t\cos(\theta_J)}(a |\mu_+\rangle |2\theta_J\rangle + b |\mu_-\rangle |-2\theta_J\rangle).$$

4. Undo phase estimation:

$$\begin{aligned} e^{is\alpha t\cos(\theta_J)}(a |\mu_+\rangle |2\theta_J\rangle + b |\mu_-\rangle |-2\theta_J\rangle) &\xrightarrow{PE_W^{-1}} e^{is\alpha t\cos(\theta_J)}(a |\mu_+\rangle + b |\mu_-\rangle) \\ &= e^{is\alpha t\cos(\theta_J)} T^1 |0\rangle_C |\phi_J\rangle_A |0\rangle_B |0\rangle_D \\ &= e^{iE_J t} T^1 |0\rangle_C |\phi_J\rangle_A |0\rangle_B |0\rangle_D \end{aligned}$$

This demonstrates that we can use quantum walks to perform Hamiltonian simulation given an eigenstate of the Hamiltonian. Since phase estimation works in superposition, this algorithm works given any input state. In the following lecture, we will see a detailed analysis of the total number of oracle calls necessary to achieve a precision ε, which we will find to be $O(s\alpha t/\varepsilon)$. This is linear in t, as promised.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 20

April 8, 2022 *Scribes: Tao Lin, Bill Zhai*

- (The rest of) Hamiltonian simulations.
- Quantum signal processing (QSP).

20.1 Preliminary: Chebyshev polynomials

By trigonometry, $\cos k\theta$ and $\sin k\theta$ can be written as polynomials of $\cos\theta$ and $\sin\theta$. These polynomials are called **Chebysev polynomials**:

- A Chebyshev polynomial of the first kind, T_k, is a degree k polynomial given by

$$T_k(\cos\theta) = \cos k\theta.$$

  Let $x = \cos\theta$. We can write $T_k(x) = \cos k(\cos^{-1} x)$.

- A Chebyshev polynomial of the second kind, S_{k-1}, is a degree $k-1$ polynomial given by

$$S_{k-1}(\cos\theta) = \frac{\sin k\theta}{\sin\theta}.$$

For example:

$$\begin{aligned} &\cos\theta = \cos\theta && k=1: && T_1(x) = x \\ &\cos 2\theta = 2\cos^2\theta - 1 && k=2: && T_2(x) = 2x^2 - 1 \\ &\cos 3\theta = \cdots && \cdots && \cdots \end{aligned}$$

and

$$\begin{aligned} S_0(x) &= \frac{\sin\theta}{\sin\theta} = 1 \\ S_1(x) &= \frac{\sin 2\theta}{\sin\theta} = \frac{2\sin\theta\cos\theta}{\sin\theta} = 2x. \end{aligned}$$

Chebyshev polynomials were discovered 200 years ago by Chebyshev. They turned out to be pretty useful in approximation theory and now appear in quantum computing, like Grover's search.

Properties Here are two properties of Chebyshev polynomials that are worth mentioning:

1. $\forall x \in [-1,1]$, $|T_k(x)| \le 1$.
2. - $T_k(x)$ is a "$k \mod 2$" parity polynomial, meaning that all the powers of x in $T_k(x)$ are even if k is even, and odd if k is odd.
   - $S_k(x)$ (or $S_{k-1}(x)$) is also a "$k \mod 2$" (or "$k-1 \mod 2$") parity polynomial.

20.2 Hamiltonian simulation

Recall Recall that we have a Hamiltonian H, which is an $N \times N$ Hermitian matrix with eigen-decomposition $H = \sum_J E_J |\phi_J\rangle\langle\phi_J|$, states $|\psi_x^1\rangle, |\psi_y^2\rangle \in \mathcal{H}_{CA} \otimes \mathcal{H}_{BD}$ where $\mathcal{H}_A, \mathcal{H}_B$ are both N-dimensional, $\mathcal{H}_C, \mathcal{H}_D$ are both 2-dimensional. The key point is

$$\langle\psi_x^1|\psi_y^2\rangle = \frac{H_{xy}}{s\alpha} \tag{20.1}$$

where s is the sparsity and α is the largest entry of H. We have the quantum walk operator W:

$$W = (2\Pi_1 - \mathbb{1})(2\Pi_2 - \mathbb{1}),$$

where

$$\Pi_1 = \sum_x |\psi_x^1\rangle\langle\psi_x^1|, \quad \Pi_2 = \sum_y |\psi_y^2\rangle\langle\psi_y^2|.$$

We said that Hamiltonian simulation (to approximate $e^{iHt}|\omega\rangle_A$ within ε error) can be achieved with $O(\frac{s\alpha t}{\varepsilon})$ calls to O_e, O_s, O_s^{-1}, using phase estimation (not proved yet).

Improvement by QSP One way to improve the $O(\frac{s\alpha t}{\varepsilon})$ bound is to use QSP. QSP gets rid of phase estimation and improves the bound to $O(s\alpha t + \log\frac{1}{\varepsilon})$, which turns out to be optimal.

Another usage of QSP is for the QLS (quantum linear system) problem. Recall that in QLS, we had a matrix A with condition number κ, sparsity s. We assumed that its largest entry $\alpha = 1$. We talked about a procedure that achieves a cost of order $O(\frac{\kappa^2 s}{\varepsilon})$, using phase estimation. QSP and its generalization instead achieve $O(\kappa s \log\frac{1}{\varepsilon})$.

We will (in Section 20.3) talk about the general idea of how QSP is used in the above problems. We will not go into the details.

20.2.1 Finishing the proof for Hamiltonian simulation

We will show the $O(\frac{s\alpha t}{\varepsilon})$ bound. Recall that

$$|\psi_x^1\rangle = T^1 |0\rangle_C |x\rangle_A |0\rangle_B |0\rangle_D,$$
$$|\psi_y^2\rangle = T^2 |0\rangle_C |y\rangle_A |0\rangle_B |0\rangle_D.$$

We find the (non-zero) singular values of $\Pi_1\Pi_2$:

$$\begin{aligned}
\Pi_1\Pi_2 &= \sum_{x,y} |\psi_x^1\rangle \langle\psi_x^1|\psi_y^2\rangle \langle\psi_y^2| \overset{(20.1)}{=} \sum_{x,y} \frac{H_{xy}}{s\alpha} |\psi_x^1\rangle \langle\psi_y^2| \\
&= T^1 \Big(|0\rangle\langle 0|_C \otimes \frac{H}{s\alpha} \otimes |0\rangle\langle 0|_B \otimes |0\rangle\langle 0|_D \Big) \big(T^2\big)^\dagger \\
&= \sum_J \underbrace{\frac{E_J}{s\alpha}}_{\text{a singular value}} \underbrace{T^1 |0\rangle_C |\phi_J\rangle_A |0\rangle_B |0\rangle_D}_{\text{a left singular vector}} \underbrace{\langle 0|_C \langle\phi_J|_A \langle 0|_B \langle 0|_D \big(T^2\big)^\dagger}_{\text{a right singular vector}}.
\end{aligned} \tag{20.2}$$

Reminder: singular value decomposition (SVD). The SVD of a matrix M is $M = \sum_J \sigma_J |\mu_J\rangle\langle\nu_J|$, where $\sigma_J > 0$ is a singular value, $|\mu_J\rangle$ is called a left singular vector, $|\nu_J\rangle$ is called a right singular vector. $|\mu_J\rangle$'s are mutually orthogonal. $|\nu_J\rangle$'s are mutually orthogonal.

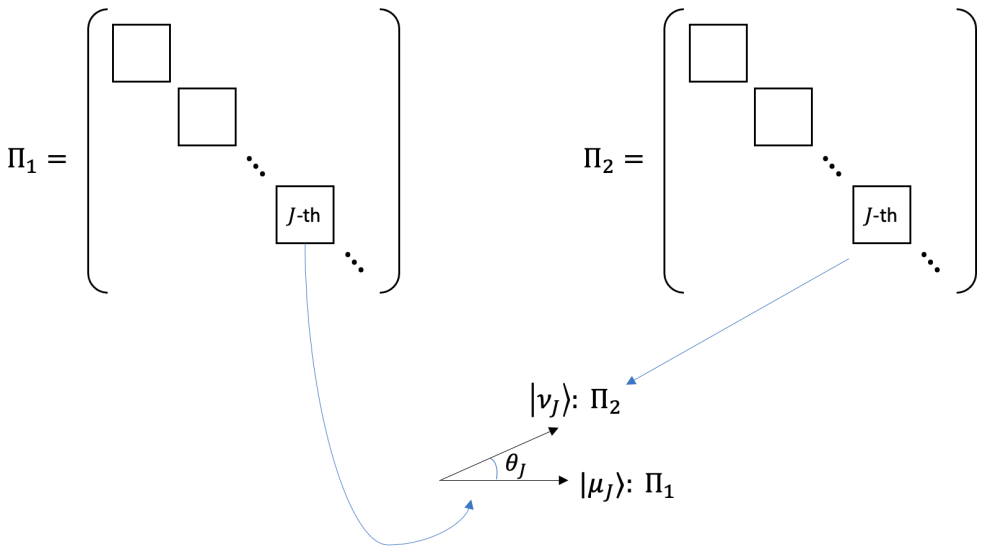


Figure 20.1: Jordan decomposition: the J-th block

Now, we figure out the eigenvalues of $W = (2\Pi_1 - \mathbb{1})(2\Pi_2 - \mathbb{1})$. We do the jordan decomposition, so $\Pi_1 = \sum_J |\mu_J\rangle\langle\mu_J|$ and $\Pi_2 = \sum_J |\nu_J\rangle\langle\nu_J|$, as shown in Figure 20.1. The J-th block of $\Pi_1\Pi_2 = \sum_J |\mu_J\rangle \langle\mu_J|\nu_J\rangle \langle\nu_J|$ is

$$|\mu_J\rangle \underbrace{\langle\mu_J|\nu_J\rangle}_{\cos\theta_J} \langle\nu_J| \,,$$

from which we see that $|\mu_J\rangle$ is a left singular vector, $|\nu_J\rangle$ is a right singular vector, and $\langle\mu_J|\nu_J\rangle = \cos\theta_J$ is the singular value. Compared with (20.2), we see that the singular value $\cos\theta_J = \frac{E_J}{s\alpha}$. So, within the J-th Jordan block, the two vectors $|\mu_J\rangle$ and $|\nu_J\rangle$ have an angle θ_J such that

$$\cos\theta_J = \frac{E_J}{s\alpha}.$$

As discussed in previous lectures, the operator $(2\Pi_1 - \mathbb{1})(2\Pi_2 - \mathbb{1})$ acts as a rotation by angle $2\theta_J$ in the J-th 2D subspace, and thus has eigenvalues

$$e^{\pm 2i\theta_J},$$

with phases $\pm\theta_J$.

Procedure of Hamiltonian simulation The following is a sketch. See Lecture Note 19 or previous scribe notes for more details. The procedure works as follows:

$$\text{Input } |w\rangle_A = \sum_J \beta_J |\phi_J\rangle_A \xrightarrow[\text{apply } T^1]{\text{add registers}} T^1 |0\rangle_C |\omega\rangle_A |0\rangle_B |0\rangle_D = \sum_J \beta_J T^1 |0\rangle_C |\phi_J\rangle_A |0\rangle_B |0\rangle_D \,.$$

For the J-th component,

$$T^1 |0\rangle_C |\phi_J\rangle_A |0\rangle_B |0\rangle_D \xrightarrow{\text{PE}_W} \begin{array}{l} \text{get } \theta'_J \approx \theta_J; \\ \text{let } E'_J = s\alpha \cos\theta'_J; \\ \text{apply } e^{iE'_J t} \longrightarrow e^{iE'_J t} T^1 |0\rangle_C |\phi_J\rangle_A |0\rangle_B |0\rangle_D. \end{array}$$

Since the above works in superposition, we obtain $\sum_J \beta_J e^{iE'_J t} T^1 |0\rangle_C |\phi_J\rangle_A |0\rangle_B |0\rangle_D$. Then,

$$\sum_J \beta_J e^{iE'_J t} T^1 |0\rangle_C |\phi_J\rangle_A |0\rangle_B |0\rangle_D \xrightarrow[\text{remove } C,B,D]{\text{undo } T^1} \sum_J \beta_J e^{iE'_J t} |\phi_J\rangle_A \approx \sum_J \beta_J e^{iE_J t} |\phi_J\rangle_A = e^{iHt} |\omega\rangle_A .$$

Analysis To obtain an ε-error Hamiltonian simulation, we can require

$$|E'_J - E_J| \le \frac{\varepsilon}{2t} \tag{20.3}$$

because $|e^{ix'} - e^{ix}| \le 2|x' - x|$. Because $|E'_J - E_J| = s\alpha|\cos\theta'_J - \cos\theta_J| \le s\alpha|\theta'_J - \theta_J|$, (20.3) is satisfied when

$$|\theta'_J - \theta_J| = O(\frac{\varepsilon}{s\alpha t}).$$

This error of phase estimation can be achieved with cost $O(\frac{s\alpha t}{\varepsilon})$.

20.3 Quantum signal processing

In retrospect, all our effort in Hamiltonian simulation is to apply the function $e^{iE_J t}$ in front of an input state. What quantum signal processing does is to provide a more general scheme for applying any polynomials of $\cos\theta_J$ (or $\cos 2\theta_J$, as we will see). Then we can apply those complicated functions like $e^{iE_J t} = e^{is\alpha\cos\theta_J t}$ by polynomial approximation.

Quantum signal processing is similar to what we did in quantum linear systems, where we wanted to apply function $f(A)$ to a state $|b\rangle$:

$$f(A) |b\rangle , \quad f(z) = \frac{1}{\kappa z},$$

$$A = \sum_i E_i |\phi_i\rangle\langle\phi_i| , \quad f(A) = \sum_i f(E_i) |\phi_i\rangle\langle\phi_i| .$$

Phase estimation has the power to do more interesting functions $f(A)$ than what we have seen so far. Quantum signal processing is slightly less general, in the sense that it can only do polynomials. But, it is more efficient, with only $O(\text{degree})$ cost.

(Main message:) QSP applies polynomials of $\cos\theta$ with $O(\text{degree})$ cost.

20.3.1 Applications of QSP

As a hint of how QSP can be useful, we look at the example of Hamiltonian simulation: in the J-th Jordan block, we want to apply the function e^{iE_Jt}:

$$e^{iE_Jt} = e^{is\alpha t\cos\theta_J} \overset{\text{(Taylor expansion)}}{=} \sum_{\ell=0}^{\infty} \frac{(is\alpha t\cos\theta_J)^\ell}{\ell!} \approx \sum_{\ell=0}^{d} \frac{(is\alpha t\cos\theta_J)^\ell}{\ell!}$$

where the approximation error is

$$\text{error} \approx \frac{(s\alpha t)^d}{d!} \overset{\text{(Stirling's approximation)}}{\approx} \Big(\frac{es\alpha t}{d}\Big)^d \le \varepsilon$$

if we choose

$$d = 4s\alpha t + \log\frac{1}{\varepsilon}.$$

This gives the $O(s\alpha t + \log\frac{1}{\varepsilon})$ bound we claimed in Section 20.2.

As another hint, for the QLS problem, we want to apply $\frac{1}{\kappa s\cos\theta}$ (since we want to apply $f(E_i) = \frac{1}{\kappa E_i} = \frac{1}{\kappa s\alpha\cos\theta_i} = \frac{1}{\kappa s\cos\theta_i}$ where we have assumed that the largest entry $\alpha = 1$). Let $x = \kappa s\cos\theta_i$. We approximate $\frac{1}{x}$ by $\frac{1-(1-x^2)^d}{x}$.[1] If x is in some "good" range, then the degree d here is roughly $O(ks\log\frac{1}{\varepsilon})$, as claimed in Section 20.2.

20.3.2 Formal description of QSP

Introduction Recall that $\big|\psi_x^1\big\rangle = T^1\,|0\rangle_C\,|x\rangle_A\,|0\rangle_B\,|0\rangle_D$, and $\Pi_1 = \sum_x \big|\psi_x^1\big\rangle\!\big\langle\psi_x^1\big|$. How do we implement $2\Pi_1 - \mathbb{1}$? To answer that, we write

$$\begin{aligned}\Pi_1 = \sum_x \big|\psi_x^1\big\rangle\!\big\langle\psi_x^1\big| &= T^1\Big(\sum_x |0\rangle\!\langle 0|_C \otimes |x\rangle\!\langle x|_A \otimes |0\rangle\!\langle 0|_B \otimes |0\rangle\!\langle 0|_D\Big)(T^1)^{-1}\\ \text{(because } \sum_x |x\rangle\!\langle x|_A = \mathbb{1}_A\text{)} \ &= T^1\Big(|0\rangle\!\langle 0|_C \otimes \mathbb{1}_A \otimes |0\rangle\!\langle 0|_B \otimes |0\rangle\!\langle 0|_D\Big)(T^1)^{-1}\end{aligned}$$

and

$$2\Pi_1 - \mathbb{1} = T^1\Big(2\,|0\rangle\!\langle 0|_C \otimes |0\rangle\!\langle 0|_B \otimes |0\rangle\!\langle 0|_D - \mathbb{1}_{CBD}\Big) \otimes \mathbb{1}_A (T^1)^{-1}.$$

So, to implement $2\Pi_1 - \mathbb{1}$ we can first apply $(T^1)^{-1}$, then do a reflection about $|0\rangle_C \otimes |0\rangle_B \otimes |0\rangle_D$ (in the C, B, D registers), and finally apply T^1.

Can we do something more general than $2\Pi_1 - \mathbb{1}$? Looking at the middle step, we have the following transformation:

$$|0\rangle_C\,|0\rangle_B\,|0\rangle_D \longrightarrow |0\rangle_C\,|0\rangle_B\,|0\rangle_D\,;$$

for any state $|\theta\rangle_{CBD}$ that is orthogonal to $|0\rangle_C\,|0\rangle_B\,|0\rangle_D$,

$$|\theta\rangle_{CBD} \longrightarrow -\,|\theta\rangle_{CBD}\,.$$

We can generalize this: instead of -1, we can apply any arbitrary phase to $|\theta\rangle_{CBD}$:

$$|0\rangle_C\,|0\rangle_B\,|0\rangle_D \longrightarrow |0\rangle_C\,|0\rangle_B\,|0\rangle_D\,;$$

[1] This is different from what we get from Taylor expansion (at $x_0 = 1$), which is $\sum_{n=0}^{d-1}(-1)^n(x-1)^n = \frac{1-(1-x)^d}{x}$. The approximation $\frac{1-(1-x^2)^d}{x}$ turns out to be better for some specific range of x.

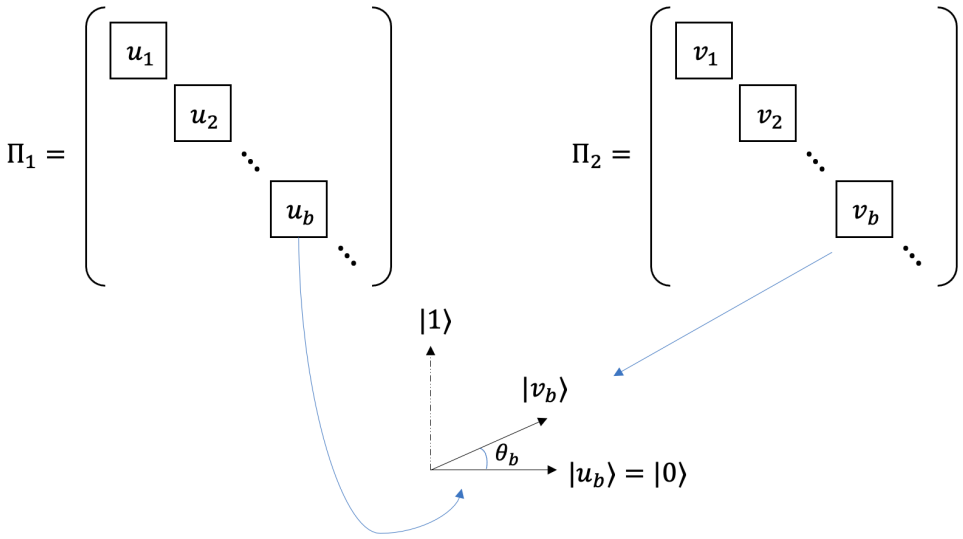


Figure 20.2: Jordan decomposition: the b-th 2D block

$$|\theta\rangle_{CBD} \longrightarrow e^{i\phi} |\theta\rangle_{CBD} .$$

In this manner, we achieve the following more general form of "reflection":

$$\Pi_1 + e^{i\phi}(\mathbb{1} - \Pi_1) \ =: \ U_\phi.$$

In particular, when $\phi = \pi$ we recover the original reflection $2\Pi_1 - \mathbb{1}$.

Our goal We ask the following question: instead of $W = (2\Pi_1 - \mathbb{1})(2\Pi_2 - \mathbb{1})$, what will happen when we apply the following operator $W_{\phi_k,\phi_{k-1},\ldots,\phi_1}$?

$$W_{\phi_k,\phi_{k-1},\ldots,\phi_1} = U_{\phi_k}(2\Pi_2 - \mathbb{1})U_{\phi_{k-1}}(2\Pi_2 - \mathbb{1}) \cdots U_{\phi_1}(2\Pi_2 - \mathbb{1}). \tag{20.4}$$

Quantum signal processing provides understanding for this question, which is our goal.

Let's begin with the example of $\phi_k = \cdots = \phi_1 = \pi$, where $W_{\pi,\ldots,\pi} = \big((2\Pi_1 - \mathbb{1})(2\Pi_2 - \mathbb{1})\big)^k$. What does the matrix $\big((2\Pi_1 - \mathbb{1})(2\Pi_2 - \mathbb{1})\big)^k$ do inside the b-th Jordan block (Figure 20.2)? Recall that $(2\Pi_1 - \mathbb{1})(2\Pi_2 - \mathbb{1})$ is the rotation by angle $2\theta_b$. So, $\big((2\Pi_1 - \mathbb{1})(2\Pi_2 - \mathbb{1})\big)^k$ is the rotation by angle $2k\theta_b$. Using $\{|u_b\rangle, |u_b^\perp\rangle\}$ as basis ($\{|0\rangle, |1\rangle\}$), the matrix in the b-th block is written as

$$\begin{bmatrix} \cos 2k\theta_b & \sin 2k\theta_b \\ -\sin 2k\theta_b & \cos 2k\theta_b \end{bmatrix}.$$

Using Chebyshev polynomials $T_k(\cos 2\theta_b) = \cos 2k\theta_b$ and $S_{k-1}(\cos 2\theta_b) = \frac{\sin 2k\theta_b}{\sin 2\theta_b}$, the above matrix becomes

$$= \begin{bmatrix} T_k(\cos 2\theta_b) & S_{k-1}(\cos 2\theta_b)\sin 2\theta_b \\ -S_{k-1}(\cos 2\theta_b)\sin 2\theta_b & T_k(\cos 2\theta_b) \end{bmatrix}.$$

Formal statements Here is the general theorem of quantum signal processing:

Theorem 20.3.1 (QSP Part 1)**.** *The matrix $W_{\phi_k,\phi_{k-1},\ldots,\phi_1}$ looks like*

$$\begin{bmatrix} P_k(\cos 2\theta) & Q_{k-1}(\cos 2\theta)\sin 2\theta \\ -Q_{k-1}^*(\cos 2\theta)\sin 2\theta & P_k^*(\cos 2\theta) \end{bmatrix}$$

(where P_k^*, Q_{k-1}^* are the complex conjugates of P_k, Q_{k-1}) in a 2D block with angle θ, such that

1. *P_k, Q_{k-1} are "$k \mod 2$", "$k-1 \mod 2$" parity polynomials, of degree k, $k-1$, respectively.*
2. *$|P_k(\cos 2\theta)|^2 + |Q_{k-1}(\cos 2\theta)\sin 2\theta|^2 = 1$.*
3. *$|P_k(\cos 2\theta)|^2 \leq 1$.*

Note: The first property is similar to the property of Chebyshev polynomials. It can be proved by induction. The second property holds true because the matrix is unitary. The third property also follows from unitarity, because the norm of any entry in a unitary matrix is at most 1.

What is more surprising is the part 2 of the quantum signal processing theorem, which says that the converse direction is also true: for any polynomials that satisfy the above three properties, we can find angles $\phi_k, \phi_{k-1}, \ldots, \phi_1$ that generate a matrix with those polynomials.

Theorem 20.3.2 (QSP Part 2, [LYC16])**.** *Given polynomials P_k, Q_{k-1} such that*

1. *P_k, Q_{k-1} are "$k \mod 2$", "$k-1 \mod 2$" parity polynomials, of degree k, $k-1$, respectively.*
2. *$|P_k(x)|^2 + |Q_{k-1}(x)|^2(1-x^2) = 1$.*
3. *$|P_k(x)|^2 \leq 1 \ \forall x \in [-1,1]$.*

Then $\exists\ \phi_k, \phi_{k-1}, \ldots, \phi_1$ such that $W_{\phi_k,\phi_{k-1},\ldots,\phi_1}$ looks like

$$\begin{bmatrix} P_k(\cos 2\theta) & Q_{k-1}(\cos 2\theta)\sin 2\theta \\ -Q_{k-1}^*(\cos 2\theta)\sin 2\theta & P_k^*(\cos 2\theta) \end{bmatrix}$$

in a 2D block with angle θ.

With this theorem, once we take P_k, Q_{k-1} to be the Taylor expansions of some functions, we can find the corresponding phases $\phi_k, \phi_{k-1}, \ldots, \phi_1$, and then do k "reflections", each of the form $U_{\phi_k}(2\Pi_2 - \mathbb{1})$.

The rough idea of the proof of Theorem 20.3.2 is as follows. According to condition 1, P_k and Q_{k-1} have the form:

$$P_k(x) = a_k x^k + a_{k-2}x^{k-2} + \cdots, \quad Q_{k-1}(x) = b_{k-1}x^{k-1} + b_{k-3}x^{k-3} + \cdots.$$

The condition $|P_k(x)|^2 + |Q_{k-1}(x)|^2(1-x^2) = 1$ implies that the x^{2k}-term in the polynomial $|P_k(x)|^2 + |Q_{k-1}(x)|^2(1-x^2)$ must have coefficient 0, which gives

$$|a_k|^2 - |b_{k-1}|^2 = 0.$$

So, we have

$$a_k = e^{i\phi_k} b_{k-1}$$

for some phase ϕ_k, which is what we want. We then reduce the degrees of P_k and Q_{k-1} and use induction to find the remaining phases $\phi_{k-1}, \ldots, \phi_1$. (See more details in Lecture Note 20.)

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 21

April 11, 2022 *Scribe: Ted Pyne*

In quantum complexity theory, we ask **"which Hamiltonians are hard?"** We have seen algorithms that show *simulating* local Hamiltonians is not hard, but it may still be hard to precisely compute their ground energy. Soon, we will discuss computing approximate ground energies, which is often sufficient. Towards this, we will discuss the idea of lower bounds in more generality, which correspond to showing a model (like quantum circuits) cannot do a good job at estimating ground energy. For today's lecture, we will first review Pauli matrices, then discuss classical error-correcting codes, then begin to explore quantum error correction.

In upcoming lectures, we will discuss specifically Gottesman-Knill Theorem, variational quantum algorithms, NLTS conjecture, the quantum PCP conjecture, and quantum circuit lower bounds.

21.1 Pauli Matrices Review

Recall the Pauli X and Z matrices, where we fix the standard computational basis $\{|0\rangle, |1\rangle\}$:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Furthermore recall (and it is easy to show) that $ZX = -XZ$ (i.e. the matrices anticommute) and $X^2 = Z^2 = I$. Because of this, for any eigenvalue λ of either unitary we have $\lambda^2 = 1 \implies \lambda \in \{-1, 1\}$.

Rather than considering Pauli matices acting on a single qubit, we can define their extension to multiple qubits:

Definition 21.1.1. An **n-bit Pauli unitary** is an unitary of the form

$$X_1 \otimes X_2 \otimes I_3 \otimes \cdots \otimes X_n$$

or

$$Z_1 \otimes I_2 \otimes Z_3 \otimes \cdots \otimes X_n$$

where the unitary acts as X or Z respectively on a subset of the bits $[n]$ and acts as the identity on the others. For $a \in \{0,1\}^n$, we denote

$$X^a \stackrel{\text{def}}{=} \bigotimes_{i=1}^{n} X_i^{a^{(i)}}$$

where $X^0 = I$ and Z^b is defined analogously.

We now prove some basic properties of multi-bit Paulis. For the remainder of the section fix $a, b \in \{0,1\}^n$ and let . denote the inner product.

Fact 21.1.2. We have

$$X^a Z^b = (-1)^{a.b} Z^b X^a.$$

Proof. For every index i where $a^{(i)} = b^{(i)} = 1$ we have that $X_i^{a^{(i)}} Z_i^{b^{(i)}} = -Z_i^{b^{(i)}} X_i^{a^{(i)}}$ and otherwise we have $X_i^{a^{(i)}} Z_i^{b^{(i)}} = Z_i^{b^{(i)}} X_i^{a^{(i)}}$, so the number of minus signs "picked up" is precisely $a.b$. □

Fact 21.1.3. Letting $|w\rangle = |w_1\rangle \dots |w_n\rangle$, we have

$$X^a |w\rangle = |a \oplus w\rangle, \qquad Z^b |w\rangle = (-1)^{b.w} |w\rangle.$$

Proof. First, for an index i where $X_i^{a^{(i)}} = X$ we flip that position of w, which can be expressed as taking the XOR of w with a. Second, for an index i where $Z_i^{b^{(i)}} = Z$, if $w_i = 1$ we pick up a minus sign and otherwise do not, so the number of minus signs is $b.w$. □

Finally, note that since X, Z have eigenvalues restricted to $\{-1, 1\}$, it is easy to write down the projections onto the relevant eigenspaces. For instance, the projections onto the $\lambda = 1, -1$ eigenspaces of X^a are given by

$$\frac{I + X^a}{2}, \qquad \frac{I - X^a}{2}$$

and an analogous fact holds for Z^b.

21.1.1 Concurrent Measurement

Recall that on a single bit, we cannot measure with respect to the eigenvalues of of X, Z simultaneously because they do not commute (as $XZ = -ZX$). However, simultaneous measurement is sometimes possible on more than one qubit. For instance, the 2-qubit unitaries $X \otimes X$ and $Z \otimes Z$ *do* commute by Fact 21.1.2. In fact, to simultaneously measure we can use a special basis consisting of 4 projection operators:

$$\{P^{\pm,\pm}\} = \left\{ \frac{I \otimes I \pm X \otimes X}{2} \cdot \frac{I \otimes I \pm Z \otimes Z}{2} \right\}.$$

We will exploit this ability to measure simultaneously while constructing error correcting codes.

21.2 Error Correction Codes

We first review some elements of classical error correction. For (many) more details, see the CS 229R notes.

Definition 21.2.1. A **(linear)** n, k, d **code** is defined by a **parity check matrix** $\mathbf{H} \in \mathbb{F}_2^{n-k \times n}$. Let a_i be the ith row of $\mathbf{H}$. The codewords are defined as

$$C = \{w : \mathbf{H}w = 0\}$$

i.e. the dot product of w with a_i is 0 mod 2 for all i. The number of codewords is 2^k (by a dimension argument), and the **distance** d is the minimum Hamming weight of a nonzero codeword.

The defintion of distance above comes from the ability of codes to detect errors - given a codeword w and $e \in \{0,1\}^n$ of Hamming weight at most $d-1$, we have $H(w \oplus e) = He \neq 0$, and hence if we receive the corrupted message $w+e$ we can notice an error has occurred. Note that if e is allowed to have Hamming weight equal to the distance or greater, we have no hope of correcting all possible errors, as we can have $w + e = w'$ for $w' \in C$.

For example, we can fix $n = 5$ and consider the **repetition code (rep code)**, specified by the following parity check matrix:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

One can see that the only two codewords are $\{00000, 11111\}$.

21.3 Quantum Error Correcting Codes

In the classical world, our error vector e flipped at most $d-1$ bits in our received message. In the quantum world, our model of corruption is different - we instead wish to detect the application of an *arbitrary unitary* that acts on only a few qubits, which seems to be a much harder problem. To illustrate a further challenge in constructing quantum EC codes, we can first attempt to define a rep code in quantum-world. Fix n and consider a code that takes

$$|0\rangle \to |0\rangle_1 \dots |0\rangle_n , \qquad |1\rangle \to |1\rangle_1 \dots |1\rangle_n .$$

Since our code should be used to transmit an arbitrary quantum state, one natural question is what should $|+\rangle$ be transmitted as? Unfortunately, sending it to $|+\rangle^n$ violates the no-cloning theorem. Even worse, how do we analyze this code's vulnerability to uncountably-many possible errors?

To solve the second problem, we use that an arbitrary unitary can be written as a linear combination of Pauli unitaries, so it suffices to analyze the code's vulnerability to Pauli operator corruption. To analyze this, we first cast the classic rep code in the language of quantum.

21.3.1 The Quantum Rep Code

A (quantum) code is defined as the **stabilizer** of a set of unitaries - that is, quantum states that are fixed by every member of the set. For concreteness, for now we focus on defining the (classical) rep code in this manner. The set of stabilizers we use is as follows:

$$\Pi_C = \text{Stab}\{Z^{a^{(1)}}, \dots, Z^{a^{(n-1)}}\} = \{|w\rangle : Z^{a^{(i)}} |w\rangle = |w\rangle \, \forall i\}$$

where a_i is the ith row of the classical parity check matrix. Recall that $Z^{a^{(i)}} |w\rangle = (-1)^{a.w}$ by Fact 21.1.2, so the only computational basis vectors in the code are $|0\rangle^n$ and $|1\rangle^n$ The code is thus $\text{Span}\{|0\rangle^n, |1\rangle^n\}$.

Now, given a vector $|w\rangle$ we can measure in the basis $\{Z^{a^{(1)}}, \dots, Z^{a^{(n-1)}}\}$. If we measure in the 1 eigenspace (rather than -1) with respect to all elements of the set, we have that the recieved word is in the code, and if we measure -1 the recieved word is not in the code.

To cast classical bit flip errors in this language, recall that $X^e |w\rangle = |w \oplus e\rangle$, so flipping the bits specified by the vector e can be detected if $Z^{a^{(i)}} X^e \neq X^e Z^{a^{(i)}}$ for some i. It is easy to see

that any error with Hamming weight at most $n-1$ is detected by some $Z^{a^{(i)}}$, exactly analogous to the classical definition. For an example of a bit flip that cannot be detected by our current set of constraints, we can consider

$$X_1 \otimes X_2 \otimes \ldots \otimes X_n.$$

Finally, we outline why this code is **not** a good quantum code. Consider the two codewords produced by input states $|+\rangle$ and $|-\rangle$ respectively. These are

$$\text{“}|+\rangle'' = \frac{|0\rangle^n + |1\rangle^n}{\sqrt{2}}, \qquad \text{“}|+\rangle'' = \frac{|0\rangle^n - |1\rangle^n}{\sqrt{2}}$$

Unfortunately, there is a unitary acting on a *single* qubit that takes the $|+\rangle$ encoding to the $|-\rangle$ encoding - the unitary $Z_1 \otimes I_2 \otimes \ldots \otimes I_n$. As such, this code has distance 1, and cannot detect a single error. In the language of quantum codes, we cannot detect this error because it commutes with $Z^{a^{(i)}}$ for all i. Luckily, we will see in the next lecture that adding a single additional element to our set, X^{1^n}, will solve this problem.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 22

April 15, 2022 *Scribes: Weiyu Li*

Roadmap

Today we will focus on

1. Recap of classical codes and their quantum view
2. CSS codes
3. Logical operators
4. How to encode quantum data
5. Local indistinguishability

22.1 Recap

Throughout this note, we consider n-bit binary code, denoted as $C \subset \{0,1\}^n$. Also, the operations are under $\mathbb{F}_2$, e.g., $1+1=0$.

Definition 22.1.1 (Linear code)**.** We say an error correcting code C is **linear**, if for any $w, w' \in C$, we have $w+w' \in C$. A linear code C can be generated from the span of its **basis** $\{b_1, b_2, \ldots, b_k\}$ as

$$C = \left\{ \sum_{i=1}^{k} \alpha_i b_i : \alpha_i \in \{0,1\}, i = 1,2,\ldots,k \right\}.$$

An equivalent definition is from the **parity check matrix**

$$H = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-k} \end{bmatrix} \in \{0,1\}^{(n-k)\times n},$$

where a_i is the i-th row vector of H. Then the code is the null space of H, written as

$$C = \{w : Hw = 0\} = \{w : a_i w = 0, \forall i \in [n-k]\}.$$

The number of codewords is 2^k, and the **distance** of the linear code is the minimum Hamming weight of the $2^k - 1$ nonzero codewords.

For $a \in \{0,1\}^n$, we use the notations

$$X^a \stackrel{\text{def}}{=} \bigotimes_{i=1}^{n} X_i^{a^{(i)}}, Z^b \stackrel{\text{def}}{=} \bigotimes_{i=1}^{n} Z_i^{b^{(i)}}$$

to denote specific n-bit Paulis with only X or Z operators. It is not hard to show that

$$X^a Z^b = (-1)^{a.b} Z^b X^a, \tag{22.1}$$

where . denotes the inner product of two vectors in $\{0,1\}^n$. More generally speaking, n-bit Paulis (defined in Definition 21.1.1) either commute or anti-commute.

Definition 22.1.2 (Quantum code)**.** Analogous to the parity check matrix in classical linear code, a set of Paulis defines a quantum code. In specific, a quantum code is the subspace **stabilized**[1] by the set, that is, the codewords are unchanged after applying every unitary in the set. A code can be represented by its **stabilizers**.
The **weight** of a multi-bit Pauli is the number of non-identity one-bit qubits it has. Then the **(quantum) distance** of a quantum code is the minimum weight of the non-identity Paulis that commute with all stabilizers but are not part of the stabilizers.

For example, the code with stabilizers $\{Z^{a_1}, \ldots, Z^{a_{n-k}}\}$ is

$$\Pi_C = \text{Stab}\{Z^{a_1}, \ldots, Z^{a_{n-k}}\} = Span\{|w\rangle = |w_1\rangle \otimes \cdots \otimes |w_n\rangle : Z^{a_i} |w\rangle = |w\rangle, \forall i \in [n-k]\}$$

where a_i is the i-th row of the classical parity check matrix. From (22.1), X^b commutes with all stabilizers if and only if $a_i.b = 0$ for all i, which is exactly $b \in C$ in the classical view. Thus the quantum distance of Π_C is exactly the same as the classical distance of C Also, X^b can be visualized as a codeword via $X^b |0\rangle = |b\rangle$. In fact, the linearity guarantees that X^b maps codewords to codewords as $X^b |w\rangle = |w+b\rangle$. That said, X^b is a logical operator in the code space.

Now, let's cast classical bit flip errors in this quantum language. Suppose X^e satisfies $a_1.e = 1$, then for any codeword $w \in C$,

$$Z^{a_1} X^e |w\rangle = (-1)^{a_1.e} X^e Z^{a_1} |w\rangle = -X^e |w\rangle = -|w+e\rangle,$$

which implies that flipping the bits specified by the vector e can be detected by the change of eigenvalue (from 1 to -1) of Z^{a_1}. Classically, the above equation can be viewed as

$$H(w+e) \stackrel{Hw=0}{=} He = \begin{bmatrix} 1 \\ \vdots \end{bmatrix} \neq 0.$$

In a nutshell, errors anticommute with at least one stabilizer, while logical errors (logical changes between codewords) commute with all stabilizers.

[1]The word "stabilizer" comes from group theory. We say a state is stabilized by a set of unitaries, or the unitaries are the stabilizer of the state space.

22.2 CSS Codes

Suppose there is a unitary Z^e and a (quantum) codeword $|b\rangle \in \Pi_C$ such that $e.b = 1$, then

$$Z^e(|0\rangle^{\otimes n} + |b\rangle) = |0\rangle^{\otimes n} - |b\rangle .$$

Since $|0\rangle^{\otimes n}$ and $|b\rangle$ are both in the 1 eigenspace of all the stabilizers, the error caused by Z^e cannot be detected from measuring the eigenvalues with respect to the existing stabilizers. Thus, we need to introduce new stabilizer(s) that (i) commutes with all other stabilizers and (ii) detects Z^e.

We claim that adding X^b in the set of stabilizers satisfies this, since b is a codeword orthogonal to a_i's, and $X^b(|0\rangle^{\otimes n} \pm |b\rangle) = \pm(|0\rangle^{\otimes n} + |b\rangle)$.

With the new stabilizers $\{Z^{a_1}, \ldots, Z^{a_{n-k}}, X^b\}$, the code changes, since $X^b |w\rangle = |w + b\rangle \neq |w\rangle$ for any $w \in C$. However, note that $|w\rangle + |w + b\rangle$ (unnormalized) is still a codeword; in fact, the new code can be characterized by

$$Span\{|w\rangle + |w + b\rangle : w \in C\}.$$

In general, we can define the CSS code, which is the initials of Calderbank, Shor and Steane.

Definition 22.2.1 (CSS code)**.** Let H_1, H_2 be linear subspaces such that $H_1 \perp H_2$. Let C_1 be the subspace orthogonal to H_1 (we can think of C_1 as a classical code and H_1 as the span of its parity check matrix). Note that $H_2 \subset C_1$. **CSS code** has stabilizers $\{Z^{a_1}, \ldots, Z^{a_{n-k}}, X^{b_1}, \ldots, X^{b_l}\}$, where $a_1, \ldots, a_{n-k} \in H_1$, $b_1, \ldots, b_l \in H_2$ are linearly independent. Then CSS code consists of the stabilized states, whose code space is

$$\Pi_{C'} = Span\{\sum_{b \in H_2} |w \oplus b\rangle : w \in C_1\}$$

with dimension 2^{k-l}.
Detectable errors are unitaries $Z^{e'}, X^{d'}$ that do not commute with at least one of the stabilizers, while **logical errors** are Z^e, X^d that commute with all stabilizers.

From the definition, the distance of a code is the minimum weight of logical errors.

Example 22.2.2 (Trivial CSS code)**.** Suppose Z_i, X_i are Z and X operators on the i-th qubit. Consider stabilizers $Z_1, \ldots, Z_{n-k}, X_{n-k+1}, \ldots, X_{n-k+l}$. One can show that the logical operators are

$$\{X_{n-k+l+1}, Z_{n-k+l+1}, \ldots, X_n, Z_n\},$$

and the codewords are as trivial as

$$\left\{ \bigotimes_{i=1}^{n-k} |0\rangle_i \bigotimes_{j=n-k+1}^{n-k+l} |+\rangle_j \bigotimes |\psi\rangle : \psi \text{ is a } (k-l)\text{-dimension state} \right\} := \Pi_{\text{trivial}}.$$

Although the code space is trivial and any error on the last $k - l$ qubits cannot be detected and are all logical operators, it turns out that any quantum code is a rotation of this trivial one! More concretely, we have the following theorem.

Theorem 22.2.3. *Any quantum code* Π *can be written as*

$$\Pi = U \Pi_{trivial} U^\dagger,$$

where the unitary U *is a* **Clifford unitary** *and can be generated by Hadamard, Phase, and CNOT gates.*

22.3 Logical Operators

Let's go one step back. This section will be an independent discussion and can be skipped. Recall the definition of qubits. A qubit, for the purpose of error correction, is a pair of X and Z operators. Note that $Y = iXZ$ and one can get any operator from X and Z. In an algebraic point of view, a qubit is a pair of operators (P, Q) that satisfies

$$PQ = -QP, \quad P^2 = Q^2 = I.$$

Thus, the way to identify qubits in the code space, is to identify the pairs of operators exhibiting the above algebra. This is captured by the following theorem.

Theorem 22.3.1. *If the code has dimension 2^{k-l}, then there exists $\{P_i, Q_i\}_{i=1}^{k-l}$ such that*

(i) $P_i Q_j = \begin{cases} -Q_j P_i, & \text{if } i = j, \\ Q_j P_i, & \text{if } i \neq j. \end{cases}$

(ii) P_i, Q_i commute with all stabilizers, but cannot be represented as simply a product of stabilizers.

Here, $\{P_i, Q_i\}$ are called the logical operators or logical errors which we briefly touched upon in the context of CSS codes.

Having identified the logical operators, quantum gates can be defined correspondingly. For example, Hadamard gate on the i-th qubit maps P_i to Q_i and vice versa.

22.4 Local Indistinguishability

Last, we give a property which is the central tool for circuit lower bounds in quantum complexity.

Theorem 22.4.1 (Local indistinguishability)**.** *For a code $\Pi_{C'}$ and two codewords $|\phi\rangle, |\psi\rangle \in \Pi_{C'}$, suppose S is a set of qubits which do not contain any undetectable error, that is, any error on the qubits in S could be detected, where the number of qubits in S is smaller than the distance of $\Pi_{C'}$. Then tracing out everything except S, we have*

$$\mathrm{Tr}_{[n]\setminus S}(|\phi\rangle) = \mathrm{Tr}_{[n]\setminus S}(|\psi\rangle).$$

Proof. A one-line explanation is from no-cloning theorem: if qubits in S are erased, then information about the encoded state can be recovered from the remaining qubits. But then S has no information about the encoded state, as required by no-cloning theorem.
More formally, we need to show that $\langle\phi| R_S |\phi\rangle = \langle\psi| R_S |\psi\rangle$ for any Pauli R_S acting on S. We discuss two different cases of R_S.

1. R_S commutes with all stabilizers. Since the size of S is smaller than the distance of the code, R_S is generated from the stabilizers (product of stabilizers). Otherwise R_S will be a logical operator, which is a contradiction. As $|\phi\rangle$ and $|\psi\rangle$ are stabilized codewords, we have

$$R_S |\phi\rangle = |\phi\rangle \quad \Rightarrow \quad \langle\phi| R_S |\phi\rangle = \langle\phi|\phi\rangle = 1 = \langle\psi| R_S |\psi\rangle .$$

2. R_S anti-commutes with at least one of the stabilizers, e.g., Z^a. Then we have

$$\langle\phi| R_S |\phi\rangle = \langle\phi| R_S Z^a |\phi\rangle = -\langle\phi| Z^a R_S |\phi\rangle = -\langle\phi| R_S |\phi\rangle \quad \Rightarrow \quad \langle\phi| R_S |\phi\rangle = 0 = \langle\psi| R_S |\psi\rangle .$$

□

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 23

April 20, 2022 *Scribes: Ratip Emin Berker*

Roadmap

Today we will be focusing on:

- Quantum PCP conjecture (QPCP)
  - Recap of Local Hamiltonian Problem (LHP)
- Description complexity
  - Approaches for relating it to QPCP

We will also talking about Local Indisguishablility and Clifford Circutis.

23.1 Local Hamiltonian Problem

Recall that in the Local Hamiltonian Problem ($LHP[b-a,m]$), we have a **Prover** and a **Verifier**. We are also given n qubits and a k-local hamiltonian H on these qubits with $k = O(1)$ and $H = \sum_{\alpha=1}^{m} b_\alpha P_\alpha$ where $m \leq n^k$ for some k, $b_\alpha \in (-1,1)$ and P_α are Pauli operators for all α. The goal is to decide if $E_1(H) < a$ or $E_1(H) \geq b$ where $E_1(H)$ is the ground state energy (smallest eigenvalue) of H and it is guaranteed that one of these cases is true. A protocol is as follows: The Prover send the Verifier the ground state $|\psi\rangle$. The Verifier measures the energy $\langle\psi| H |\psi\rangle$ up to $\pm\frac{b-a}{100}$ precision. Recall that Kitaev showed in 1999 that $LHP[\frac{O(1)}{m^3}, m]$ is QMA complete.

23.2 Quantum PCP Conjecture

Quantum PCP Conjecture: There exists a constant $\epsilon > 0$ such that $LHP[\epsilon m, m]$ is QMA complete.

The main difference of the conjecture from the proof by Kitaev ($LHP[\frac{1}{m^3}, m]$) is that now we are allowed to measure the energy up to a much larger asymtotatic value, $O(m)$ instead of $O(1/m^3)$, even though ϵ is small (e.g. 10^{-35}). Note that in order for the conjecture to be true, we would need two parts:

- $LHP[\epsilon m, m]$ is in QMA (Easy to show).
- $LHP[\epsilon m, m]$ is in QMA-hard (Hard, open problem).

What we do know about the hardness is:

Theorem 23.2.1. *$LHP[\epsilon m, m]$ is NP-hard.*

Unlike the QPCP, this is indeed proven, but it is still a highly nontrivial theorem. It's proof follows from the classical PCP Theorem, which is from 1992.

How do we approach the open problem of showing that $LHP[\epsilon m, m]$ is QMA-hard? One method is "Gap amplification", which involves starting with any constant-local Hamiltonian H (with m terms, i.e. $H = \sum_{\alpha=1}^{m} b_\alpha P_\alpha$) and mapping/reducing it to some other Hamiltonian H' (with m' terms, i.e. $H' = \sum_{\alpha=1}^{m'} b_\beta P_\beta$) through some efficent transformation such that we have one of these two cases:

- **Case 1:** $E_0(H) \approx 0 \Rightarrow E_0(H') \approx 0$
- **Case 2:** $E_0(H) \geq \frac{1}{m^3} \Rightarrow E_0(H') \geq \epsilon \cdot m'$

Notice that if this reduction could indeed be done efficiently, then the QMA-hardness would be shown, as we would solve $LHP[\frac{1}{m^3}, m]$ (which is proven to be QMA-hard) on H by tranforming it to H' and solving $LHP[\epsilon m', m']$ on H', implying that the latter must be QMA-hard. The main difficulty in this reduction is to identify a small amount of energy present in the system ($E_0(H) \geq \frac{1}{m^3}$) and then amplify it to a much larger energy ($E_0(H') \geq \epsilon m'$).

As mentioned above, Quantum PCP conjecture is a Quantum analog of the classical PCP theorem, which shows that any NP protocol to prove $x \in L$ can be reduced to a 3SAT instance such that if the proof is accepted, then the 3SAT is satisfiable, and if the proof is rejected, then $\geq \frac{1}{8} + \epsilon$ fraction of the clauses are unsatisfiable. This is a powerful transformation that, on an intuitive level, enables us to accept/reject a proof by checking the satisfiability at a few places, rather than checking all of them.

As discussed, showing that $LHP[\epsilon m, m]$ is QMA-Hard is difficult. As a warm up, we will be exploring the question of ruling out all ways which would put $LHP[\epsilon m, m]$ in NP. In other words, we are trying to prove that any person who claims that "the witness for this problem is easy (i.e. a simple NP proof rather than a ground state) " is wrong. In order to this, we will have to discuss the notion of Description Complexity.

23.3 Description Complexity

As a warm up, consider the following questions:
Q1: You are given $|\psi_1\rangle\langle\psi_1|$ as a quantum state on n-qubits. How many complex numbers do you need in order to describe it?
Answer: You need $2^n - 1$. But perhaps in *some* special quantum states, we need less (see next question).
Q2: Say instead that our state is the tensor product n of single qubit states: $|\psi_2\rangle\langle\psi_2| = |\phi_1\rangle\langle\phi_1| \otimes |\phi_2\rangle\langle\phi_2| \otimes ... \otimes |\phi_n\rangle\langle\phi_n|$ where each $|\phi_i\rangle$ lives in $\mathbb{C}^2$. How many complex numbers would you need to describe this state?
Answer: In order to describe this state, you need just $2n$ complex numbers, 2 amplitudes per each $|\phi_i\rangle\langle\phi_i|$.

$|\psi_1\rangle\langle\psi_1|$ and $|\psi_2\rangle\langle\psi_2|$ are both quantum states on n qubits, but $|\psi_2\rangle\langle\psi_2|$ is much easier to describe. This is what description complexity is about (i.e. how easy it is to describe a quantum state). We present various notions to approach Description Complexity:

Notion 1: Circuit complexity. Consider a state $|\psi\rangle$ that is a result of applying a quantum circuit with depth t to n qubits initially set to $|0\rangle^{\otimes n}$ Note that the number of complex numbers we need to describe $|\psi\rangle$ (i.e. the description complexity of $|\psi\rangle$) is $\boxed{nt \times O(1)}$ where $O(1)$ comes from the number of complex numbers required to specify a unitary gate acting on two qubits (2 times 2 matrix), which is a simpler way to describe the state rather than describing the whole n qubit unitary. Note that $|\psi_2\rangle$ described in **Q2** above is a special case of this, as it can be obtained by starting from $|0\rangle^{\otimes n}$ and applying a single rotation (hence $t = 1$). Most quantum states, however, require t to be much larger, as it can be seen form the fact that $2^n - 1$ should be of same complexity as $ntO(1)$ in the worst case.

Notion 2: Stabilizer Rank. This is a slightly less natural notion than Circuit complexity. Recall that Clifford circuits are made of H (Hadamard), CNOT, and Phase gates, and they are circuits that take Paulis to Paulis. For instance:

$$HXH^\dagger = Z \tag{23.1}$$

$$HZH^\dagger = X \tag{23.2}$$

$$CNOT(X \otimes I)CNOT = X \otimes X \tag{23.3}$$

The whole study of Paulis are simply linear algebra, which implies that the study of Clifford circuits is also about linear algebra and is efficient. Esspecially:

Theorem 23.3.1 (Gottesman-Knill Theorem)**.** *All Clifford circuits are classically simulatable.*

In other words, if you apply a Clifford circuit to a $|0\rangle^{\otimes n}$ state, you can compute the outcome probabilities with a classical computer. On the other hand, Shor's Algorithm, most likely, is not classically simulatable (and hence cannot be done with a Clifford circuit, as far as we know) as factoring a large number N cannot be efficiently done by a classical computer, as far as we know.

Say we apply a Clifford circuit to $|0\rangle^{\otimes n}$ qubits, and we get $|\theta\rangle$, which we call a stabilizer state, which is easy to represent, with $2n^2$ real numbers. In order to this , consider $ClZ_1Cl^\dagger$, $ClZ_2Cl^\dagger$,..., $ClZ_nCl^\dagger$ which are all Pauli operators and stabilize the state that we are interested in (since $|0\rangle^{\otimes n}$ is stabilized by each Z_i). In fact, a linear combination of such stabilizer states is also easy to study. The stabilizer rank of a state $|\psi\rangle$ is the minimum number R such that we can write $|\psi\rangle$ as a linear combination of R stabilizer state $|\psi\rangle = \sum_{\alpha=1}^{R} \mu_[\alpha] |\theta_\alpha\rangle$ where each $|\theta_\alpha\rangle = Cl_\alpha |0\rangle^{\otimes n}$ is a stabilizer state (a Clifford circuit Cl_α applied to the zero state). Note that each of these stabilizer states be described using $2n^2$ classical bits [1] so the overall state $|\psi\rangle$ needs $2n^2R$ classical bits to describe. Most quantum states will have R to be very large, the same way that ciruit complexity for most quantum states is high.

23.4 Connecting Description Complexity to QPCP

We now discuss how the notions of Description Complexity discussed above ties back into our attempts at ruling out ways of "disproving" the QPCP. Here is a theorem that provides one potential way of disproving QPCP:

[1] To see why this is the case, notice that each $Cl_\alpha |0\rangle$, is stabilized by the corresponding operators $Cl_\alpha Z_1 Cl_\alpha^\dagger, Cl_\alpha Z_2 Cl_\alpha^\dagger, \ldots, Cl_\alpha Z_n Cl_\alpha^\dagger$. Each $Cl_\alpha Z_i Cl_\alpha^\dagger$ is a Pauli operator, so can be described using two n-bit numbers a_α and b_α (as we can write the Pauli operator as $X^{a_\alpha} Z^{b_\alpha}$. Hence, $Cl_\alpha |0\rangle$ can be described using $2n^2$ classical bits ($2n$ per Pauli operator).

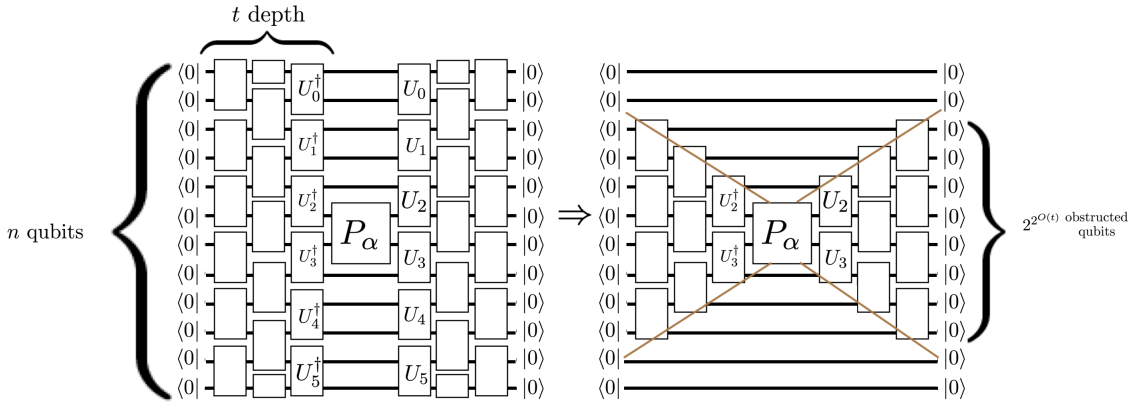
Theorem 23.4.1. *Fix any $\epsilon > 0$ and take any $H = \sum_\alpha b_\alpha P_\alpha$. Assume that for all such H (i.e. for all local Hamiltonians), $\exists \left|\psi\right\rangle$ with description complexity $D(\epsilon)$ (low description complexity) such that $\left\langle\psi\right| H \left|\psi\right\rangle \leq E_0 + \frac{\epsilon m}{2}$. In other words, for each local Hamiltonian, there is a low-energy state $\left|\psi\right\rangle$ that both achieves an energy as low as $\frac{\epsilon m}{2}$ above the ground energy and is also easy to describe. Then $LHP[\epsilon m, m]$ is NP (and hence QPCP is false).*

Proof. Assume that that the conditions given in the theorem is true. In that case, the prover can just sent the description of $\left|\psi\right\rangle$ rather than the quantum state itself. For instance, the prover could be sending the list of all the gates that generated $\left|\psi\right\rangle$, or the list of the coeffiecents μ_α for the corresponding stabilizer states $\left|\theta_\alpha\right\rangle$. The verifier can then efficently compute $\left\langle\psi\right| H \left|\psi\right\rangle = \sum_\alpha b_\alpha \left\langle\psi\right| P_\alpha \left|\psi\right\rangle$. If the energy it compuets is less than $\frac{\epsilon m}{2}$ then the verifier accepts (as the Hamiltonian must have ground state energy $E_0 \approx 0$ for this to be true), if it is more than ϵm, then the verifier rejects. Thus, $LHP[\epsilon m, m]$ is NP. □

One key assumption we made in the proof of the above theorem is that the verifier can then efficently (i.e. in polynomial time) compute $\left\langle\psi\right| H \left|\psi\right\rangle = \sum_\alpha b_\alpha \left\langle\psi\right| P_\alpha \left|\psi\right\rangle$ from the description of $\left|\psi\right\rangle$, which is not immediately obvious and needs further justification. This comes in the form of the following theorem:

Theorem 23.4.2. *There exists a classical algorithm that outputs $\left\langle\psi\right| P_\alpha \left|\psi\right\rangle$ in 2^{2^t} time. As a result, $\left\langle\psi\right| H \left|\psi\right\rangle = \sum_\alpha b_\alpha \left\langle\psi\right| P_\alpha \left|\psi\right\rangle$ can be computed in $m2^{2^t}$ time (where m is the number of terms in the sum), implying that if t is a constant, then this is an efficent algorithm.*

Proof. Lets consider the example of circuit depth complexity (the case of stabilizer states is also easy to prove, but we will not show it here). The proof is based on the idea of constructing what is called a "light cone". Fix a α and consider $\left\langle\psi\right| P_\alpha \left|\psi\right\rangle$. Recall that we have $\psi = U \left|0\right\rangle^{\otimes n}$ where U is a depth t quantum circuit. Hence, $\left\langle\psi\right| P_\alpha \left|\psi\right\rangle = \left\langle\psi\right|^{\otimes n} U^\dagger P_\alpha \left|0\right\rangle^{\otimes n}$. This setup is depicted in the figure below:



Since the Hamiltonian is k-local, then P_α has at most k non-identity Pauli matirces, and thus only prevent a certain number of components U_i within the final layer of U from meeting their conjugate $U_i^\dagger$ (in this example, it prevents $U_2^\dagger$ and U_2 to meet, but does not prevent $U_1^\dagger$ and U_1 to meet, so these matrices cancel by producing identity as their product (hence we can erase them from the cricuit). Simlarly all the components of the final layer of U that were obstructed by P_α from meeting their conjugate and hence becoming identity (in this example, U_2 and $U_2^\dagger$ as well as

U_3 and $U_3^\dagger$) will in turn obstruct a constant number of components in the second to last layer from meeting their conjugate and hence becoming identity, and so on. This local obstruction effect leads to a "light cone" as depicted in the figure above, where all the gates outside light cone meet with their conjugates as cancel out to identity, whereas all the gates that are within light cone may not, as they are obstructed by another gate in front of them. So if ψ_α is a $k - local$ term, it obstructs a constant number of gates, and so does the gates obstructed by it, and so on. Thus, as the light cone spreads out, roughly $2^O(t)$ qubits get obstructed, and all the remaining qubits can simply get together with their conjugates and produce 1 as their inner product. Hence, $\langle\psi| P_\alpha |\psi\rangle$ is a computation on $2^O(t)$ qubits, for which you need $2^{2^O(t)}$ time to compute, which is constant if t is a constant. Hence, you need $O(n \cdot 2^{2^O(t)})$ to compute the overall energy $\langle\psi| H |\psi\rangle = \sum_\alpha b_\alpha \langle\psi| P_\alpha |\psi\rangle$, which is indeed efficent. □

The above two theorems imply that if QPCP is indeed true, then we should not be able to find, for each local Hamiltonian H, a state $|\psi\rangle$ with both low description complexity description complexity ($D(\epsilon)$) and low energy ($\langle\psi| H |\psi\rangle \leq E_0 + \frac{\epsilon m}{2}$).

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 24 (Guest Lecture by Dr. Sona Najafi)

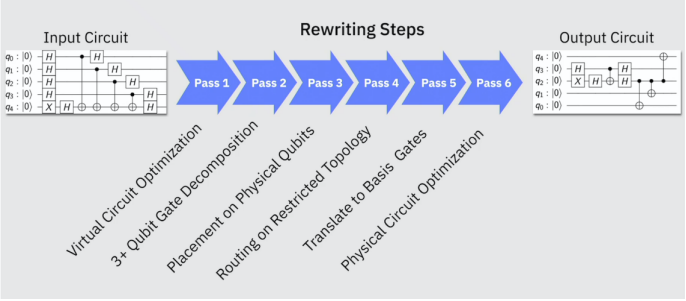
April 22, 2022 *Scribe: Elbert Du*

24.1 Qisket Basics

Qisket is a python library that allows you to simulate quantum computation and access IBM's quantum computers. We can prepare states and apply unitary gates with the package. The qisket package also allows you to draw the circuit (visualization).

24.1.1 Qisket Transpiler

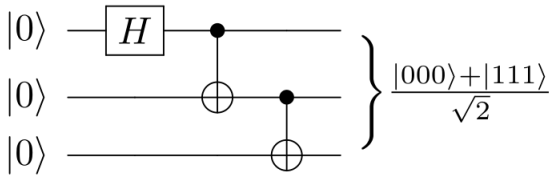
The transpiler converts the input circuit in your code to a physical circuit that can be run on the IBM hardware as follows:



24.1.2 Qisket Tutorials

Preparing the Bell and GHZ states

Recall that to prepare these basic states (and the corresponding state in n qubits), we start by initializing our qubits to the $|0\rangle$ state. Then, we apply the Hadamard to the first qubit and iteratively take the cnot with each other qubit. For example, the GHZ state (3 qubits) is prepared as in the following diagram:



For the full tutorials, see Jupyter notebooks in this Ed post: `https://edstem.org/us/courses/19209/discussion/1427518`

Current problems with quantum computing in practice: the hardware is currently extremely noisy. Current research on fixing this with error correcting codes and at the hardware level. We can use IBM's hardware by making an account here: `https://quantum-computing.ibm.com`

Things to note:

1. When running the "Hello_ibmq.ipynb" put 'ibm-q' in the hub and 'open' in the group to make it work without a paid subscription.
2. It's currently difficult to distinguish between queue time and runtime. It would be good to implement a way to do this in the future.

24.2 Hybrid Classical-Quantum Algorithms

24.2.1 Adiabatic Simulation

Given a hamiltonian H, we can break it into $H = sH_0 + (1-s)H_1$ where H_0 (the ground state) is easy to prepare. The idea is that if we start from the ground state H_0, we can gradually vary it until we get the desired H. This is basically what we're doing:

24.2.2 Variational Circuits

Variational Circuits are hybrid algorithms used to solve hard problems

General architecture:

1. Prepare an initial state
2. evaluate a quantum circuit with some parameters (free variables) θ
3. Measure the output and feed this into a classical optimization algorithm to optimize the parameters for the quantum circuit.
4. Feed the parameters to the quantum circuit.
5. Repeat until done

Many problems can be encoded as NP-complete optimization problems like Max Cut.

Example 24.2.1. QAOA Variational Form for solving Max Cut:

1. Prepare $|+\rangle^n$
2. Apply the mixing unitary $U(H_M) = e^{-i\beta H_M}$ where H_M is the mixing hamiltonian $\sum_{i\in[n]} \bigotimes I \otimes X_i$.
3. Apply problem unitary $U(H_p) = e^{-i\gamma H_p}$ where H_p is the problem hamiltonian $\sum_{(i,j)\in E} \bigotimes I \otimes Z_i \otimes Z_j$.
4. Measure, evaluate mean of cuts (running the above quantum circuit many times)

5. Optimize and update parameters (γ, β) for mixer hamiltonian and cost function respectively.

6. Repeat p times for some p.

If we repeat this iteratively, it should converge to the optimal solution. Even at a single iteration, if we have sufficiently many copies of the quantum circuit described in steps 1,2,3, this will give us the best cuts with high frequency for simple graphs.

CS 231 Quantum Computation and Quantum Complexity *Instructor: Anurag Anshu*

Lecture 25

April 27, 2022 *Scribe: Jothi Ramaswamy*

25.1 Quantum Error Correcting and Toric Codes

25.1.1 Main Takeaways from this class

If there is anything to takeaway from this class, it is the following two notions that have described many of the theorems and algorithms we have worked with this semester.

1. "Discreteness in continuous" - quantum states can be very complex, but we could always hold onto something discrete.
   (a) Jordan's Lemma is an example of this, which we have used in many of the quantum algorithms we have discussed so far, and have used briefly to discuss non-local games
   (b) We also recently discussed how much of quantum error correction condenses down to linear algebra in $\mathbb{F}_2^n$
2. "Static ideas in dynamics"
   (a) An example of this is hamiltonian simulation and quantum walks, boiled down to some simple quantum walk operator w.
   (b) We also discussed the Quantum Cook Levin Theorem as a general method to map computation to a hamiltonian.

25.1.2 Agenda

1. Recap of lectures 22 and 23
2. Search for hamiltonians with "hard" ground states.
3. Bill's discussion on Matrix Product States (MPS)
4. Toric Code

25.1.3 Recap: Description Complexity

During lectures 22 and 23, we discussed the notion of the description complexity of some $|\psi\rangle$. The description complexity is the number of bits needed to describe $|\psi\rangle$.

One way to measure this is to look at the quantum circuit complexity of $|\psi\rangle$, or $\text{CC}(|\psi\rangle)$. Quantum circuit complexity is the depth of the circuit you need to prepare this state from all zeros. If we say that $|\psi\rangle = U|0\rangle^{\otimes n}$, the quantum circuit complexity is how many gates went into U.

Another notion of description complexity is the stabilizer rank. The stabilizer rank is a way to write a quantum state $|\psi\rangle$ as a sum over stabilizer states which can be generated by some Clifford circuits. In other terms, $|\psi\rangle = \sum_{\alpha=1}^{R} \mu_\alpha CL_\alpha |0\rangle^{\otimes n}$. We generally want the rank to be low, so the smaller R is, the better.

The third notion is tensor network representation. They are usually helpful in what is known as a "spin chain," and this concept can also be seen as a matrix product state, which Bill discusses later in this lecture.

25.1.4 Recap on undetectable errors

One of the things that we learned from HW 10 was the concept that if we have some $|\psi\rangle, |\phi\rangle$, which are k-local indistinguishable (LI), then $\mathrm{CC}(|\psi\rangle) \geq \log k$ and $\mathrm{CC}(|\phi\rangle) \geq \log k$

In lecture 22, we also discussed quantum codes (QECC). We would have stabilizers $Z^{a_1}, Z^{a_2}, \cdots Z^{a_{n-k}} X^{b_1}, X^{b_2}, \cdots X^{b_l}$ where $a_i \in H_1$ and $b_i \in H_2$. The codestates are the set of $\{|\psi\rangle\}$ such that $Z^{a_i}|\psi\rangle = |\psi\rangle$ and $X^{b_i}|\psi\rangle = |\psi\rangle$. We also want Z^{a_i} and X^{b_i} to commute. We know that $Z^{a_i}X^{b_i} = (-1)^{a_i.b_i} X^{b_i} Z^{a_i}$. This means that we want $a_i.b_i = 0$ for all possible i, so H_1 and H_2 must be orthogonal.

What about undetectable errors? Let's say we have some X^e, Z^f that commute with all the stabilizers. So why are they undetectable? We have $Z^{a_i}X^e = X^e Z^{a_i}$, which ensures that $Z^{a_i}X^e|\psi\rangle = X^e|\psi\rangle$. Thus, $X^e|\psi\rangle$ satisfies the definition of a codestate. As a result, these errors are undetectable. The size of these errors, or logicals, is also known as the **distance**.

One of the other theorems we previously discusses is $d-1$ local indistinguishability (LI). This is the idea that any two codestates $|\psi\rangle, |\phi\rangle$ are $d-1$ LI. This idea ties into the no-cloning theorem because you can't clone quantum information. Both $|\psi\rangle, |\phi\rangle$ encode quantum information, so if any of the d-1 qubits are lost, your code is able to correct those errors and the information is still there.

Now we can combine the ideas in this section to come up with the following theorem:

Theorem 25.1.1. *If a quantum error correcting code encodes at least one qubit (so there are at least 2 codestates), then any codestate $|\psi\rangle$ has a CC($|\psi\rangle$) $\geq \log(d-1)$.*

This means that quantum error correcting codes have a very high complexity! It requires a circuit of large depth to prepare the codewords.

25.1.5 Hard Hamiltonians

Let's consider the following:

$$H_{code} = \sum_i (I - Z^{a_i}) + \sum_i (I - X^{a_i})$$

What is the ground space? The ground space in this case is just the codespace. This is because both matrix terms in this summation are nonnegative, and the eigenvalues of these matrices are also at least 0. The eigenvalues are exactly 0 where the codestate conditions are satisfied, so the ground space is just the codespace.

Physicists like when these Hamiltonians are local, where the a_i's have a low hamming weight (e.g. $a_i = 01000\cdots$. Computer scientists like having a low hamming weight as well since this makes it easier to check if you are in the codespace. We can thus define:

Quantum LDPC (lower density parity check code): Codes in which stabilizers are local. It is also desirable to have large distance, such as $d \sim poly(n)$.

We might also be concerned about other aspects other than local stabilizers, etc, like geometry. For example, we can also say that $H = \sum_i P_{i,i+1}$. These are often not the best examples of hard hamiltonians. In some cases, we may end up with a gapped hamiltonian, where the ground state is an MPS with low entanglement.

25.1.6 Matrix Product States

Anurag's main takeaway: Matrix Product States are states with low entanglement

Bill's starting example takeaways: We have two registers A and B, where A has one qubit and B has 2 qubits. This means that some state $|\psi\rangle$ is an 8 dimensional vector. If we reshape this vector into a matrix, let's say a 2x4 matrix (which ends up being equivalent to $|\psi\rangle_A \langle\psi|_B$, both the vector and the matrix we created have the same resulting SVD and Schmidt decomposition. This decomposition ends up being something of the form $\sum_{i=1}^{r} \lambda_i u_i v_i^\dagger$. The rank is represented by r, and iff we have that r=1, then A and B are separable (the lower the rank, the more separable they are).

This is the most simple case. Now, if we have many qubits, how do we find the separability? Let's write our new

$$|\psi\rangle_{AB} = \sum_{j_1,j_2,\cdots j_n} C_{j_1,j_2,\cdots j_n} |j_1, j_2, \cdots j_n\rangle$$

. This is a vector of size 2^n. If we separate out j_1, we get

$$\sum_{j_1,<j_2,\cdots j_n>} C_{j_1,<j_2,\cdots j_n>} |j_1\rangle |j_2, \cdots j_n\rangle$$

This ends up being a matrix of size $2\text{x}2^{n-1}$. So we can separate states and end up with reshaped matrices, but we don't end up losing or gaining information. If we write out the SVD and look at specific elements, we observe that the maximum rank that we can achieve is 2, depending on the entanglement of the states. We can do this with each vector to unwrap the entanglement relationship between each pair.

[Bill's Notes:] Then continuing from the above separation result, where $C = \{C_{j_1,<j_2,\cdots j_n>}\}$ is a size $2 \times 2^{n-1}$ matrix, if we do a SVD on it, we get $C = USV^\dagger$, which in element-wise form, becomes:

$$C_{j_1,<j_2,\cdots j_n>} = \sum_{a_1} U_{j_1,a} S_{a_1,a_1} V^\dagger_{a_1,<j_2,\cdots j_n>}, \tag{25.1}$$

where index a_1 goes from 1 to the rank (at most two) of the matrix $C = \{C_{j_1,<j_2,\cdots j_n>}\}$ (you can already see that if C is rank 1, meaning that qubit 1 is separable/unentangled from qubits 2 to n, we are thus saving the "storage space"–the number of parameters need–by having a single term $a_1 = 1$ rather than summing over $a_1 = 1, 2$).

We substitute the expression for $C_{j_1,<j_2,\cdots j_n>}$ into the big sum:

$$\begin{aligned}|\Psi\rangle &= \sum_{j_1\cdots j_N} \underbrace{\sum_{a_1} U_{j_1,a_1} S_{a_1,a_1} V^{\dagger}_{a_1,\langle j_2\ldots j_N\rangle}}_{C_{j_1,<j_2,\cdots j_n>}} |j_1\rangle\,|j_2\ldots j_N\rangle \\ &= \sum_{j_1\cdots j_N}\sum_{a_1} U_{j_1,a_1} C^{(a_1)}_{\langle j_2\ldots j_N\rangle}\,|j_1\rangle\,|j_2\ldots j_N\rangle\end{aligned} \tag{25.2}$$

where we absorb the scalar $S_{a,a}$ into the matrix $\{V^{\dagger}_{a_1,<j_2,\cdots j_n>}\}$ and write the scalars product $S_{a_1,a_1}V^{\dagger}_{a_1,<j_2,\cdots j_n>}$ as $C^{(a_1)}_{\langle j_2\ldots j_N\rangle}$. **Do you notice the pattern?** We now have exactly another $\{C^{(a_1)}_{\langle j_2\ldots j_N\rangle}\}$ to decompose on, **given** a rank index a_1.

Now, try to make sense of this to yourself (we keep decomposing by SVD, first two lines replicated from above):

$$\begin{aligned}|\Psi\rangle &= \sum_{j_1\cdots j_N} \underbrace{\sum_{a_1} U_{j_1,a_1} S_{a_1,a_1} V^{\dagger}_{a_1,\langle j_2\ldots j_N\rangle}}_{C_{j_1,<j_2,\cdots j_n>}} |j_1\rangle\,|j_2\ldots j_N\rangle \\ &= \sum_{j_1\cdots j_N}\sum_{a_1} U_{j_1,a_1} C^{(a_1)}_{\langle j_2\ldots j_N\rangle}\,|j_1\rangle\,|j_2\ldots j_N\rangle \\ &= \sum_{j_1\cdots j_N}\sum_{a_1} U_{j_1,a_1}\sum_{a_2} U^{(a_1)}_{j_2,a_2} S^{(a_1)}_{a_2,a_2} {V^{(a_1)}_{a_2,\langle j_2\ldots j_N\rangle}}^{\dagger}\,|j_1\rangle\,|j_2\rangle\,|j_3\ldots j_N\rangle \\ &= \sum_{j_1\cdots j_N}\sum_{a_1} U^{(a_1)}_{j_1}\sum_{a_2} U^{(a_1,a_2)}_{j_2} C^{(a_1,a_2)}_{\langle j_2\ldots j_N\rangle}\,|j_1\rangle\,|j_2\rangle\,|j_3\ldots j_N\rangle \\ &= \sum_{j_1\cdots j_N}\sum_{a_1} U^{(a_1)}_{j_1}\sum_{a_2} U^{(a_1,a_2)}_{j_2}\sum_{a_3} U^{(a_1,a_2,a_3)}_{j_3} C^{(a_1,a_2,a_3)}_{\langle j_3\ldots j_N\rangle}\,|j_1\rangle\,|j_2\rangle\,|j_3\ldots j_N\rangle \\ &\ldots \\ &= \sum_{j_1\cdots j_N}\sum_{a_1} U^{(a_1)}_{j_1}\sum_{a_2} U^{(a_1,a_2)}_{j_2}\sum_{a_3} U^{(a_1,a_2,a_3)}_{j_3}\cdots\sum_{a_{N-1}} U^{(a_1,a_2,a_3,\ldots,a_{N-1})}_{j_{N-1}} C^{(a_1,a_2,a_3,\ldots,a_{N-1})}_{j_N}\,|j_1\rangle\,|j_2\rangle\ldots|j_{N-1}\rangle\,|j_N\rangle \\ &= \sum_{j_1\cdots j_N}\sum_{a_1a_2\ldots a_{N-1}} U^{(a_1)}_{j_1} U^{(a_1,a_2)}_{j_2} U^{(a_1,a_2,a_3)}_{j_3}\ldots U^{(a_1,a_2,a_3,\ldots,a_{N-1})}_{j_{N-1}} U^{(a_1,a_2,a_3,\ldots,a_{N-1})}_{j_N}\,|j_1\rangle\,|j_2\rangle\ldots|j_{N-1}\rangle\,|j_N\rangle\,,\end{aligned} \tag{25.3}$$

where in the last step, we write $C^{(a_1,a_2,a_3,\ldots,a_{N-1})}_{j_N}$ as $U^{(a_1,a_2,a_3,\ldots,a_{N-1})}_{j_N}$ for notational consistency.

This might not be what you would see in textbooks (for some reason many of the online sources are very lazy on the derivations). I hope I made the following clear: from the last step of the derivation, we can equivalently redefine the indexing by the following:

1. let (a_1, a_2) be the new "α_2" index
2. let (a_1, a_2, a_3) be the new "α_3" index
3. ...

4. let $(a_1, a_2, a_3, \ldots, a_{N-1})$ be the new "a_{N-1}" index.

Note, now $\alpha_1 \ldots \alpha_{N-1}$ indices are no longer independent indices! α_k depends on the choice of α_{k-1}.

Then we have:

$$\begin{aligned}|\Psi\rangle &= \sum_{j_1 \ldots j_N} \sum_{a_1 a_2 \ldots a_{N-1}} U_{j_1}^{(a_1)} U_{j_2}^{(a_1,a_2)} U_{j_3}^{(a_1,a_2,a_3)} \ldots U_{j_{N-1}}^{(a_1,a_2,a_3,\ldots,a_{N-1})} U_{j_N}^{(a_1,a_2,a_3,\ldots,a_{N-1})} |j_1\rangle |j_2\rangle \ldots |j_{N-1}\rangle |j_N\rangle \\ &= \sum_{j_1 \ldots j_N} \sum_{a_1 a_2 \ldots a_{N-1}} U_{j_1}^{(\alpha_1)} U_{j_2}^{(\alpha_2)} U_{j_3}^{(\alpha_3)} \ldots U_{j_{N-1}}^{(\alpha_{N-1})} U_{j_N}^{(\alpha_{N-1})} |j_1\rangle |j_2\rangle \ldots |j_{N-1}\rangle |j_N\rangle .\end{aligned} \tag{25.4}$$

Exactly as what I said before, α_k depends on the choice of α_{k-1}, so in the summation, we are still effectively summing over $a_1 a_2 \ldots a_{N-1}$ in stead of $\alpha_1 \alpha_2 \ldots \alpha_{N-1}$ (this might sound a bit confusing). How do we break this dependency? We add in the so-called pair-wise interaction!

$$\begin{aligned}|\Psi\rangle &= \sum_{j_1 \ldots j_N} \sum_{\alpha_1 \alpha_2 \ldots \alpha_{N-1}} U_{j_1}^{(\alpha_1)} U_{j_2}^{(\alpha_1,\alpha_2)} U_{j_3}^{(\alpha_2,\alpha_3)} \ldots U_{j_{N-1}}^{(\alpha_{N-2},\alpha_{N-1})} U_{j_N}^{(\alpha_{N-1})} |j_1\rangle |j_2\rangle \ldots |j_{N-1}\rangle |j_N\rangle \\ &= \sum_{j_1 \ldots j_N} \sum_{a_1 a_2 \ldots a_{N-1}} U_{j_1}^{(a_1)} U_{j_2}^{(a_1, \underbrace{a_1, a_2}_{\alpha_2})} U_{j_3}^{(\underbrace{a_1, a_2}_{\alpha_2}, \underbrace{a_1, a_2, a_3}_{\alpha_3})} \ldots U_{j_{N-1}}^{(\alpha_{N-2},\alpha_{N-1})} U_{j_N}^{(\alpha_{N-1})} |j_1\rangle |j_2\rangle \ldots |j_{N-1}\rangle |j_N\rangle .\end{aligned} \tag{25.5}$$

Do you see what is happening here!? By adding some redundant indices, we are breaking the long dependency chain $a_1, (a_1, a_2), (a_1, a_2, a_3), \ldots, (a_1, a_2, \ldots, a_{N-1})$ into a nearest-neighbor dependency chain $\underbrace{a_1}_{\alpha_1}, (\underbrace{a_1}_{\alpha_1}, \underbrace{a_1, a_2}_{\alpha_2}), (\underbrace{a_1, a_2}_{\alpha_2}, \underbrace{a_1, a_2, a_3}_{\alpha_3}), \ldots, (\underbrace{a_1, a_2, \ldots, a_{N-2}}_{\alpha_{N-2}}, \underbrace{a_1, a_2, \ldots, a_{N-2}, a_{N-1}}_{\alpha_{N-1}})$, at the cost of having very sparse matrices.

Why sparse? Note that within each U matrix, for example, $U_{j_2}^{(a_1,a_1,a_2)} = U_{j_2}^{(a_1,\alpha_2)}$, we must require that the two a_1's must agree, otherwise we set the element to zero:

$$U_{j_2}^{(\underbrace{a_1', a_1}_{\text{these have to agree}}, a_2)} = 0 \text{ if } a_1' \neq a_1 \tag{25.6}$$

We end up observing that the amplitudes C can be estimated using a product of matrices. This idea of using the product of matrices to determine the amplitude of basis vectors is what comprises Matrix Product States. The more entangled a state is, the larger these states are and the harder they are to store. As a result, we prefer these states to have low entanglement.

25.1.7 Back to code Hamiltonians

Let's go back to our definition of H_{code}.

$$H = \sum_i (I - Z^{a_i}) + \sum_i (I - X^{a_i})$$

$a_1, a_2, \cdots a_{n-k}$ are the rows of H_1 and $b_1, b_2, \cdots b_l$ are the rows of H_2. These are all rows of low hamming weight. So what can be an example of a code? We looked at repetition codes previously,

which looked like

$$H_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

With this definition of H_1, our code words are

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

What about H_2? Can that just be another repetition code? We do have one constraint that $a_i.b_i = 0$ because X^{a_i} and Z^{b_i} have to commute. This means that H_1 and H_2 must be orthogonal.

Which b is orthogonal to all the codes in H_1? $b = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$. This is also a codeword of H_1. This means that b has to be a code word of H_1 in order for it to hold that $a_i.b_i = 0$. If H_1 is a good error correcting code, its code words will be very big with a large hamming weight, which means we will have small a_i's and large b_i's, which seems contradictory. This means that the construcion of a good quantum error correcting code is magic.

25.1.8 Toric Code

When we were discussing repetition code previously, we were discussing stabilizers that looked like the following

$$Z_1 \otimes Z_2, Z_2 \otimes Z_3, Z_3 \otimes Z_4$$

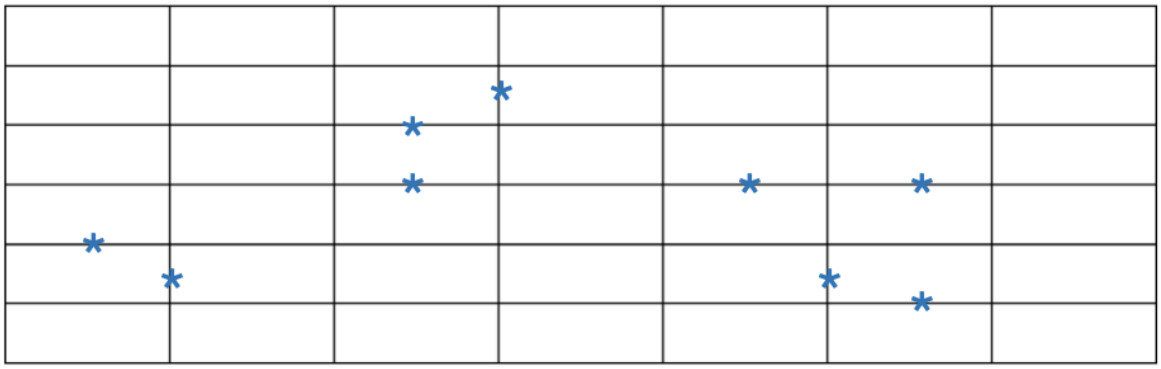
While this was a good classical code, it wasn't able to detect all quantum errors, so we had to introduce

$$X_1 \otimes X_2 \otimes X_3 \cdots X_b$$

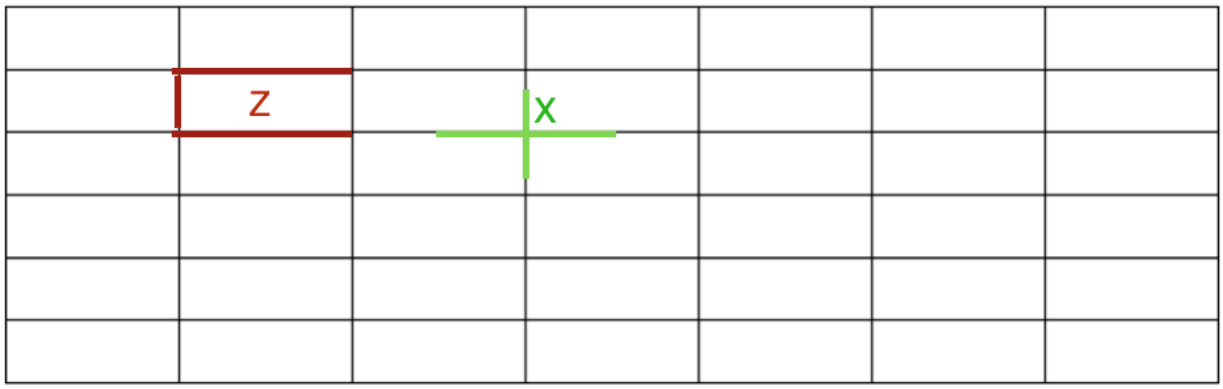
The Z's were able to detect the X errors, but we needed a very long X to detect all the Z errors.

So how do you find a local hamiltonian where both a_i and b_i have a low hamming weight? The answer to this comes from Toric Code. Toric Code combines these classical codes while avoiding a high weight b_i.

How it works: Think of a torus, and place qubits on all the edges, so it would look like the following:

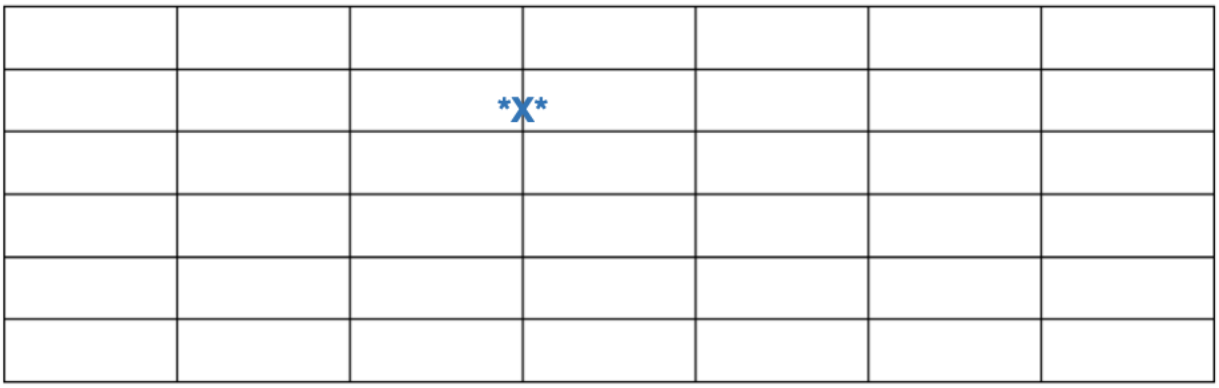


Now let's define two checks. X-checks will be checks sitting on the vertices and Z-checks will be checks sitting on the faces. This looks like the following

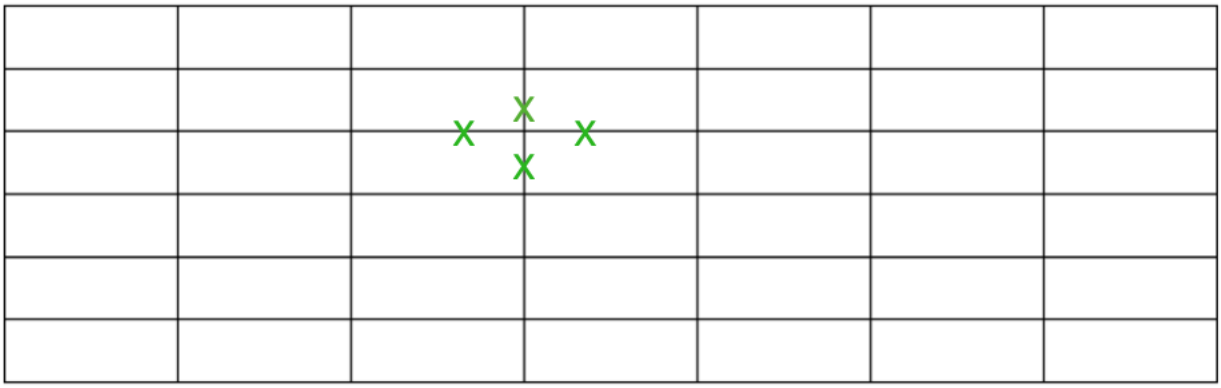


This ends up making a 4-local hamiltonian because each matrix term in H has a rank 4. X and Z also all commute.

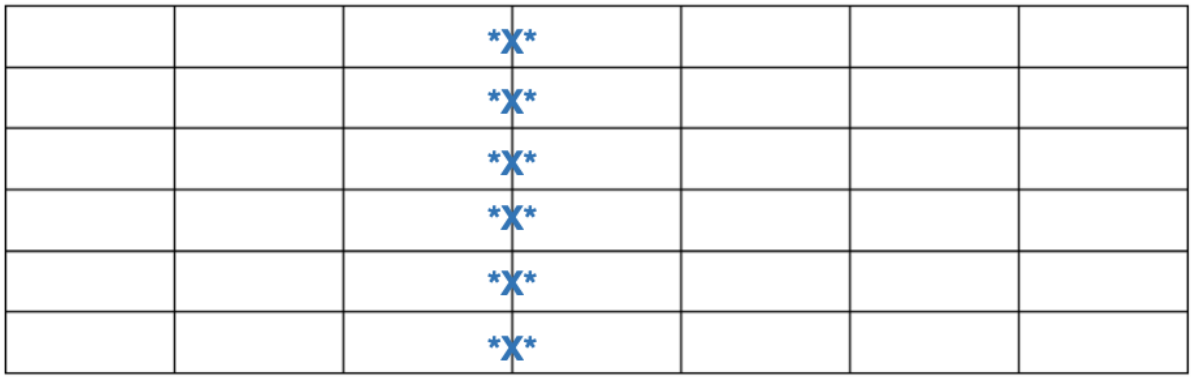
How does this detect errors? If we have an X error that looks like the following



The error will anticommute with the Z's on the adjacent face so you can still easily detect it. How about an error like the following?



While these errors commute still, we aren't as concerned because this actually still a stabilizer. However, we would be concerned about the error below



We can group some groups of 2 together to close loops on the torus, but the rest cannot be closed and will be undetectable. Overall, the toric code distance is $\log(\sqrt{n})$, so the circuit complexity lower bound is around $\log(\sqrt{n})$.

Bibliography

[AHL+14] Dorit Aharonov, Aram W. Harrow, Zeph Landau, Daniel Nagaj, Mario Szegedy, and Umesh Vazirani. Local tests of global entanglement and a counterexample to the generalized area law. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 246–255, 2014.

[AN02] Dorit Aharonov and Tomer Naveh. Quantum np-a survey. *arXiv preprint quant-ph/0210077*, 2002.

[BB84] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *International Conference on Computers, Systems, and Signal Processing*, 1:175–179, 1984.

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, jul 2001.

[BG20] Anne Broadbent and Alex B. Grilo. Qma-hardness of consistency of local density matrices with applications to quantum zero-knowledge. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 196–205, 2020.

[Eke91] Artur K. Ekert. Quantum cryptography based on bell's theorem. *Phys. Rev. Lett.*, 67:661–663, Aug 1991.

[GH10] Daniel Gottesman and M B Hastings. Entanglement versus gap for one-dimensional spin systems. *New Journal of Physics*, 12(2):025002, feb 2010.

[Ira10] Sandy Irani. Ground state entanglement in one-dimensional translationally invariant quantum systems. *Journal of Mathematical Physics*, 51(2):022101, 2010.

[LYC16] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang. Methodology of Resonant Equiangular Composite Quantum Gates. *Physical Review X*, 6(4):041067, December 2016.

[Mah18] Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018.

[SV14] Sushant Sachdeva and Nisheeth K. Vishnoi. *Faster Algorithms via Approximation Theory. Foundations and Trends® in Theoretical Computer Science*, 9(2):125–210, 2014.

[Wat06] John Watrous. Zero-knowledge against quantum attacks. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, page 296–305, New York, NY, USA, 2006. Association for Computing Machinery.

[YK17] Beni Yoshida and Alexei Kitaev. Efficient decoding for the hayden-preskill protocol, 2017.