MIT 8.370/18.435/2.111: Quantum Computation

Kan, Benji
benji_kan@college.harvard.edu

Kaxiras, Vassilios
vkaxiras@college.harvard.edu

Li, Austin
awli@college.harvard.edu

Fall 2021

Abstract

These are notes[1] for MIT's *8.370*, a graduate class on quantum computation, as taught by Professor Peter Shor in Fall 2021. We will cover most of the textbook *Quantum Computation and Quantum Information* by Nielsen and Chuang.

Course description: Provides an introduction to the theory and practice of quantum computation. Topics covered: physics of information processing; quantum algorithms including the factoring algorithm and Grover's search algorithm; quantum error correction; quantum communication and cryptography. Knowledge of quantum mechanics helpful but not required.

Contents

[1] With thanks to Eric K. Zhang for the template.

1 September 8th, 2021

1.1 Class organization

Grading is 50% HW, 20% Midterm (on some Wednesday in the middle of the term), and 30% Final. $>90\%$ is an A, $>80\%$ is a B, although these cutoffs can be lowered.

Weekly homeworks will be due Fridays at 8pm on Gradescope, and will be posted on either Friday or Saturday on Canvas. Homeworks may be submitted up to 24 hours late with a 10% penalty, after that solutions will be posted and we will need a note from S^3 (or Harvard equivalent). Collaboration is strongly encouraged, and a Piazza will be set up.

Lectures will be automatically recorded and posted to Panopto. The textbook is Nielsen and Chuang, and John Preskill's (Caltech) lecture notes will be helpful. Prof. Shor will post reading sections on Canvas ahead of lectures.

1.2 History of quantum information (Nielsen 1.1)

Quantum Mechanics was developed in 1925 by Bohr, Heisenberg, Schrodinger, Born. In 1935, the EPR triplet found the EPR paradox, where two entangled particles appear to have transmitted both position and momentum information faster than light. In the 1960s, Bell proposed that QM is either 1. non-relativistic or 2. non-local by using discretized EPR.

There are various "interpretations" of QM, and we will use the Copenhagen interpretation: Merman interprets this as "shut up and calculate".

What can we do with this weirdness? In 1967, Wiesman wrote a paper about uncounterfeitable quantum money, and in 1983, Bennett used the idea in quantum key distribution, where two people can use a quantum channel to agree on a secret key without a prior secret key.

In 1981, Feynman noticed that it seemed exponentially hard to simulate quantum systems on classical computers, and suggested using a *quantum computer*. In 1985, Deutsch asked if a quantum computer can solve problems faster than classical computers. Simon proposed a problem which quantum computers can solve faster, and Shor came up with Shor's factoring algorithm.

Chuang will cover quantum communication channels in 8.371 in the spring.

2 September 10th, 2021

2.1 Single qubits

Theorem 2.1 (Superposition principle)**.** *If $|A\rangle$ and $|B\rangle$ are two distinguishable states of a quantum system, then $\alpha|A\rangle + \beta|B\rangle$ is also a state if $|\alpha|^2 + |\beta|^2 = 1$.*

Mathematically, this means that a quantum system is a complex vector space where $|A\rangle$ and $|B\rangle$ are unit column vectors. Moreover, distinguishable states are *orthogonal* vectors. So, inputting $\alpha|A\rangle + \beta|B\rangle$ into an experiment that distinguishes the states $|A\rangle$ and $|B\rangle$ will output $|A\rangle$ with probability $|\alpha|^2$ and $|B\rangle$ with probability $|\beta|^2$.

Additionally, applying a unit (complex) phase to a state $|q\rangle$ gives the same state $|q\rangle$, that is, there is no way to distinguish between $|q\rangle$ and $e^{i\theta}|q\rangle$. (However, a rotation of 360° induces a phase change of -1 and it may interfere with itself).

Definition 2.2 (Inner product)**.** We use the inner product $\langle v|w\rangle = v^\dagger w$.

Thus, if $|\alpha|^2 + |\beta|^2 = 1$, we have $|v\rangle = \begin{bmatrix}\alpha \\ \beta\end{bmatrix}$, $\langle v| = \begin{bmatrix}\alpha & \beta\end{bmatrix}$.

Example 2.3 (Polarizing filter)**.** Let a horizontal polarizing filter be one that allows $\leftrightarrow$ horizontal photons to go through, so $\updownarrow$ vertical photons do not.

Then, no light goes through if we use both a horizontal and vertical polarizing filter.

Definition 2.4 (Qubit)**.** A **qubit** is a 2D quantum vector space with only two distinguishable states. Everything else is a linear combination of those states.

Example 2.5. $|\updownarrow\rangle$ is vertical, $|\leftrightarrow\rangle$ is horizontal. Then, we also have diagonal photon states $|⤢\rangle = \frac{1}{\sqrt{2}}(|\leftrightarrow\rangle + |\updownarrow\rangle)$ and $|⤡\rangle = \frac{1}{\sqrt{2}}(|\leftrightarrow\rangle - |\updownarrow\rangle)$.

When we put a diagonal photon through a vertical polarizer, we get $|\updownarrow\rangle$ with probability $(\frac{1}{\sqrt{2}})^2 = \frac{1}{2}$, and $|\leftrightarrow\rangle$ also with probability $(\frac{1}{\sqrt{2}})^2 = \frac{1}{2}$.

Recall that quantum mechanics occurs in a complex vector space. As a result, we define

Definition 2.6 (Right and left circularly polarizations)**.**

$$\frac{1}{\sqrt{2}}(|\leftrightarrow\rangle + i|\updownarrow\rangle) = |\circlearrowleft\rangle, \qquad \frac{1}{\sqrt{2}}(|\leftrightarrow\rangle - i|\updownarrow\rangle) = |\circlearrowright\rangle \tag{1}$$

Definition 2.7. Principle: QM is linear*???*

Theorem 2.8. *An isolated quantum state evolves according to a **unitary matrix**. Additionally, an isolated quantum state evolves linearly.*

$$|\psi\rangle \xrightarrow{\text{time evolution}} M|\psi\rangle \tag{2}$$

We want evolution to preserve lengths. That is, we want M to preserve lengths: unit vectors are taken to unit vectors. Thus,

$$\langle\psi|M^\dagger M|\psi\rangle = 1 \tag{3}$$

So,

$$\langle\psi|M^\dagger M|\psi\rangle = 1 \ \forall \ \text{ unit } |\psi\rangle \implies M^\dagger M = I \tag{4}$$

Recall $\langle\psi| \equiv \psi^\dagger$.

Note. M rotates vectors in complex space.

Definition 2.9 (Polarization states)**.** Each of these states is one instance of a qubit:

$$|\leftrightarrow\rangle \equiv |0\rangle, \qquad |\updownarrow\rangle \equiv |1\rangle, \qquad |⤢\rangle \equiv |+\rangle, \qquad |⤡\rangle \equiv |-\rangle, \tag{5}$$

$$|\circlearrowleft\rangle \equiv \frac{1}{\sqrt{2}}(|\leftrightarrow\rangle + i|\updownarrow\rangle)(\equiv |+i\rangle), \qquad |\circlearrowright\rangle \equiv \frac{1}{\sqrt{2}}(|\leftrightarrow\rangle - i|\updownarrow\rangle)(\equiv |-i\rangle) \tag{6}$$

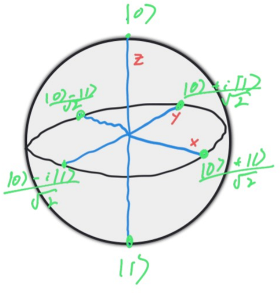
2.2 The Bloch sphere

Example 2.10 (Stern-Gerlach experiment)**.** The setup for the Stern-Gerlach experiment begins with a spin $\frac{1}{2}$ particle, with two basis states: $|\uparrow\rangle$ and $|\downarrow\rangle$. The apparatus contains a varying magnetic field such that it separates $|\uparrow\rangle$s up from $|\downarrow\rangle$s down.

We also define mixed states

$$|\rightarrow\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle), \qquad |\leftarrow\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle - |\downarrow\rangle), \qquad |\cdot\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle + i|\downarrow\rangle), \qquad |\times\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle - i|\downarrow\rangle) \tag{7}$$

Definition 2.11 (Bloch sphere)**.** We define the Bloch Sphere as “the geometrical representation of the pure state space of a two-level quantum mechanical system."



Bloch Sphere

Definition 2.12 (Unitary rotation matrices)**.** A matrix is **unitary** if $U^\dagger U = UU^\dagger = I$. Equivalently, both the columns and rows are unit vectors. We define the following useful rotation matrices:

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{8}$$

Note. A silly mnemonic for remembering σ_y: the $-i$ is lighter and floats up.

On the Bloch sphere, σ_x rotates vectors 180° about the x axis:

$$\sigma_x|\uparrow\rangle = |\downarrow\rangle, \qquad \sigma_x|\downarrow\rangle = |\uparrow\rangle, \qquad \sigma_x|\rightarrow\rangle = |\rightarrow\rangle, \qquad \sigma_x|\leftarrow\rangle = -|\leftarrow\rangle = |\leftarrow\rangle \tag{9}$$

because $|\leftarrow\rangle, -|\leftarrow\rangle$ differ by a unit phase. Note that σ_y and σ_z act analogously.

3 September 13th, 2021

Announcements

- OH W2:30-4
- Problem set posted on Friday
- Guest lecturers next week: Ike Chuang and Aram Harrow

Recall: A quantum state is a unit vector in $\mathbb{C}^k$, and isolated quantum systems are evolved by unitary matrices. A matrix U is unitary if and only if $UU^\dagger = U^\dagger U = I$. Fact: unitary states are diagonalizable, with $U = V^* D V$, with V is unitary and D a diagonal matrix with unit complex numbers along the diagonal ($e^{i\theta}$ for real θ).

3.1 Measurement

Definition 3.1 (Quantum measurement)**.** A complete von Neumann measurement is determined by an orthonormal basis of $\mathbb{C}^k$.

For example, $|0\rangle$ and $|1\rangle$ could be a qubit. If a quantum state is $\alpha|0\rangle + \beta|1\rangle$, then we see $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. But we can also take the basis

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{10}$$

Example 3.2. Measure state $|\psi\rangle = .8|0\rangle + .6|1\rangle$ in the $\{|+\rangle, |-\rangle\}$ basis.

$$P(|+\rangle) = |\langle + | \psi\rangle|^2 = \left| \frac{1}{\sqrt{2}}(\langle 0| + \langle 1|)(.8|0\rangle + .6|1\rangle)\right|^2 = \left|\frac{14}{\sqrt{2}}\right|^2 = .98 \tag{11}$$

$$P(|-\rangle) = |\langle - | \psi\rangle|^2 = .02 \tag{12}$$

Intuitively, the probability is a projection of the desired state onto the basis.

We had a basis $|0\rangle\langle 0|, |1\rangle\langle 1|$ (1-dimensional projection). So,

$$|0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{13}$$

Thus, $|0\rangle\langle 0| + |1\rangle\langle 1| = I$. Suppose we have a basis of four states $\{|0\rangle, |1\rangle, |2\rangle, |3\rangle\}$

We claim there is a measurement that answers the question: in the state either $\{|0\rangle, |1\rangle\}$ or $\{|2\rangle, |3\rangle\}$? In the Copenhagen interpretation, measurement “collapses" into the measured state.

Definition 3.3 (Von Neumann measurement)**.** A **von Neumann measurement** corresponds to a complete set of projection matrices. A projection matrix has eigenvalues 0 and 1, where complete means

$$\Pi_i \Pi_j = 0, i \neq j, \qquad \sum_{i=1}^{k} \Pi_i = I. \tag{14}$$

Π_i always looks like $|v_1\rangle\langle v_1| + \cdots + |v_\ell\rangle\langle v_\ell|$ where the v_i's are some set of orthonormal $|v_j\rangle$.

Example 3.4.

$$\Pi_A = |0\rangle\langle 0| + |1\rangle\langle 1| = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0 & \\ & & & 0 \end{bmatrix}, \qquad \Pi_B = |2\rangle\langle 2| + |3\rangle\langle 3| = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \tag{15}$$

Here, $\Pi_1\Pi_2 = 0, \Pi_1 + \Pi_2 = I$, so it is complete.

For example: $|\psi\rangle = .5|0\rangle + .7|1\rangle + .5|2\rangle + .1|3\rangle$. When we measure it, we get $\Pi_1|\psi\rangle$ with probability $|\Pi_1|\psi\rangle|^2 = \langle\psi|\Pi_1|\psi\rangle$ and $\Pi_2|\psi\rangle$ with probability $|\Pi_2|\psi\rangle|^2 = \langle\psi|\Pi_2|\psi\rangle$. Remember to normalize the original vectors.

We have

$$\Pi_1|\psi\rangle = .5|0\rangle + .7|1\rangle \tag{16}$$

so we get $\Pi_1|\psi\rangle$ with probability $.25 + .49 = .74$. We normalize as $\frac{.5|0\rangle+.7|1\rangle}{\sqrt{.74}}$. Similarly, we have

$$\Pi_2|\psi\rangle = .5|2\rangle + .1|3\rangle \tag{17}$$

with probability .26.

3.2 Circuit Model of Quantum Computation

We start with the state $|\psi\rangle = |v_1\rangle|v_2\rangle\ldots|v_n\rangle$ where each qubit $|v_i\rangle$ is a 2D quantum state, so the overall n qubits are a 2^n dimensional quantum state. Now, we apply quantum gates (unitary matrices) to small sets of qubits to make qubit measurements. Due to interference, we measure everything at the end.

Example 3.5 (Interference)**.** Interference happens in quantum computation. Consider the unitary matrix $M = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1\\1 & 1\end{bmatrix}$. Apply it to $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ to get $\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} = |+\rangle$. Apply it twice to get $\begin{bmatrix}0\\1\end{bmatrix} = |1\rangle$.

$M\begin{bmatrix}1\\0\end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$: measure it, and get $|0\rangle$ with probability $\frac{1}{2}$ and $|1\rangle$ with probability $\frac{1}{2}$ in the $\{|0\rangle, |1\rangle\}$ basis. Then, $M^2\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix}$, since $M^2 = \begin{bmatrix}0 & -1\\1 & 0\end{bmatrix} = -i\sigma_y$.

Example 3.6 (Interferometer)**.** If light is split into two paths and recombined, and we want to measure which path it was taken. Each of the detectors at the end see 50% of the photons. Note that M tells us what happens along a beam splitter, as a superposition of each of the two paths

4 September 15th, 2021

4.1 Eitzur-Vaidman bomb test

Consider an interferometer that splits incoming light into $|h\rangle$ and $|v\rangle$ with 50% probability each. Thus, the matrix for this beamsplitter is

$$M = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \qquad \text{single photons} \tag{18}$$

Note that this is the asymmetric version, although the symmetric version is equivalent (under unit phases).

Example 4.1. Start with $|h\rangle$. Then,

$$M|h\rangle = \frac{1}{\sqrt{2}}(|h\rangle + |v\rangle) \implies M\frac{1}{\sqrt{2}}(|h\rangle + |v\rangle) = \frac{1}{2}(|h\rangle + |v\rangle - |h\rangle + |v\rangle) = |v\rangle \tag{19}$$

This means that after splitting and recombining in the interferomter experiment, an incoming horizontal photon always becomes a vertical photon and triggers the vertical photodetector.

Example 4.2 (Eitzur-Vaidman bomb test)**.** Factory produces bombs, where bomb will explode if a photon hits the detector. But some bombs are defective (photodetector is broken)! How can we find the non-defective bombs without exploding it?

Classically, no way to find out. How do we do this quantum mechanically?

Proof. We put the bomb in the middle of an interferometry experiment.

Suppose the bomb is defective. Then, the photon will travel along both paths and (per above) the vertical detector will detect.

Suppose the bomb is now not defective. Then, the bomb will measure the photon, and with probability $\frac{1}{2}$ it will be in $|v\rangle$ and go off. With probability $\frac{1}{2}$ it will not go off, so it will be in $|h\rangle$ and get to the beam splitter vertically. Applying $M|v\rangle = -|h\rangle + |v\rangle$ (unnormalized), so we'll see it in the vertical detector 50% of the time and the horizontal detector 50% of the time. Thus, if not defective, the horizontal detector detects $\frac{1}{4}$ of the time.

Comparing, we see that horizontal detector detecting means bomb not defective, and it did not explode! □

Note. There is a way of doing this where the probability the bomb explodes is 1 in 10 million, or arbitrarily small (give more complex experiments), without decreasing the probability that you detect it to be not defective and not exploding. So! Not a trade-off.

4.2 Measurement

Definition 4.3 (Observable and Hermitian)**.** An **observable** on a quantum state space is a Hermitian matrix. A **Hermitian** matrix is the analog of a symmetric matrix. That is, M is Hermitian if $M = M^\dagger$. A Hermitian matrix has real eigenvalues and is diagonalizable: $M = V^\dagger D V$, where D is diagonal and real.

So if M is observable, then we can write

$$M = \sum_i \lambda_i |v_i\rangle\langle v_i| \tag{20}$$

where $|v_i\rangle$ are the eigenvectors and λ_i are real.

So how does an observable correspond to a von Neumann measurement? An observable measures some physical quantity, like spin. An observable corresponds to subspaces for each eigenvalues, so

$$\Pi_{\lambda_i} \rightarrow \text{ projection onto subspace of eigenvectors with eigenvalue } \lambda_i$$

the corresponding measured eigenvalue.

Example 4.4 (Spin $\frac{1}{2}$ matrices)**.** An observable with spin in the z direction is $\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} \end{bmatrix} = \frac{1}{2}\sigma_z$. So, we measure $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ with eigenvalue $\frac{1}{2}, |\uparrow\rangle$, and we measure $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ with eigenvalue $-\frac{1}{2}, |\downarrow\rangle$ (eigenvalues measured in the z direction).

Then, the observable for spin along the x direction is

$$\frac{1}{4}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix} - \frac{1}{4}\begin{bmatrix} 1 \\ -1 \end{bmatrix}\begin{bmatrix} -1 & 1 \end{bmatrix} = \frac{1}{4}\left(\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\right) = \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix} = \frac{1}{2}\sigma_x \tag{21}$$

Theorem 4.5. *Given an observable M and a quantum state $|\psi\rangle$, the expected value of the observable is $\langle\psi|M|\psi\rangle$.*

Example 4.6. Suppose $|\psi\rangle = \begin{bmatrix} \frac{3}{5} \\ \frac{4}{5} \end{bmatrix} = \frac{3}{5}|0\rangle + \frac{4}{5}|1\rangle$, and $M = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} = 3\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} + 1\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$. The first matrix projects onto $|+\rangle$ and the second onto $|-\rangle$.

Then,

$$\langle\psi|M|\psi\rangle = \begin{bmatrix} \frac{3}{5} & \frac{4}{5} \end{bmatrix}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} \frac{3}{5} \\ \frac{4}{5} \end{bmatrix} = \frac{74}{25} \tag{22}$$

Note that if we measure $|\psi\rangle$ in the $|+\rangle, |-\rangle$ basis, we have

$$P(+) = \left(\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}\begin{bmatrix} \frac{3}{5} \\ \frac{4}{5} \end{bmatrix}\right)^2 = \frac{49}{50}, \qquad P(-) = \left(\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}\begin{bmatrix} \frac{3}{5} \\ \frac{4}{5} \end{bmatrix}\right)^2 = \frac{1}{50} \tag{23}$$

So the expectation value is $3\frac{49}{50} + 1\frac{1}{50} = \frac{74}{25}$, as expected.

Proof. We have

$$\mathbf{E}\left[outcome\right] = \sum_i \lambda_i \Pr outcome = \sum_i \lambda_i |\langle v_i|\psi\rangle|^2 \tag{24}$$

Note. Note if Π_{λ_i} is more than rank one, then we can write the matrix out as a sum of projections onto eigenvectors:

$$\Pi_{\lambda_i} = |v_{\lambda_i}^{(1)}\rangle\langle v_{\lambda_i}^{(1)}| + |v_{\lambda_i}^{(2)}\rangle\langle v_{\lambda_i}^{(2)}| \tag{25}$$

Recall that Π_{λ_i} is the subspace of eigenvectors with eigenvalue λ_i.

We continue writing

$$\sum_i \lambda_i |\langle v_i|\psi\rangle|^2 = \sum_i \lambda_i \langle\psi|v_i\rangle\langle v_i|\psi\rangle = \langle\psi|\left(\sum_i \lambda_i |v_i\rangle\langle v_i|\right)|\psi\rangle = \langle\psi|M|\psi\rangle \tag{26}$$

□

5 September 17th, 2021

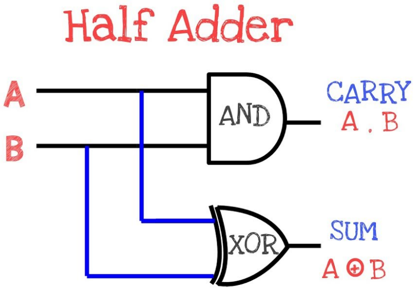
Announcements

- Guest lecturers Aram Harrow and Ike Chuang next Monday and Wednesday

5.1 Classical circuits

Classical circuits map binary inputs to binary outputs with Boolean functions. They can also be represented as truth tables, such as the exclusive-or and the and.

We consider the circuit for the **half-adder**: it has two FANOUT gates ($x \to x, x$), a SWAP gate ($x, y \to y, x$), an XOR and an AND gate, each with an output.



Theorem 5.1. *Any binary function can be computed using 2-bit gates of (AND, NOT).*

N.B. also 1-bit gate: NOT.

Proof. Suppose we have $f(x_1, x_2, \dots, x_n)$. We proceed by induction. Assume we have

$$f_0 = f(0, x_2, x_2, \dots x_n), \qquad f_1 = f(1, x_2, x_2, \dots x_n) \tag{27}$$

Then, we can compute

$$f = (x_0 \wedge f_1) \vee (\neg x_1 \wedge f_0) \tag{28}$$

□

How many gates are in this construction? We have 2 AND and 1 OR gate, so

$$gates(n) = 3 + 2 \cdot gates(n-1) \implies gates(n) \sim O(2^n) \tag{29}$$

Is this necessary? We can count the number of Boolean functions: 2^{2^n} (two outputs for each of 2^n possible inputs of length n), and the number of circuits with k gates has an upper bound of $2^{k \cdot O(k)}$.

Thus, you really do need $k \approx 2^n$ gates to make a Boolean function.

5.1.1 Quantum circuit diagram

A qubit is represented by a wire in a quantum circuit diagram.

Claim. In an isolated system, QM is reversible due to measurements.

How about reversible classical computation? What 2-bit gates are reversible?

Example 5.2. NOT is reversible. AND is not reversible. Note that for it to be reversible, there must be the same number of input and output gates.

$x, y \to x \wedge y, x \vee y$ is also not reversible, due to repeat.

$$\textbf{CNOT}: x, y \to \begin{cases} x, y \text{ if } x = 0 \\ x, \neg y \text{ if } x = 1 \end{cases}$$

x is the control bit, y is the target bit.

5.2 Universal reversible computation

Can you do universal reversible computation with 2-bit reversible gates? **No.**

Theorem 5.3. *Any output made from CNOT, NOT is of the form:*

$$x_1 \oplus x_2 \oplus \dots x_k$$

$$\neg x_1 \oplus x_2 \oplus \dots x_k$$

Proof. By induction. **Base case**: Check 2-bit reversible gates (NOT, SWAP, CNOT). Assume induction hypothesis. Then, repeated elements in the XOR work out. □

Can you do universal reversible computation with 3-bit reversible gates? **Yes.**

Definition 5.4 (Toffoli gate)**.** The Toffoli gate is a CCNOT, where the first two bits are both control bits, and the last bit is the target bit.

$$\text{Toff}(x, y, z) = x, y, z \oplus (x \wedge y) \tag{30}$$

000	000
001	001
010	010
011	011
100	100
101	101
110	111
111	110

Note the last two rows have the target bit flipped.

Why do Toffoli gates let us perform universal reversible computation?

$$\text{Toff}(x, y, 0) = (x, y, x \wedge y), \qquad \text{Toff}(x, 1, 1) = (x, 1, x \oplus 1) = (x, 1, \neg x) \tag{31}$$

Can do any computation $(\text{input}, 0^a, 1^b) \to (\text{input}, \text{output}, \text{intermediate results})$. By keeping input, all computations are reversible.

6 September 20th, 2021

Announcements

- Today is a guest lecture given by Aram Harrow.

6.1 Multiple quantum systems

We begin with the case of **two qubits**. They can be in a superposition of $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Thus,

$$|\psi\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle = \begin{bmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{bmatrix} \in \mathbb{C}^4 \tag{32}$$

When we generalize to n **qubits**, then there are 2^n states, which is a superposition of all such states. Thus, $|\psi\rangle \in \mathbb{C}^{2^n}$. This is a central attraction of quantum computing, as there is exponential growth.

$$|\psi\rangle = \sum_{x \in [0,1]^n} c_x |x\rangle, \qquad |x\rangle = |x_1, x_2, \ldots, x_n\rangle \tag{33}$$

More generally, we consider if system 1 has state space $V = \mathbb{C}^{d_1}$ and system 2 has state space $W = \mathbb{C}^{d_2}$. Then, the combined system has state space $\mathbb{C}^{d_1 d_2}$ in the superpositions.

Definition 6.1 (Tensor products)**.** We define **tensor products** of two state spaces as

$$V \otimes W = \operatorname{span}\{|v\rangle \otimes |w\rangle, |v\rangle \in V, |w\rangle \in W\} \tag{34}$$

For example, $|01\rangle = |0\rangle \otimes |1\rangle$.

1.

$$|v\rangle = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{d_1} \end{bmatrix}, \qquad |w\rangle = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{d_2} \end{bmatrix}, \qquad |v\rangle \otimes |w\rangle = \begin{bmatrix} v_1 w_1 \\ \cdots \\ v_1 w_{d_2} \\ v_2 w_1 \\ \cdots \\ v_{d_1} w_{d_2} \end{bmatrix} \tag{35}$$

   The tensor products of the basis vectors for the subspaces become the basis vectors for the larger space.

2. Moreover, the tensor product $|v\rangle, |w\rangle \to |v\rangle \otimes |w\rangle$ is **bilinear**, that is, it is linear in both arguments

$$|v\rangle \otimes (c_1|w_1\rangle + c_2|w_2\rangle) = c_1|v\rangle \otimes |w_1\rangle + c_2|v\rangle \otimes |w_2\rangle \tag{36}$$

   and similarly in the other argument.

Note (Probability)**.** We have seen $\otimes$ before in probability theory, when combining two probabilities:

$$p = \begin{bmatrix} p_0 \\ p_1 \end{bmatrix}, \qquad q = \begin{bmatrix} q_0 \\ q_1 \end{bmatrix}, \qquad p \otimes q = \begin{bmatrix} p_0 q_0 \\ p_0 q_1 \\ p_1 q_0 \\ p_1 q_1 \end{bmatrix} \tag{37}$$

(all ways of combining them)

For quantum states, the joint system of $|\alpha\rangle, |\beta\rangle$ is $|\alpha\rangle \otimes |\beta\rangle$.

Note (Outer products)**.** Not only is it similar to inner products, but it is also similar to outer products.

$$|x\rangle = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}, \qquad |y\rangle = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}, \qquad \langle y| = \begin{bmatrix} \bar{y}_0 & \bar{y}_1 \end{bmatrix}, \implies |x\rangle\langle y| = \begin{bmatrix} x_0\bar{y}_0 & x_0\bar{y}_1 \\ x_1\bar{y}_0 & x_1\bar{y}_1 \end{bmatrix} \tag{38}$$

Note that outer products do not generalize well beyond 2 systems, but it allows us to analyze systems as matrices.

Note (Independence)**.** In probability, $r = p \otimes q$ for some p, q, then they are independent random variables. If $r \neq p \otimes q$ for any p, q, then random variables are non-independent. Can think about this in terms of free parameters.

In the quantum analogy, $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ is a **product state**, and $|\psi\rangle \neq |\psi_1\rangle \otimes |\psi_2\rangle$ means $|\psi\rangle$ is entangled.

For n qubits, $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$ is a n-fold product state. This product state can be describe in $2n$ parameters, while an entangled state has 2^n parameters.

Claim. Let the state be

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{39}$$

We claim $|\psi\rangle \neq |\alpha\rangle \otimes |\beta\rangle$. Note this is true, which we can observe by writing in matrix form and seeing no outer product exists (rank 2 matrix vs. rank 1 outer product). So, this is entangled: this is called a **Bell/EPR pair**.

6.2 Unitary transformations

U acting on system 1, and V acting on system 2. We want these to be independent systems, so

$$|\alpha\rangle \otimes |\beta\rangle \to U|\alpha\rangle \otimes V|\beta\rangle \tag{40}$$

At the same time, QM should be linear, so their **joint action** should be $U \otimes V$.

In this notation, acting only on system 1 is $U \otimes I$, and analogously for system 2 is $I \otimes V$. Also,

$$(U \otimes I)(I \otimes V) = (I \otimes V)(U \otimes I) = U \otimes V \tag{41}$$

To perform tensor product multiplication, we have

$$\boxed{(A \otimes B)(C \otimes D) = AC \otimes BD.} \tag{42}$$

Next time: measuring tensor product states.

7 September 22nd, 2021

Today the guest lecture is given by Ike Chuang.

7.1 Entangled states

Definition 7.1 (Entanglement)**.** A state $|\psi_{AB}\rangle$ of a *bipartite system* is **entangled** iff

$$\nexists |\psi_A\rangle, |\psi_B\rangle : |\psi_{AB}\rangle = |\psi_A\rangle \otimes |\psi_B\rangle \tag{43}$$

Example 7.2. Consider the following examples of (not) entanglement:

1. Bell states or EPR pairs: $|\Phi_\pm\rangle = \frac{|00\rangle \pm |11\rangle}{\sqrt{2}}, |\Psi_\pm\rangle = \frac{|01\rangle \pm |10\rangle}{\sqrt{2}}$ Entangled.
2. $|00\rangle + |11\rangle + |22\rangle$ not entangled.
3. $$\begin{aligned} (|00\rangle + |11\rangle)^{\otimes 2} &= |00\ 00\rangle + |00\ 11\rangle + |11\ 00\rangle + |11\ 11\rangle && \text{labelling first tensor, second tensor} \quad (44) \\ &= |00\ 00\rangle + |01\ 01\rangle + |10\ 10\rangle + |11\ 11\rangle && \text{grouping first qubit, second qubit} \quad (45) \\ &= |0\ 0\rangle + |1\ 1\rangle + |2\ 2\rangle + |3\ 3\rangle && (46) \\ &= \sum_{x=0}^{3} |xx\rangle && (47) \end{aligned}$$

   Not entangled.
4. Generalizing the above,

   $$(|00\rangle + |11\rangle)^{\otimes n} = \sum_{x=0}^{2^n - 1} |xx\rangle \tag{48}$$

   Not entangled.
5. $|00\rangle + |01\rangle + |10\rangle + |11\rangle = (|0\rangle + |1\rangle)(|0\rangle + |1\rangle)$ not entangled.
6. $|00\rangle + |01\rangle + |10\rangle$ is entangled.
7. $\sqrt{0.99999}|00\rangle + \sqrt{0.000001}|11\rangle$ is entangled, but not very much.
8. GHZ state: $|000\rangle + |111\rangle$. Every single bipartition of this three-qubit state is entangled, so this state is entangled.

Claim. Almost all quantum states are entangled!

7.2 Quantum circuits

Recall single qubit gates.

Example 7.3 (Single qubit gates)**.** The Pauli matrices are below:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{49}$$

We can see each gate as a wire with a box in it, although the identity matrix is just a wire.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{50}$$

Note that

$$X|0\rangle = |1\rangle, \qquad H|0\rangle = |0\rangle + |1\rangle, \qquad H|1\rangle = |0\rangle - |1\rangle \tag{51}$$

Example 7.4 (Two qubit gates)**.** Recall for two-qubit gates, we can write them as 4-vectors $|00\rangle = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix}$. Consider

$$U|00\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} = |\Phi_+\rangle \tag{52}$$

We did not specify the middle, so we let it be another Hadamard (nested Hadamards!)

$$U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \tag{53}$$

Here, the columns are the states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$.

First matrix is the MSB (most significant bit) and the second matrix is the LSB:

$$X \otimes I = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{54}$$

We put the LSB in the spots of the MSB.

Example 7.5 (Hadamard)**.**

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ I & -I \end{bmatrix} \tag{55}$$

Example 7.6 (CNOT)**.**

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{56}$$

So it acts as

$$\text{CNOT}|a\ b\rangle = |a\ (a \oplus b)\rangle \tag{57}$$

7.3 Entangling circuits

For a two qubit system, we can draw a circuit by writing the MSB on top and the LSB on the bottom, and we point the states into the circuit.

$$U = H \oplus CNOT \tag{58}$$

$\lvert\psi_0\rangle$	$\lvert\psi_1\rangle$	$\lvert\psi_2\rangle$
00	(0+1)0 = 00 + 10	00+11 = Φ_+
01	(0+1)1 = 01+ 11	01 + 10 = Ψ_+
10	(0-1)0 = 00 - 10	00 - 11 = Φ_-
11	(0 -1) 1 = 01 - 11	01 - 10 = Ψ_-

Thus, we can write

$$U = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} I & I \\ X & -X \end{bmatrix} \tag{59}$$

Multiple qubits

$$(|00\rangle + |11\rangle)|00\rangle = |00\ 00\rangle + |11\ 00\rangle \implies |0000\rangle + |1111\rangle \tag{60}$$

This is useful for interferometry!

7.4 Entanglement and precision measurement

Given $\boxed{\varphi} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}$, we wish to determine φ using $\boxed{\varphi}$ as few times as possible.

Inteferometry

$$|0\rangle - \boxed{H} - |\psi_1\rangle - \boxed{\varphi} - |\psi_2\rangle - \boxed{H} - |\psi_3\rangle - \boxed{Measurement} \tag{61}$$

We can calculate φ with $\Delta\varphi = \frac{1}{\sqrt{n}}$ uses. [details] We can use n uses of $\boxed{\phi}$ to measure this to $\Delta\varphi = \frac{1}{n}$ precision with entanglement, with 1 Hadamard and several CNOTs.

8 September 24th, 2021

Announcements

- Wednesday lecture had no audio, sad. Find lecture notes on Canvas!

8.1 Tensor product of measurements

Recall a *von Neumann* measurement corresponds to a set of Hermitian projectors $\Pi_1, \cdots \Pi_j$ with

$$\sum_i \Pi_i = I, \qquad \Pi_i^2 = \Pi_i, \qquad \Pi_i^\dagger = \Pi_i. \tag{62}$$

We call Π_i a *measurement outcome.*

Theorem 8.1. *For projectors Π_A, Π_B on Hilbert spaces A, B, $\Pi_A \otimes \Pi_B$ is a projector onto $A \otimes B$.*

Proof. $(\Pi_A \otimes \Pi_B)^2 = \Pi_A^2 \otimes \Pi_B^2 = \Pi_A \otimes \Pi_B$. □

Theorem 8.2. *If $\Pi_1, \cdots, \Pi_{k_A}$ and $\Pi'_1, \cdots, \Pi'_{k_B}$ are measurements, then $\{\Pi_i \otimes \Pi'_j\}_{1\le i\le k_A, 1\le j\le k_B}$ is a measurement.*

Proof. We need to show that the sum of these is the identity on the tensor product space $A \otimes B$. We have

$$\sum_{j=1}^{k_B}\sum_{i=1}^{k_A} \Pi_i \otimes \Pi'_j = \left(\sum_{i=1}^{k_A} \Pi_i\right) \otimes \left(\sum_{j=1}^{k_B} \Pi_j\right) = I_A \otimes I_B = I_{A\otimes B}. \tag{63}$$

□

Example 8.3. Suppose we have state $|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle_{AB} + \frac{1}{2}|01\rangle_{AB} + \frac{1}{\sqrt{2}}|10\rangle_{AB}$ and we measure the second qubit in the $\{|0\rangle, |1\rangle\}$ basis.

Then

$$P(|0\rangle_B) =_B \langle 0|\psi\rangle_{AB} = \frac{1}{2}|0\rangle_A + \frac{1}{\sqrt{2}}|1\rangle_A \text{ with probability } \frac{3}{4} \tag{64}$$

We can normalize the resulting state. This is a von Neumann measurement with projectors $I_A \otimes |0\rangle_B\langle 0|_B, I_A \otimes |1\rangle_B\langle 1|_B$.

8.2 EPR Paradox

Consider rewriting the Bell state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ in the $\{|+\rangle, |-\rangle\}$ basis:

$$\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = \frac{1}{2\sqrt{2}}((|+\rangle + |-\rangle)(|+\rangle - |-\rangle) - ((|+\rangle - |-\rangle)(|+\rangle + |-\rangle))) = \frac{1}{\sqrt{2}}(|-+\rangle - |+-\rangle) \tag{65}$$

More generally, we can show that measuring the qubits in Bell state in any basis produces orthogonal states.

If Alice and Bob prepare this Bell state, if Alice and Bob measure simultaneously, Bob's qubit must "know" what result it will give to be compatible with Alice's measurement. There is no way to have QM where Bob's measurement of entangled state is not affected by the basis of Alice's measurement.

If there is no non-local correlation, information cannot be transmitted faster than light.

Bell showed no way to have a locally realistic theory.

9 September 27th, 2021

Announcements

- Shor going to Spain, notes to be prepared soon!

9.1 Joint observables

Recall that an observable is

$$M = \sum_i \lambda_i \Pi_i, \qquad \lambda_i \text{ an eigenvalue} \tag{66}$$

and a measurement is a projection onto one of the eigenspaces.

Suppose two observables M_λ, M_μ giving values of λ on space A and μ on space B. The observable for $\lambda\mu$ (λ and μ) is $M_\lambda \otimes M_\mu$ and the observable for $\lambda + \mu$ is $M_\lambda \otimes I + I \otimes M_\mu$.

9.2 Gates

9.2.1 Classical analog

We recall that AND, OR, and NOT gates are *universal*, meaning we can construct *any* Boolean function from $\wedge, \vee, \neg$ gates. Note that these are all 2-bit gates.

Theorem 9.1 (Extended Church-Turing thesis)**.** *Any efficiently computable function can be expressed with a relatively small number of 2-bit gates. More precisely, the number of gates is polynomial in the input.*

9.2.2 Quantum gates and circuits

Examples of quantum gates are CNOT and the Pauli matrices.

Definition 9.2 (Quantum gate)**.** A *quantum gate* is a $2^n \times 2^n$ unitary matrix where n is the number of qubits.

Example 9.3 (CNOT)**.** The CNOT gate is given by

$$\text{CNOT} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{bmatrix} \tag{67}$$

We can build quantum circuits out of 1 and 2-qubit gates. We represent qubits as wires, and gates as lines connecting wires.

Note. Any Boolean function can be constructed out of AND, OR, NOT gates, and Toffoli gates can produce any reversible Boolean function.

Theorem 9.4. *There is no finite set of quantum gates that will produce any unitary transformation.*

However, we can approximate any unitary transformation arbitrarily well with a finite gate set (Solovay-Kitaev theorem).

We show a simpler theorem.

Theorem 9.5. *Any unitary transformation can be produced using CNOT gates and (a small set of) one-qubit gates.*

Note that

$$e^{-i\phi\sigma_z} = \begin{bmatrix} e^{-i\phi} & 0 \\ 0 & e^{i\phi} \end{bmatrix} \tag{68}$$

So, $e^{-i\epsilon\sigma_z}$ is an infinitesimal rotation around the z-axis.

$$e^{-i\phi\sigma_y} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \tag{69}$$

Theorem 9.6. *Combining these and defining $R_x(\theta) = e^{-i\frac{\theta}{2}\sigma_x}$ and similarly for R_y and R_z, we can represent any unitary transformation (which is on the Bloch sphere) by applying*

$$R_z(\theta_1)R_y(\theta_2)R_z(\theta_3) \tag{70}$$

We can get any unitary with determinant 1, but global phase is irrelevant in QM, so this is sufficient.

9.2.3 Properties of Pauli matrices

Recall

$$\sigma_x = \begin{bmatrix} & 1 \\ 1 & \end{bmatrix}, \qquad \sigma_y = \begin{bmatrix} & -i \\ i & \end{bmatrix}, \qquad \sigma_z = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix}. \tag{71}$$

We have

$$\boxed{\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I, \qquad \sigma_j\sigma_k = i\varepsilon_{jkl}\sigma_l.} \tag{72}$$

9.2.4 Hadamard transform

Definition 9.7 (Hadamard transform)**.** The *Hadamard transform* is

$$H_0 = 1, \qquad H_m = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{bmatrix} \tag{73}$$

with properties $H^2 = 1, H\sigma_z H = \sigma_x$. Alternatively,

$$H_m = H_1 \otimes H_{m-1}. \tag{74}$$

10 September 29th, 2021

10.1 Gates for universal quantum computation

Any arbitrary unitary can be represented with CNOT, C-$R_y(\theta)$, and C-$R_z(\theta)$ gates. A C-U gate applies I to the target qubit if the control qubit is $|0\rangle$ and U if the control qubit is $|1\rangle$.

We note that

$$\sigma_x R_z(\theta)\sigma_x = R_z(-\theta), \qquad R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & \\ & e^{i\theta/2} \end{pmatrix} \tag{75}$$

We can construct these control-rotations based on a control bit with a cleverly designed negative rotation.

For an arbitrary $U = R_z(\theta_3)R_y(\theta_2)R_z(\theta_1)$, we can implement C-$U$ with 1-qubit gates. Note that this is because an arbitrary unitary matrix is a rotation of the Bloch sphere.

Note. There exists a circuit with six CNOTs and nine 1-qubit gates that does a Toffoli gate, a CCNOT.

Moreover, we can construct doubly controlled not gates, and C^k-NOT gates where the first k bits must all be $|1\rangle$ to apply the NOT gate.

10.1.1 Two-level matrices

This way, we can construct arbitrary unitary matrices by composition of two-level matrices.

Definition 10.1 (Two-level matrices)**.** We consider the following two-level matrix:

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

where $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is a unitary matrix.

A two-level matrix only affects the two rows and two columns which are non-identity.

We can construct two-level matrices with multiply controlled gates with $n-1$ control qubits and 1 target qubit, so that the unitary is in the last two rows/columns. We can move it to any arbitrary two row/columns with a sequence of controlled NOTs, which we will discuss next time.

11 October 1st, 2021

11.1 Two-level gates

Last time, showed *any* unitary can be decomposed into a product of *two-level matrices.* Recall that a two-level matrix is given by

$$\begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & \alpha & & \beta \\ & & & & 1 & \\ & & & \delta & & \gamma \end{bmatrix}, \qquad \text{where } \begin{bmatrix} \alpha & \beta \\ \delta & \gamma \end{bmatrix} \text{ unitary} \tag{76}$$

We also showed that we can construct C$n-1$-U with Toffoli, CNOT, and $R_y(\theta), R_z(\theta)$ gates.

We want to be able to construct a two-level gate from C^{n-1}-U. If we consider a general permutation matrix P, the two-level gate is given by $P^{-1}C^{n-1}UP$.

11.2 Density matrices

We want to represent probabilistic mixtures of quantum states, for example if we prepare a state that gives $|0\rangle, |+\rangle, |-\rangle$ with probabilities $\frac{1}{2}, \frac{1}{3}, \frac{1}{6}$, respectively.

We can consider the density matrix given by

$$\rho = \sum_i p_i |v_i\rangle\langle v_i| \tag{77}$$

where p_i is the probability of seeing $|v_i\rangle$. Important properties to note are: $\text{Tr}\rho = 1, \rho$ is positive semi-definite and $\rho = \rho^\dagger$.

Note. Any positive semi-definite matrix on a quantum state is a density matrix.

Example 11.1. If we prepare a state that gives $|0\rangle, |+\rangle, |-\rangle$ with probabilities $\frac{1}{2}, \frac{1}{3}, \frac{1}{6}$, respectively, we have

$$\rho = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{3}|+\rangle\langle +| + \frac{1}{6}|-\rangle\langle -| = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} + \frac{1}{6}\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{3}{4} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{4} \end{bmatrix}. \tag{78}$$

We note that **multiple** probability distributions can give the **same** density matrix. Why do we use the density matrix, if it is an incomplete description of the system? It turns out it is sufficient to determine any experimental outcome:

Example 11.2. Consider

$$\frac{1}{2}\left(\frac{4}{5}|0\rangle + \frac{3}{5}|1\rangle\right), \qquad \frac{1}{2}\left(\frac{4}{5}|0\rangle - \frac{3}{5}|1\rangle\right). \tag{79}$$

If we want to know what's going to happen in a quantum system, we simply need the density matrix.

11.2.1 Probability of measurements

Suppose we have $|v_i\rangle$ with probability p_i. We want the probability of getting $|w\rangle$ in some basis of $\{|w_j\rangle\}$ given by

$$\sum_i p_i |\langle w|v_i\rangle|^2 = \sum_i p_i \langle w|v_i\rangle\langle v_i|w\rangle = \langle w| \sum_i p_i |v_i\rangle\langle v_i|w\rangle = \langle w|\rho|w\rangle \tag{80}$$

which is independent of the basis of the density matrix.

Definition 11.3 (Mixed quantum state)**.** A *mixed* quantum state has ρ rank > 1. A *pure* quantum state has ρ rank 1.

Moreover, $\alpha|0\rangle + \beta 1 + \gamma|2\rangle$ is called a *superposition* of quantum states and $p_1|0\rangle\langle 0| + p_2|+\rangle\langle +| + p_3|-\rangle\langle -|$ is called a *mixture* of quantum states.

Note. A *superposition* of states is not the same as a *mixture* of states. A superposition of states could be $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, while a mixed state could be $|0\rangle$ with probability $\frac{1}{2}$ and $|1\rangle$ with probability $\frac{1}{2}$. While in both cases, we measure $|0\rangle$ and $|1\rangle$ with equal probability, they do not behave the same, for example when a Hadamard gate is applied. A superposition is an inherently quantum mechanical property of the system; a mixed state reflects our uncertainty about how the system was prepared.

11.2.2 von Neumann measurements

More generally, suppose we have von Neumann measurement projectors $\Pi_1, \Pi_2, \cdots, \Pi_k$ with $\sum_i \Pi_i = I$. We want to show that results depend *only on* ρ. Assume states $|v_i\rangle$ with probability p_i.

We see projector Π_j with probability

$$\sum_i p_i \langle v_i|\Pi_j|v_i\rangle = \sum_i p_i \mathrm{Tr}\Pi_j|v_i\rangle\langle v_i| = \mathrm{Tr}\Pi_j\left(\sum_i p_i \mathrm{Tr}|v_i\rangle\langle v_i|\right) = \mathrm{Tr}\Pi_j\rho \tag{81}$$

where we have used the trace of a scalar and the cyclic property of trace, where $\mathrm{tr}(ABC) = \mathrm{tr}(BCA)$.

The **reduced state** (after measurement) of observing the jth measurement outcome having started in the $|v_i\rangle$ state is simply

$$\frac{\Pi_j|v_i\rangle}{|\Pi_j|v_i\rangle|} = \frac{\Pi_j|v_i\rangle}{\sqrt{\langle v_i|\Pi_j|v_i\rangle}}. \tag{82}$$

Using Bayes' Rule, we derive the **conditional probability** we start with $|v_i\rangle$ given that we observe the jth outcome

$$\frac{p_i\langle v_i|\Pi_j|v_i\rangle}{\sum_i p_i\langle v_i|\Pi_j|v_i\rangle} = \frac{p_i\langle v_i|\Pi_j|v_i\rangle}{\mathrm{Tr}(\Pi_j\rho)}. \tag{83}$$

Then, we can calculate the **conditional density matrix** given that we observe the jth outcome:

$$\implies \rho_j = \sum_i \underbrace{\frac{\Pi_j|v_i\rangle\langle v_i|\Pi_j}{\langle v_i|\Pi_j|v_i\rangle}}_{\text{the state we see}} \times \underbrace{\frac{p_i\langle v_i|\Pi_j|v_i\rangle}{\mathrm{tr}\Pi_j\rho}}_{\text{probability of obtaining result}} = \sum_i \frac{\Pi_j p_i|v_i\rangle\langle v_i|\Pi_j}{\mathrm{tr}\Pi_j\rho} = \frac{\Pi_j\rho\Pi_j}{\mathrm{tr}\Pi_j\rho} \tag{84}$$

where ρ_j is the density matrix given we see the jth outcome.

Suppose we have two density matrices ρ_1, ρ_2 such that $\rho_1 \neq \rho_2$. Claim: $\rho_1 - \rho_2 \neq 0 \implies \rho_1 - \rho_2$ has nonzero eigenvalues. Use one eigenvector from them. We have

$$\langle w|(\rho_1 - \rho_2)|w\rangle = \lambda \neq 0 \implies \langle w|\rho_1|w\rangle \neq \langle w|\rho_2|w\rangle \tag{85}$$

and ρ_1, ρ_2 do **not** have the same behavior, and we can experimentally distinguish between them.

12 October 4th, 2021

12.1 Density matrices and partial trace

Last time from probabilistic mixtures of states. Today, we will talk about density matrices from missing information about parts of *entangled quantum states.*

Example 12.1. Take $|\psi\rangle = \frac{2}{\sqrt{5}}|00\rangle + \frac{1}{\sqrt{5}}|11\rangle$. Alice measures in the $\{|0\rangle, |1\rangle\}$ basis, Alice gets $|0\rangle$ (Bob gets $|1\rangle$) with probability $\frac{4}{5}$ and $|1\rangle$ (Bob gets $|0\rangle$) with probability $\frac{1}{5}$.
If Alice measures in the $\{|+\rangle, |-\rangle\}$ basis. Alice gets $\langle +|\psi\rangle$ and $\langle -|\psi\rangle$, each with probability $\frac{1}{2}$.

Thus a density matrix comes from discarding half of an entangled system. Suppose we have $|\psi\rangle_{AB}$. Alice's density matrix is the partial trace over B of $\rho, \mathrm{tr}_B|\psi\rangle\langle\psi|$ and Bob's density matrix is $\mathrm{tr}_A|\psi\rangle\langle\psi|$. We show that Bob's density matrix doesn't depend on Alice's basis for measurement in general.

Definition 12.2 (Partial trace)**.** Suppose we have a block matrix

$$\rho = \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \text{ on } A \otimes B. \tag{86}$$

Then

$$\mathrm{tr}_A\rho = P + S, \qquad \mathrm{tr}_B\rho = \begin{bmatrix} \mathrm{tr}P & \mathrm{tr}Q \\ \mathrm{tr}R & \mathrm{tr}S \end{bmatrix}. \tag{87}$$

These results generalize in a straightforward way even to non-qubits, when A has dimension $j \times j$ and B has dimension $k \times k$.

Note. If you completely ignore Alice, Bob's system behaves like $\mathrm{tr}_A\rho_{AB}$. If Bob does not know what Alice is doing, his state should behave the same whether or not Alice has measured it.

We show an example of the above calculation, but omit it from these notes.

Alternatively, we can represent the trace in a general form based on the basis of each system.

$$\mathrm{tr}_A\rho = \sum_{|e_i\rangle_A} {}_A\langle e_i|\rho|e_i\rangle_A, \qquad \mathrm{tr}_B\rho = \sum_{|e_i\rangle_B} {}_B\langle e_i|\rho|e_i\rangle_B \tag{88}$$

where $|e_i\rangle_A$ is any basis of A and likewise for $|e_i\rangle_B$.

Note. The projection matrix represented by $|0\rangle_{AA}\langle 0|$ is actually $|0\rangle_{AA}\langle 0| \otimes I_B$.

Theorem 12.3 (Independence of basis)**.** *Suppose we have two bases $|e_i\rangle_A, |f_i\rangle_A$. Then $|f_i\rangle = \alpha_{ij}|e_j\rangle$ where α_{ij} is a unitary matrix. Then*

$$tr_A^{(f)}\rho = \sum_i \langle f_i|\rho|f_i\rangle = \sum_{j'}\sum_j \alpha_{ij}^*\langle e_j|\rho|e_{j'}\rangle\alpha_{ij'} = \sum_{j'}\sum_j \langle e_{j'}|\rho|e_j\rangle \underbrace{\sum_i \alpha_{ij'}^*\alpha_{ij}}_{\delta_{jj'}} = \sum_j \langle e_j|\rho|e_j\rangle = tr_A^{(e)}\rho \tag{89}$$

*so the partial trace is **independent of basis**.*

Suppose we have a *tensor product matrix.* Then

$$\mathrm{tr}_A\left(\rho_A \otimes \rho_B\right) = \rho_B\mathrm{tr}\rho_A, \qquad \mathrm{tr}_B\left(\rho_A \otimes \rho_B\right) = \rho_A\mathrm{tr}\rho_B \tag{90}$$

13 October 6th, 2021

No recording today.

13.1 No-cloning theorem

Theorem 13.1 (No cloning theorem)**.** *No quantum operation takes the state $|\psi\rangle \otimes |0\rangle \to |\psi\rangle \otimes |\psi\rangle$.*

Proof. We will use density matrices to show this fact, although there exist more general methods. Consider if Alice and Bob hold the Bell state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. This state has the property that Alice and Bob will always get orthogonal vectors if they measure with the same basis.

If Alice measures in the $\{|0\rangle, |1\rangle\}$ basis, Bob has the density matrix

$$\rho_B = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \tag{91}$$

If Alice measures in the $\{|+\rangle, |-\rangle\}$ basis, Bob has the density matrix

$$\rho_B = \frac{1}{2}(|+\rangle\langle +| + |-\rangle\langle -|) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \tag{92}$$

These are the same density matrix, so Bob cannot tell what basis Alice used.

However, consider if Bob can clone his state after Alice has measured. Then, if Alice measures in the $\{|0\rangle, |1\rangle\}$ basis, Bob has the density matrix

$$\rho_B = \frac{1}{2}(|00\rangle\langle 00| + |11\rangle\langle 11|) = \begin{pmatrix} \frac{1}{2} & & & \\ & & & \\ & & & \\ & & & \frac{1}{2} \end{pmatrix} \tag{93}$$

However, if Alice measures in the $\{|+\rangle, |-\rangle\}$ basis, Bob has the density matrix

$$\rho_B = \frac{1}{2}(| + +\rangle\langle + + | + | - -\rangle\langle - - |) = \begin{pmatrix} \frac{1}{4} & & & \frac{1}{4} \\ & \frac{1}{4} & \frac{1}{4} & \\ & \frac{1}{4} & \frac{1}{4} & \\ \frac{1}{4} & & & \frac{1}{4} \end{pmatrix} \tag{94}$$

Since these two density matrices are different, then Bob can tell which basis Alice used. But without classical communication from Alice, he should be able to make this cloning/measurement immediately after Alice's measurement, meaning she transmits information to him faster than the speed of light, which is a contradiction.

□

13.2 Quantum gates and circuits

Definition 13.2 (Hadamard gate)**.** The Hadamard gate is

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \tag{95}$$

H is its only inverse: $H^2 = I$. Moreover we have that

$$H\sigma_x H = \sigma_z, \qquad H\sigma_z H = \sigma_x, \qquad H\sigma_y H = \sigma_y^\dagger \tag{96}$$

The tensor product of n Hadamard gates is the *Hadamard transform.*

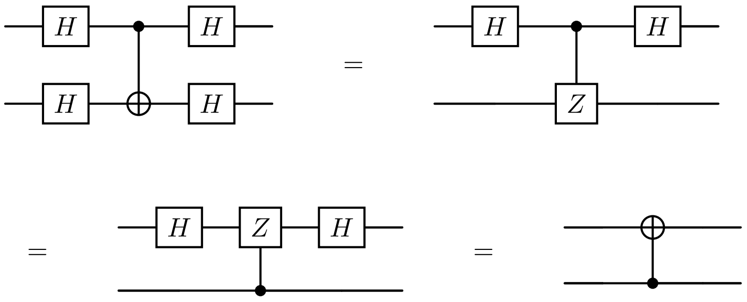
13.2.1 CNOT and Hadamard gate identity

First, consider the following result:

$$\mathrm{CNOT}|-\rangle|-\rangle = \mathrm{CNOT}\frac{1}{2}(|0\rangle - |1\rangle)(|0\rangle - |1\rangle) = \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) \tag{97}$$

In this case, the target qubit stayed the same, and the control bit changed!

Now we consider the quantum circuit



where we have used the identity that $H\sigma_x H = \sigma_z$, and a C-Z is the same even if the control and target qubits are flipped (since the action on the qubits is symmetric up to a phase). That is, SWAP $\circ$ C-Z $\circ$ SWAP $=$ C-Z.

Note. Thus, we've shown that putting two H gates before/after a CNOT reverses the direction of a CNOT.

This is an example of a more general QM principle: “back-action."

14 October 8th, 2021

14.1 Points on a Bloch sphere

Lemma 14.1. *Let $\{|v_0\rangle, |v_1\rangle, \ldots, |v_{k-1}\rangle\}$ be an orthonormal basis for a quantum state space. Then,*

$$\sum_{i=0}^{k-1} |v_i\rangle\langle v_i| = I \tag{98}$$

This can be easily shown by applying a unitary transformation.

Note that any arbitrary point on the Bloch sphere can be represented via rotation transformations,

$$(\alpha, \beta, \gamma) = (\cos\phi \sin\theta, \sin\phi \sin\theta, \cos\theta) = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \tag{99}$$

Consider a point (α, β, γ) on the Bloch sphere and consider looking at the matrix

$$M_{\alpha,\beta,\gamma} = \alpha\sigma_x + \beta\sigma_y + \gamma\sigma_z, \qquad \text{where } \alpha^2 + \beta^2 + \gamma^2 = 1. \tag{100}$$

Note that

$$\text{tr}(\alpha\sigma_x + \beta\sigma_y + \gamma\sigma_z) = 0, \qquad (\alpha\sigma_x + \beta\sigma_y + \gamma\sigma_z)^2 = I \implies \alpha\sigma_x + \beta\sigma_y + \gamma\sigma_z = |v\rangle\langle v| - |\bar{v}\rangle\langle\bar{v}|. \tag{101}$$

where $M^2_{\alpha,\beta,\gamma} = I \implies$ its eigenvalues are ± 1. This matrix is also Hermitian and thus diagonalizable.

Then we can derive $|v\rangle$ from $M_{\alpha,\beta,\gamma}$ by

$$\frac{1}{2}(I + \alpha\sigma_x + \beta\sigma_y + \gamma\sigma_z) = \frac{1}{2}(|v\rangle\langle v| + |\bar{v}\rangle\langle\bar{v}| + |v\rangle\langle v| - |\bar{v}\rangle\langle\bar{v}|) = |v\rangle\langle v|. \tag{102}$$

A density matrix ρ corresponds to a point *inside* the Bloch sphere and *on* the Bloch sphere if ρ is rank 1.

Note. I and the Pauli matrices $(\sigma_x, \sigma_y, \sigma_z)$ form a basis for 2×2 matrices.

If ρ a density matrix, then $\text{tr}\rho = 1$. We can write

$$\rho = \zeta I + \alpha\sigma_x + \beta\sigma_y + \gamma\sigma_z \tag{103}$$

for any density matrix. Note that $\text{tr}\rho = 1, \text{tr}I = 2 \implies \zeta = \frac{1}{2}$. From ρ, we can get coordinates on the Bloch sphere.

Note. $\alpha^2 + \beta^2 + \gamma^2 > 1 \implies \rho$ has negative eigenvalues.

We then show the example for deriving the state $\boldsymbol{v}$ for a point $(\alpha, \beta, \gamma) = \left(\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}\right)$, halfway between $(1, 0, 0)$ $(|+\rangle)$ and $(0, 0, 1)$ $(|0\rangle)$.

14.1.1 Two points on the Bloch sphere

Suppose we have two points on the Bloch sphere, $|v_1\rangle, |v_2\rangle$.

$$|v_1\rangle\langle v_1| = \frac{1}{2}I + \frac{\alpha_1\sigma_x + \beta_1\sigma_y + \gamma_1\sigma_z}{2}, \qquad |v_2\rangle\langle v_2| = \frac{1}{2}I + \frac{\alpha_2\sigma_x + \beta_2\sigma_y + \gamma_2\sigma_z}{2}. \tag{104}$$

We want to determine the relationship between the angles $|\langle v_1|v_2\rangle| \equiv \cos\varphi$ and $(\alpha_1, \beta_1, \gamma_1) \cdot (\alpha_2, \beta_2, \gamma_2) = \cos\theta$. We have

$$\mathrm{tr}|v_1\rangle\langle v_1||v_2\rangle\langle v_2| = \mathrm{tr}\langle v_2|v_1\rangle\langle v_1|v_2\rangle = \cos^2\varphi \tag{105}$$

and

$$\begin{aligned}&\frac{1}{4}\mathrm{tr}\left[(I + \alpha_1\sigma_x + \beta_1\sigma_y + \gamma_1\sigma_z)(I + \alpha_2\sigma_x + \beta_2\sigma_y + \gamma_2\sigma_z)\right]\\ &\qquad = \frac{1}{4}\mathrm{tr}\left(I + \alpha_1\alpha_2 I + \beta_1\beta_2 I + \gamma_1\gamma_2 I + \cdots\right) = \frac{1}{2}\left(1 + (\alpha_1, \beta_1, \gamma_1) \cdot (\alpha_2, \beta_2, \gamma_2)\right) = \frac{1 + \cos\theta}{2}.\end{aligned} \tag{106}$$

where we drop the terms which are products of Pauli matrices (and are thus Pauli matrices themselves), since they have trace 0. Thus, the only terms that remain are those with $\sigma_i\sigma_i = I$, with trace 2.

Setting these equal, we have

$$\cos^2\varphi = \frac{1 + \cos\theta}{2} \implies \varphi = \frac{\theta}{2}. \tag{107}$$

14.2 Information in qubits

Theorem 14.2. *You cannot send an unknown qubit without classical information*

Proof. Suppose you could. Then A could send a qubit to B, who could duplicate the digital information and send it to C, who would then have cloned the qubit, violating the no-cloning theorem. □

So you can ask, how much information is in a qubit? If you have $\alpha|0\rangle + \beta|1\rangle$, then you have two complex numbers, which takes infinite bits to represent unless you specify some precision of the numbers. On the other hand, the theorem above implies no amount of bits can encode a qubit.

Theorem 14.3 (Holevo)**.** *You can only extract n bits of information from n qubits.*

How to resolve all these claims about the amount of information in qubits? Simply put, quantum information and classical information are *different.*

15 October 13th, 2021

Announcements

- Today, a lecture on quantum teleportation

15.1 Quantum teleportation

On last Friday, we showed you cannot send a qubit in isolation over a classical channel. This is actually not true if the sender and receiver share an EPR pair.

15.1.1 Teleportation protocol

Consider the following situation. They start out with a two qubit state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Alice has a different, unknown qubit $\alpha|0\rangle + \beta|1\rangle$ that she wants to send to Bob. Alice measures her unknown qubit and her qubit in the EPR pair, and then sends to Bob two classical bits. Then Bob will be able to use his half of the EPR pair and the two classical bits to reconstruct $|\psi\rangle$. This doesn't contradict the theorem from Friday, since any third person will not share the required EPR state with Alice or Bob to perform this action.

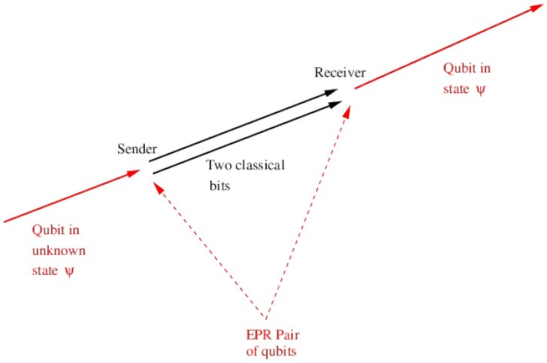


Figure 1: Teleportation protocol schematic

So how does Bob implement teleportation?

Definition 15.1. The Bell basis (for 2 qubits) is

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \qquad \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \qquad \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \qquad \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{108}$$

where all states are *maximally entangled.*

We start with $|\psi\rangle = \frac{1}{\sqrt{2}}(\alpha|0\rangle_{A_1} + \beta|1\rangle_{A_1}) \otimes (|00\rangle_{A_2B} + |11\rangle_{A_2B})$.

Alice then measures A_1, A_2 in the Bell basis, and sends the results to Bob. Some of the possible results after Alice's measurement are (each of these has probability 1/4 based on the four possible Bell basis states)

$$\left(\frac{1}{\sqrt{2}}(\langle 00|_{A_1A_2} + \langle 11|_{A_1A_2}) \otimes I\right)|\psi\rangle = \frac{1}{2}\left(\alpha|0\rangle_B + \beta|1\rangle_B\right) \tag{109}$$

$$\left(\frac{1}{\sqrt{2}}(\langle 01|_{A_1A_2} - \langle 10|_{A_1A_2}) \otimes I\right)|\psi\rangle = \frac{1}{2}\left(\alpha|1\rangle_B - \beta|0\rangle_B\right) \tag{110}$$

In the second case, we see

$$\sigma_y \frac{1}{2}(\alpha|1\rangle - \beta|0\rangle) = -i\frac{1}{2}(\alpha|0\rangle + \beta|1\rangle) \tag{111}$$

We observe that we can apply similar Pauli transformations for each of the four cases to transform between Bell states. We can thus see that each of Alice's 4 measurement results (each of the 4 Bell basis vectors) puts Bob's qubit in a state that, after you apply I or σ_i as necessary, results in $|\psi\rangle$. So Alice only needs to send which results she observed to Bob (which takes 2 bits), and Bob can then apply the correct σ_i or I to achieve the correct qubit, completing the teleportation.

Now, we can show the above more concisely with

$$\sigma_x \otimes \sigma_x \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{112}$$

$$\sigma_y \otimes \sigma_y \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{113}$$

$$\sigma_z \otimes \sigma_z \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{114}$$

Also we have

$$\frac{1}{2}(\langle 00| + \langle 11|)_{AA}|\psi\rangle_A \otimes (|00\rangle + |11\rangle)_{AB} = |\psi\rangle_B \tag{115}$$

and

$$\frac{1}{2}(\langle 00| + \langle 11|)\sigma_x^{A_1} \otimes \sigma_x^{A_2}|\psi\rangle_A(|00\rangle + |11\rangle)_{A_2B_2} = |\psi\rangle_B \tag{116}$$

$$\begin{aligned}&\frac{1}{2}(\langle 01| + \langle 10|)|\psi\rangle_{A_1}\sigma_x^{A_2}(|00\rangle + |11\rangle)_{A_2B_2} \\ &\qquad = |\psi\rangle_B = \frac{1}{2}(\langle 01| + \langle 10|)|\psi\rangle_{A_1}\sigma_x^{B_2}(|00\rangle + |11\rangle)_{A_2B_2} = \sigma_x^{B_2}|\psi\rangle_B\end{aligned} \tag{117}$$

$$\sigma_x^{B_2}\frac{1}{2}(\langle 01| + \langle 10|)|\psi\rangle_{A_1}\sigma_x^{B_2}(|00\rangle + |11\rangle)_{A_2B_2} = \sigma_x^{B_2}|\psi\rangle_B. \tag{118}$$

This is Bob's state when Alice gets the third basis element as her measurement, and we can repeat for the other basis elements.

15.1.2 Teleportation circuit

Let's construct a circuit on two qubits, followed by measurement on both qubits. We want the following measurements to correspond to the following input Bell basis states:

$$|00\rangle \to \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \qquad |01\rangle \to \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \qquad |10\rangle \to \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \qquad |11\rangle \to \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{119}$$

How can we construct this circuit that maps Bell states to $|00\rangle, |01\rangle, |10\rangle, |11\rangle$? We can construct the inverse first, which turns out to be a Hadamard on the first qubit followed by a CNOT. The inverse of this is a CNOT followed by a Hadamard on the first qubit.

So the total circuit for performing teleportation is, apply the CNOT and Hadamard above to the first two qubits, measure them and convert the information to two bits, and then apply σ_x if the first bit is 1 and apply σ_z if the second bit is 1 on the last qubit (Bob's qubit).

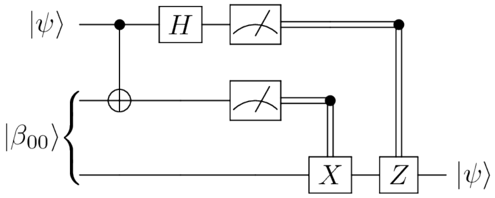


Figure 2: Teleportation circuit

In the below figure, the top two wires are Alice's qubits, while the bottom wire is Bob's. The bottom two wires are the entangled Bell pair. Double wires are classical bits, while single wires are quantum bits.

16 October 15th, 2021

Announcements

- Midterm in 1.5 weeks (not in the main room!)

16.1 Superdense coding

Superdense coding is the inverse of teleportation. With Alice and Bob sharing an EPR pair ($\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$), Alice can send 2 classical bits of information to Bob by sending one qubit.

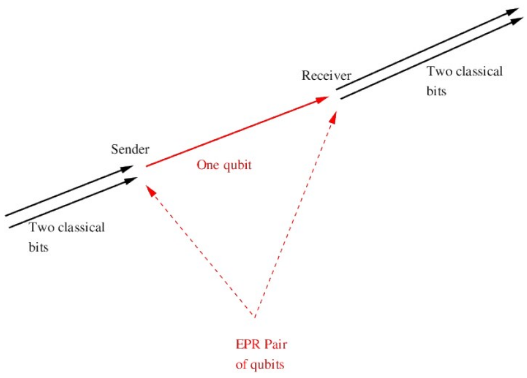


Figure 3: Superdense coding protocol schematic

Theorem 16.1 (Holevo bound)**.** *You can only encode n classical bits in n quantum bits.*

Note. This theorem does **not apply** because we share an EPR pair.

Recall there are four Bell states:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \qquad \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \qquad \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \qquad \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{120}$$

Alice applies one of the matrices $I, \sigma_x, \sigma_y, \sigma_z$ to her qubit (part of the EPR pair) to get one of these Bell states. Alice then sends her qubit in the EPR pair to Bob. Bob then measures in the Bell basis and learns Alice's Pauli matrix. This sends 2 bits of information.

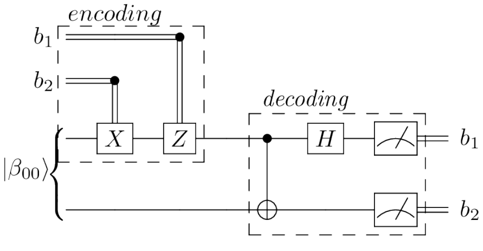


Figure 4: Superdense coding circuit

In the diagram, we note that double lines denote classical bits and single lines denote quantum bits.

Superdense coding was discovered as part of a proof that 1 qubit is exactly equal to 2 bits when sharing an EPR pair (can't transmit more qubits without more bits and vice versa), showing quantum teleportation is optimal.

16.2 Reversible gates

The Toffoli gate is a CCNOT gate. It can be used to make many gates:

$$\text{AND}(x, y) = \text{Toffoli}(x, y, 0) \to (x, y, x \wedge y), \qquad \text{XOR}(x, y) = \text{Toffoli}(x, 1, z) \to (x, 1, x \vee y), \tag{121}$$

$$\text{NOT}(x, y) = \text{Toffoli}(x, 1, 1) \to (x, 1, \neg x), \qquad \text{FANOUT}(x) = \text{Toffoli}(x, 1, 0) \to (x, 1, x) \tag{122}$$

with appropriate choices of which of the Toffoli outputs to extract. This makes the Toffoli gate *universal for reversible computation.*

We can write *any* classical circuit with Toffoli gates. This **is not good enough** because in quantum computation, we need to do interference.

Example 16.2. Suppose you have a state given by $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Applying an H gate sends it to $|0\rangle$. Suppose you obtained the $|0\rangle, |1\rangle$ states from Toffoli computations, which leave behind some left over state $|c_0\rangle, |c_1\rangle$. Then applying a Hadamard gate here actually gives (assuming $|c_1\rangle \neq |c_2\rangle$

$$H\frac{1}{\sqrt{2}}(|0\rangle \otimes |c_0\rangle + |1\rangle \otimes |c_1\rangle) = \frac{1}{\sqrt{2}}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}|c_0\rangle + \frac{|0\rangle - |1\rangle}{\sqrt{2}}|c_1\rangle\right) \tag{123}$$

which is not what we want. So, **we have to get rid of intermediate results**.

We will do this with FANOUT gates, which do not produce extraneous outputs (and thus have no intermediate results).

16.2.1 Computation of $f(x)$

Now let's consider a computation of some function $f(x)$, where we keep the input around because we want a *reversible computation.*

Step 1: We take the input, as well as work bits $0^a, 1^b, 0^c, 1$. (We require work bits that are both 0s and 1s because Toffoli gates don't generate any new bits, e.g. only input work 0s do not allow for 1s). Doing the computation with Toffoli gates will leaves us at the end with the input, the output, some junk (intermediate work space), 0^c, and 1.

Step 2: We can copy the output into 0^c with Toffoli gates because Toffoli$(x, 1, 0) = (x, 1, x)$. So now we have input, junk, output, output, 1.

Step 3: We can run all of the Toffoli gates that generated the junk and the first copy of output backwards, to get input, $0^a, 1^b$, output, 1. That leaves us with input, $0^a, 1^b$, output, 1, which has **no left over intermediate results** to mess up the interference.

So this is a way of performing reversible computation with Toffoli gates. But we need to keep the input around, since otherwise we can't apply Toffoli.

input	$000\ldots0$	$111\ldots1$	$000\ldots0$	1
input	output	junk	$000\ldots0$	1
input	output	junk	output	1
input	$000\ldots0$	$111\ldots1$	output	1

Figure 5: Schematic of reversible Toffoli computation

16.2.2 Computation without keeping input

If we have a classical circuit taking x to $f(x)$, and a classical circuit taking y to $g(y)$ where $g = f^{-1}$, then we can use the above circuit that keeps the input around, reverse it, and append it to the earlier computation to implement this.

17 October 18th, 2021

17.1 Deutsh-Jozsa algorithm

Definition 17.1. A *Hadamard transform* is a Hadamard gate on each qubit.

We want to see how $H^{\otimes n}|j\rangle$ behaves on some $|j\rangle$ where j is a binary string length n.

Example 17.2.

$$H^{\otimes 5}|0\rangle|1\rangle|1\rangle|0\rangle|0\rangle = \frac{1}{\sqrt{2^5}}(|00000\rangle + |00001\rangle + \cdots - |01011\rangle + \dots) \tag{124}$$

Claim.

$$H^{\otimes n}|j\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}(-1)^{j\cdot k}|k\rangle \tag{125}$$

where k ranges over all binary strings of length n and the $j \cdot k$ is the dot product in binary (number of places where both j and k have a 1): $01100 \cdot 01011 = 1$.

17.1.1 The Deutsh-Jozsa problem

Given a binary function $f : \{0,1\}^n \to \{0,1\}$, you are promised f is either *constant* or *balanced* (number of x such that $f(x) = 0$ is the same as the number for which $f(x) = 1$). Determine if f is constant or balanced.

Classically, it takes $2^{n-1} + 1$ time in the worst case, but is easy with a probabilistic algorithm.

17.1.2 Algorithm

We assume we have a *phase oracle* $U_f|x\rangle = (-1)^{f(x)}|x\rangle$ for $f(x)$ with binary output. This is called a phase oracle because it encodes information about the function in the *phase*. The 5-step algorithm is as follows: Start with $|0^n\rangle$. Apply $H^{\otimes n}$, then U_f, and then $H^{\otimes n}$. Then measure the bits of the quantum state individually. Let's see why this works:

$$H^{\otimes n}U_fH^{\otimes n}|0^n\rangle = H^{\otimes n}U_f\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)^{\otimes n} = H^{\otimes n}\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}|j\rangle = \frac{1}{2^n}\sum_{j=0}^{2^n-1}\sum_{k=0}^{2^n-1}(-1)^{f(j)}(-1)^{j\cdot k}|k\rangle \tag{126}$$

Case 1: Suppose f is constant. Then $f(j) = f(0)$. So,

$$H^{\otimes n}U_fH^{\otimes n}|0^n\rangle = \frac{1}{2^n}(-1)^{f(0)}\sum_{j,k=0}^{2^n-1}(-1)^{j\cdot k}|k\rangle = (-1)^{f(0)}|0\rangle \tag{127}$$

since if $k = 0$ you get $2^n|0\rangle$ from the sum and for $k \neq 0$ you get 0 from the sum since there are as many j's that give $j \cdot k \mod 2 = 0$ as those that give $j \cdot k \mod 2 = 1$. So, we see $|0^n\rangle$ with probability 1.

Case 2: Suppose f is balanced. Then the component for $k = 0$ is

$$\frac{1}{2^n}\sum_{j=0}^{2^n-1}(-1)^{f(j)}|0\rangle = 0 \tag{128}$$

Since the amplitude is 0, then the probability of seeing $|0^n\rangle$ is 0.

So, we can conclude **if we measure $|k\rangle = |0^n\rangle$, f is constant. If not, then it is balanced.**

This algorithm took one phase oracle call, and linear gates in n, while deterministic classical algorithm takes $2^{n/2}$ gates. So, this is exponentially faster. However, randomized algorithms can do very well with a constant number (20 or 30) queries, so this is not a very convincing argument that quantum computation is powerful.

17.2 Phase and bit oracle equivalence

Note (Phase oracle)**.** A phase oracle cannot tell the difference between f and $1-f$. Because

$$(-1)^{f(x)} = -(-1)^{1-f(x)} = \begin{cases} -1, & f(x)=0 \\ 1, & f(x)=1 \end{cases} \tag{129}$$

and they differ by a global phase.

Suppose we were given a bit oracle for f.

Definition 17.3 (Bit oracle)**.** A *bit oracle* O_f is

$$O_f|x\rangle|z\rangle = |x\rangle|z\oplus f(x)\rangle \implies O_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle. \tag{130}$$

We can also construct a *controlled bit oracle*:

$$CO_f|0\rangle|a\rangle|b\rangle = |0\rangle|a\rangle|b\rangle, \qquad CO_f|1\rangle|x\rangle|z\rangle = |1\rangle|x\rangle|z\oplus f(x)\rangle. \tag{131}$$

We would like to construct a phase oracle from a bit oracle. Consider

$$\sigma_z O_f|x\rangle|0\rangle = (-1)^{f(x)}|x\rangle|f(x)\rangle \implies O_f\sigma_z O_f|0\rangle = (-1)^{f(x)}|x\rangle|0\rangle. \tag{132}$$

This gives a phase oracle from a bit oracle.

18 October 20th, 2021

Announcements

- Midterm Wednesday

18.1 Classical complexity theory

In theoretical CS, we consider algorithms *efficient* if they are in the **class P**; that is, running in polynomial time in the length of the input. In 1936, Alan Turing defined Turing machines, which became standard objects to reason about computer science (although transformations between different classical computational models preserve P).

We now discuss the **class NP**, which are the set of problems which can be easily verified (e.g. 3-colorability of a graph). Formally, this requires a polynomial time algorithm that can take a problem instance and a witness and verify that the witness provides a solution to the problem (e.g. a 3-coloring, checking all edge pairs).

Steve Cook first defined **NP-completeness**, where if you can solve one problem in the class in polynomial time, you can solve all NP problems in polynomial time. Cook found a master problem that can solve any problem in NP.

Claim (Cook)**.** Given a TM and program/problem B is there an input of length n that runs in polynomial time and outputs “yes"?

We note the nested complexity classes given in the diagram

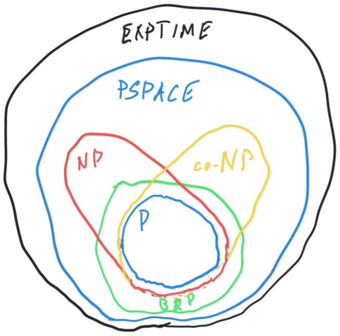


Figure 6: Relevant complexity classes

Definition 18.1 (BQP)**.** *BQP* is the class of problems that can be solved on a quantum computer with *high probability* in polynomial time.

Definition 18.2 (BPP)**.** *BPP* is the class of problems that can be solved on a classical computer with a random number generator with high probability in probability time.

There is a lot we do not know about relationships between these classes. For example, how do BQP and NP compare? In general, people believe they are incomparable. We also don't know if P=PSPACE. However, we do know that PSPACE=QPSPACE.

Theorem 18.3 (Church thesis)**.** *It doesn't matter how you define computable functions, any reasonable definition is equivalent to Turing machines.*

Theorem 18.4 (Cobham thesis)**.** *Any reasonable definition of machine computes the same class of functions in polynomial time. This is disproved by quantum computers!*

18.2 Oracles

We consider algorithms in group theory. For generality, we consider group elements are represented as **oracles**. In the oracle model, we assume different operations (O_m for multiplication, O_i for inverses) are oracles and can be used regardless of how it is implemented on a Turing machine (Church Thesis).

18.2.1 Quantum oracles

We show the equivalence of bit and phase oracles. Last time, we showed a bit oracle $\implies$ phase oracle, which we review. A bit oracle operates as

$$O_f|x\rangle|z\rangle = |x\rangle|z \oplus f(x)\rangle \tag{133}$$

where $|x| = n, |z| = 1$. We can construct a phase oracle by computing $|f(x)\rangle$ in the second register, applying a Z gate to get $(-1)^{f(x)}$, and then uncomputing $|f(x)\rangle$.

We now show phase oracle $\implies$ bit oracle. We assume $f(0) = 0$ to account for the possible global phase in the phase oracle. The basic idea is that we put it into the $\frac{1}{\sqrt{2}}(|0\rangle|0\ldots0\rangle + |1\rangle|x\rangle)$ state, and then use the phase oracle to turn $|+\rangle$ to $|-\rangle$.

In more detail, we put a register into $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, perform a conditional FANOUT to get $\frac{1}{\sqrt{2}}(|0\rangle|0\ldots0\rangle + |1\rangle|x\rangle)$.

$$|x\rangle|0\rangle|0\ldots0\rangle \to |x\rangle|+\rangle|0\ldots0\rangle \to \frac{1}{\sqrt{2}}(|x\rangle|0\rangle|0\ldots0\rangle + |x\rangle|1\rangle|x\rangle). \tag{134}$$

Now, the phase oracle is applied, after which we apply the conditional FANOUTS in reverse, taking it to

$$\begin{aligned} \to \frac{1}{\sqrt{2}}(|x\rangle|0\rangle|0\ldots0\rangle + (-1)^{f(x)}|x\rangle|1\rangle|x\rangle) \to \frac{1}{\sqrt{2}}(|x\rangle|0\rangle|0\ldots0\rangle + (-1)^{f(x)}|x\rangle|1\rangle|0\rangle) \\ = \frac{1}{\sqrt{2}}|x\rangle(|0\rangle + (-1)^{f(x)}|1\rangle)|0\ldots0\rangle \end{aligned} \tag{135}$$

Now, we apply the final Hadamard gate, which gives us

$$|x\rangle|f(x)\rangle|0\ldots0\rangle \tag{136}$$

which is the bit oracle for f we desired.

19 October 22nd, 2021

19.1 Simon's problem

This problem was invented to show quantum computers could be faster than classical computers. We are given a function $f : \{0,1\}^n \to \{0,1\}^n$ with the 2-to-1 property that $f(x) = f(x \oplus c)$ for some c. We are asked to find c for this function.

Classically, if we find $f(x_1) = f(x_2)$, then $c = x_1 \oplus x_2$. By trying out evaluations of inputs, we can choose them carefully to eliminate an additional $j-1$ possible values of c on the jth call to the function, one for each XOR with a previously tested value. So after t evaluations, we have eliminated $t(t-1)/2$ values for c. To be sure of finding c, we need

$$\frac{t(t-1)}{2} \geq 2^n - 1 \implies t \approx 2^{(n+1)/2} \tag{137}$$

evaluations.

If we consider classical randomized algorithms over functions with Simon's property, we know that c is equally likely to be among any of the values we have not tried. So, we need to eliminate $\frac{1}{2}$ of all possible values of c for a 50% chance of finding c, which has runtime $t \approx 2^{n/2}$.

19.2 Simon's algorithm

Assume we have a bit oracle

$$O_f|x\rangle|z\rangle \to |x\rangle|z \oplus f(x)\rangle. \tag{138}$$

Recall the Hadamard transform:

$$H^{\otimes n}|j\rangle = \sum_{k=0}^{2^n-1}(-1)^{j\cdot k}|k\rangle. \tag{139}$$

Simon's algorithm is the following:

$$\begin{aligned}(H\otimes I)O_f(H\otimes I)|0^n\rangle|0^n\rangle = (H\otimes I)O_f\frac{1}{2^{n/2}}\sum_{j=0}^{2^n-1}|j\rangle|0^n\rangle = (H\otimes I)\frac{1}{2^{n/2}}\sum_{j=0}^{2^n-1}|j\rangle|f(j)\rangle \\ = \frac{1}{2^n}\sum_{k=0}^{2^n-1}\sum_{j=0}^{2^n-1}(-1)^{j\cdot k}|k\rangle|f(j)\rangle\end{aligned} \tag{140}$$

We then measure the registers. Note that the probability of seeing $|k\rangle|f(j)\rangle$ is the square of the amplitude in this state; note only $f(j) = f(j\oplus c)$ can produce this value. So, the only amplitude contribution is

$$\frac{1}{2^n}\left((-1)^{j\cdot k} + (-1)^{(j\oplus c)\cdot k}\right) = \frac{1}{2^n}\left((-1)^{j\cdot k} + (-1)^{j\cdot k\oplus c\cdot k}\right) = \frac{1}{2^n}(-1)^{j\cdot k}(1+(-1)^{c\cdot k}) \tag{141}$$

This expression is 0 if $c\cdot k = 1$, and it is $\pm\frac{2}{2^n}$ if $c \cdot k = 0$. Then, we will never see k if $c\cdot k = 1$, and we will have probability $\frac{1}{2^{n-2}}$ of seeing k if it is perpendicular to c.

So, if we repeat the process and find $n-1$ linearly independent vectors perpendicular to c, then there is exactly one vector in $n-$dimensional space that is perpendicular to them all, so it must be c.

If we have some vectors, we can use *Gaussian elimination* to check that they are all linearly independent (and remove any linear dependence). We can do this by XORing a given row (vector) with all other rows below it that has a 1 in the desired place spot. Once it is in row-echelon form, we are done.

Now, we can derive the perpendicular vector by working from the last row (vector). At each step, we can derive one more bit of the vector c; using the $n-1$ linearly independent vectors, all but 1 of the coordinates of c will be determined by some row, which gives a **unique non-zero vector** c.

How long does our algorithm take to find $n-1$ linearly independent vectors? At the first step, there are $2^n - 1$ vectors perpendicular to c, and at the ith step generally, there are $2^{n-i} - 1$ vectors that are linearly independent from the first $i-1$. Each probability of finding a linearly dependent vector is the reciprocal of that value, and each is less than $\frac{1}{2}$, so the expected total number of steps is $O(2n)$. Since each step takes time $O(n)$, the total runtime is $O(n^2)$, which is much less than the exponential classical algorithm.

20 October 25th, 2021

20.1 Fourier series

Consider a periodic function with period 2π. You can write it as

$$c + \sum_{j=1}^{\infty} \alpha_j \cos(jx) + \sum_{k=1}^{\infty} \beta_k \sin(kx) \tag{142}$$

This is good for analyzing periodicity. Simon's problem has periodicity! $f(x) = f(x \oplus c)$ is periodic with period mod $\mathbb{Z}_2^n$. The problem of factoring is *very* related to periodicity.

In real life, you never have a continuous function but a bunch of samples from a continuous function. Can use the discrete Fourier transform (DFT) for this.

Definition 20.1 (Discrete Fourier transform)**.** Suppose you have a m samples spaced $1/a$ apart from each other. The DFT gives you m values in k space for your Fourier components. Label the points $a_0, a_1, \dots, a_{n-1}$. These get taken to $b_0, b_1, \dots, b_{n-1}$, by

$$b_k = \sum_{j=0}^{n-1} e^{-2\pi ijk/n} a_j. \tag{143}$$

The inverse DFT is

$$a_j = \frac{1}{n} \sum_{k=0}^{n-1} e^{2\pi ijk/n} b_k. \tag{144}$$

Definition 20.2 (Quantum Fourier transform)**.** The *quantum Fourier transform* is a unitary transformation that satisfies

$$|j\rangle \to \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} e^{2\pi ijk/n} |k\rangle, \qquad |k\rangle \to \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} e^{-2\pi ijk/n} |j\rangle \tag{145}$$

We can write the matrix for the quantum Fourier transform. Let $\omega = e^{2\pi i/n}$. We have

$$\frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \omega^{n-2} \\ 1 & \omega^3 & \omega^6 & \omega^{n-3} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{n-1} & \omega^{n-2} & \omega \end{bmatrix}. \tag{146}$$

We note that this is unitary. The inverse quantum Fourier transform is given by

$$\frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \omega^{-(n-2)} \\ 1 & \omega^{-3} & \omega^{-6} & \omega^{-(n-3)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-(n-2)} & \omega^{-1} \end{bmatrix} \tag{147}$$

which is just the conjugate transpose of the matrix for the quantum Fourier transform.

20.2 Implementing the QFT

We want to decompose this into one and two-qubit gates. We will assume $n = 2^L$ so we have L qubits. We have

$$\sum_{n=0}^{2^L-1} e^{2\pi i jk/2^L} \tag{148}$$

and we will represent $j = j_{L-1}j_{L-1}\dots j_0$ and $k = k_{L-1}k_{L-2}\dots k_0$ as binary strings and

$$j = \sum_{s=0}^{L-1} j_s 2^s, \qquad k = \sum_{t=0}^{L-1} k_t 2^t \tag{149}$$

We can substitute the binary expansions of j, k into the quantum Fourier transform and we have

$$\begin{aligned}|j_{L-1}\rangle|j_{L-2}\rangle\dots|j_0\rangle \to \frac{1}{2^{L/2}} &\sum_{k_{L-1}\dots k_0=0}^{2^L-1} e^{2\pi i \sum_{s=0}^{L-1}\sum_{t=0}^{L-1} 2^{s+t-L} j_s k_t} |k_{L-1}\rangle|k_{L-2}\rangle\dots|k_0\rangle \\ &= \frac{1}{2^{L/2}} \sum_{k_{L-1}\dots k_0=0}^{2^L-1} \prod_{s=0}^{L-1}\prod_{t=0}^{L-1} e^{2\pi i 2^{s+t-L} j_s k_t} |k_{L-1}\rangle|k_{L-2}\rangle\dots|k_0\rangle. \end{aligned} \tag{150}$$

We can show how we turn $|j_{L-1}\rangle\dots|j_0\rangle \to |k_{L-1}\rangle\dots|k_0\rangle$ while applying relevant phases.

Case 1: $s + t = L - 1$. Then

$$e^{2\pi i 2^{s+t-L} j_s k_t} = e^{\pi i j_s k_t} = (-1)^{j_s k_t}. \tag{151}$$

Case 2: $s + t < L - 1$. Then

$$e^{2\pi i 2^{s+t-L} j_s k_t} = 1. \tag{152}$$

In the first case, we recall the Hadamard gate

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \frac{1}{\sqrt{2}}\sum_{j=0}^{1}\sum_{k=0}^{1}(-1)^{jk}|k\rangle\langle j| \tag{153}$$

which applies a -1 phase from $|1\rangle \to |1\rangle$ and a $+1$ phase otherwise. Then we should apply a Hadamard gate of the form

$$\frac{1}{\sqrt{2}}\sum_{j_s=0}^{1}\sum_{k_t=0}^{1}(-1)^{j_s k_t}|k_t\rangle\langle j_s| \tag{154}$$

and $t = L - 1 - s$ turns $|j_s\rangle \to |k_t\rangle$ and applies the correct phase. Then it is clear that we can turn all qubits $|j_s\rangle \to |k_t\rangle$ by applying L Hadamards one at a time.

For the second case, we need to show that this can apply other phases of the form

$$e^{2\pi i 2^{-\ell} j_s k_t} \tag{155}$$

for some $\ell = L - s - t \geq 2$. We can implement these as controlled R_ℓ gates with

$$R_\ell = \begin{bmatrix}1 & 0\\ 0 & e^{2\pi i 2^{-\ell}}\end{bmatrix}. \tag{156}$$

We note that

$$C - R_\ell = \begin{bmatrix}1 & & & \\ & 1 & & \\ & & 1 & \\ & & & e^{2\pi i/2^\ell}\end{bmatrix} = |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 01| + e^{2\pi i/2^\ell}|11\rangle\langle 11| \tag{157}$$

so this applies the rotation $\iff$ $|j_s\rangle = |k_t\rangle = |1\rangle$, which in fact they are.

21 October 29th, 2021

21.1 More quantum Fourier transforms

This is a continued discussion of the last lecture. Note that the circuit for the quantum Fourier transform for $L = 4$ is given by

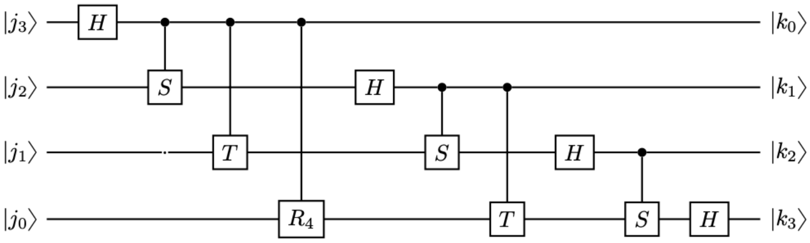


Figure 7: Quantum Fourier transform circuit

where

$$S = R_2 = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \qquad T = R_3 = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1+i}{\sqrt{2}} \end{bmatrix}, \qquad R_4 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{bmatrix} \tag{158}$$

The general circuit is given on page 219 of the textbook.

We can do the multiplication for $L = 2$ explicitly. The circuit is

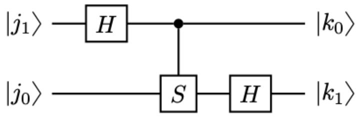


Figure 8: QFT circuit for $L = 2$

where we can compute $(I \otimes H)(C - S)(H \otimes I)$ and do a SWAP gate to get the qubits in the correct order.

22 November 1st, 2021

Today, we will cover an application of the quantum Fourier transform.

22.1 Phase estimation

Suppose we have a unitary U and an eigenvector $|v_\theta\rangle$ where

$$U|v_\theta\rangle = e^{i\theta}|v_\theta\rangle. \tag{159}$$

We seek to find an approximate θ. The phase estimation algorithm does not give accurate estimates of θ unless we take high powers of U. Let us assume that we can get compute U^{2^ℓ} in polynomial time in ℓ.

Suppose the unitary takes a number $s \mod p$ and multiplies it by another number $g \mod p$ and

$$U|s \mod p\rangle = |gs \mod p\rangle. \tag{160}$$

Then

$$U^k|s \mod p\rangle = |g^k s \mod p\rangle. \tag{161}$$

If we know $g^k \mod p, U^k$ is no harder to implement than U. We can find $g^{2^\ell} \mod p$ efficiently by repeated squaring.

The basic idea for the algorithm is to create the state

$$\frac{1}{2^{L/2}} \sum_{k=0}^{2^L-1} e^{ik\theta}|k\rangle. \tag{162}$$

This *looks like* the quantum Fourier transform. Recall QFT maps

$$|j\rangle \to \frac{1}{2^{L/2}} \sum_{k=0}^{2^L-1} e^{2\pi ijk/2^L}|k\rangle. \tag{163}$$

Note that $\{|j_i\rangle\}$ form a basis and so the right-hand side also forms a basis.

We note that if $\theta = 2\pi m/2^L$ for some $m \in \mathbb{Z}$, then it is one of these basis states. Then to identify θ, all we need to do is to measure in that (right-hand side) basis. We can achieve this by taking the *inverse quantum Fourier transform*, which takes

$$\frac{1}{2^{L/2}} \sum_{k=0}^{2^L-1} e^{2\pi ijk/2^L}|k\rangle \to |j\rangle \tag{164}$$

and then we measure in the standard basis. If we measure $j \implies \theta = 2\pi j/2^L$.

We now need to show how to create the desired state and that this algorithm works for $\theta \neq 2\pi m/2^L$ for some $m \in \mathbb{Z}$.

22.1.1 Creating the desired state

Let us assume that we can implement the transformation U^{2^k} efficiently in polynomial time. Consider the quantum circuit

If the input is $k = k_{L-1}k_{L-2}\ldots k_0$, the circuit applies $U^{k_0+2k_1+4k_2+\cdots+2^{L-1}k_{L-1}}$ to $|v_\theta\rangle$, which is the same as $U^k|v_\theta\rangle = e^{ik\theta}$, which gives output $|k\rangle e^{i\theta k}$.

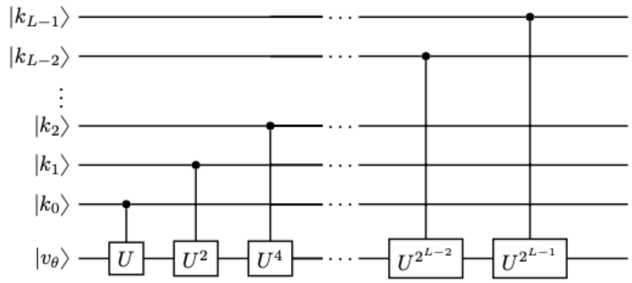


Figure 9: Circuit for creating desired state

Note that in order to get our desired state, we input

$$\frac{1}{2^{L/2}}\sum_{k=0}^{2^L-1}|k\rangle|v_\theta\rangle \tag{165}$$

into the circuit. We can do this by putting a $|+\rangle$ state into each of the first L quantum wires.

Now we observe what happens if we take the above state and apply the inverse Fourier transform, following our algorithm. For $\theta \neq 2\pi m/2^L$, the inverse Fourier transform is

$$|k\rangle \to \frac{1}{2^{L/2}}\sum_{j=0}^{2^L-1}e^{-2\pi ijk/L}|j\rangle. \tag{166}$$

Plugging this into (164), we obtain

$$\frac{1}{2^L}\sum_{k=0}^{2^L-1}e^{ik\theta}\sum_{j=0}^{2^L-1}e^{-2\pi ijk/2^L}|k\rangle = \frac{1}{2^L}\sum_{j=0}^{2^L-1}|j\rangle\left(\sum_{k=0}^{2^L-1}e^{ik\left(\theta-2\pi j/2^L\right)}\right). \tag{167}$$

The probability of seeing $|j\rangle$ is

$$\left|\frac{1}{2^L}\sum_{k=0}^{2^L-1}e^{ik\left(\theta-2\pi j/2^L\right)}\right|^2 = \frac{1}{4^L}\left|\frac{1-e^{i\left(2^L\theta-2\pi j\right)}}{1-e^{i(\theta-2\pi j/2^L)}}\right|^2. \tag{168}$$

Note that the value of j we obtain gives us a good approximation of θ. We can bound the numerator by 2 and approximate the denominator by $i(\theta-2\pi j/2^L)$. Let j' be the value that gives us the right value of θ. That is, $2\pi j'/2^L=\theta$. Then the probability seeing some $j > j'+\alpha$ or $j < j'-\alpha$ is around

$$\frac{1}{4^L}\left|\frac{2}{2\pi(j'-j)/2^L}\right|^2 = \frac{1}{|\pi\alpha|^2}, \tag{169}$$

which shows that j' is tightly concentrated around $2^L\theta/(2\pi)$ and we can get a good estimate of the phase θ.

23 November 3rd, 2021

Today, we will talk about order-finding and factoring.

23.1 Background

For factoring problems, given N, we want to find $P, Q < N : PQ = N$. To break RSA, we need to factor when P, Q are prime.

For the quadratic sieve (QS) algorithm, which is the second fastest classical factoring method, and for quantum factoring, we want to find two integers where $x^2 \cong y^2 \mod N$ but $x \ncong y \mod N$. Note that by factoring the first condition, we have

$$(x-y)(x+y) \cong 0 \mod N \tag{170}$$

and we have $x-y, x+y$ our desired factors. This is a common technique used by many factoring algorithms.

Example 23.1 (Factoring)**.** Consider

$$32^2 \cong 1024 \cong 49 \mod 65 \implies 32^2 - 7^2 \cong 0 \mod 65 \implies (32+7)(32-7) \mod 65 \implies 39 \times 25 \mod 65 \tag{171}$$

where the first factor contains 13 and the second factor contains 5.

The quadratic sieve works kind of like this, but we will use period finding to find x^2, y^2.

23.2 Shor's algorithm

Given a function $f : \mathbb{Z} \to \mathbb{Z}, f(x) = f(x-c)$, we want to find c.

The function we will use is

$$f(x) = a^x \mod N \tag{172}$$

Example 23.2. Suppose we want to factor $N = 119$ and we choose $x = 2$. We can do some calculations and note that $f(24)$ is when it starts repeating.

Because there are finite number of residues modulo N, the sequence is guaranteed to start repeating. We will get $a^k \equiv a^{k+r} \mod N$. If $(a, N) = 1$, we obtain $a^r \equiv 1 \mod N$ and r is the period of the sequence.

Note that if r even, this gives

$$\left(a^{r/2}\right)^2 \equiv 1^2 \mod N. \tag{173}$$

If $a^{r/2} \equiv -1 \mod N$, this does not work and we need to try again. Note that r odd also does not work.

Theorem 23.3. *Choose a random a. The probability that this method works is weakly greater than $\frac{1}{2}$. That is, repeating this process for different values of a will eventually give a factor in polynomial number of trials.*

To find the period of a sequence, we can consider the unitary transformation that takes us from one element of the sequence to the next. In this case, we have

$$U_a|y \mod N\rangle = |ay \mod N\rangle. \tag{174}$$

This is reversible since $(a, N) = 1$. It is possible to find $a^{-1} \mod N$ in polynomial time in the length of N using the Euclidean algorithm.

Recall a theorem from earlier in the course:

Theorem 23.4. *If there exists a reversible circuit that takes*

$$|x\rangle|0\rangle \to |x\rangle|f(x)\rangle, \qquad |x\rangle|0\rangle \to |x\rangle|f^{-1}(x)\rangle \tag{175}$$

then there exists a reversible circuit $|x\rangle \to |f(x)\rangle$.

Then it there exists a reversible circuit for U_a. We can apply the phase estimation algorithm by performing $U_a^{2^k}$. This is easy because $U_a^{2^k} = U_{a^{2^k} \mod N}$. We can find $a^{2^k} \mod N$ using repeated squaring and then implement $U_{a^{2^k}}$ and can do phase estimation.

We seek to find the eigenvectors and eigenvalues of U_a. Consider the quantum state

$$|\zeta_k\rangle = \frac{1}{\sqrt{r}}\left(|1\rangle + 2^{i\pi k/r}|a\rangle + e^{4\pi ik/r}|a^2\rangle + \cdots + e^{2\pi(r-2)k/r}|a^{r-2}\rangle + e^{2\pi(r-1)k/r}|a^{r-1}\rangle\right). \tag{176}$$

When we apply U_a, we obtain

$$U_a|\zeta_k\rangle = \frac{1}{\sqrt{r}}\left(|a\rangle + 2^{i\pi k/r}|a^2\rangle + e^{4\pi ik/r}|a^3\rangle + \cdots + e^{2\pi(r-2)k/r}|a^{r-1}\rangle + e^{2\pi(r-1)k/r}|a^r\rangle\right). \tag{177}$$

Note that $|a^r\rangle \equiv |1\rangle$ and so $U_a|\zeta_k\rangle = e^{-2\pi k/r}|\zeta_k\rangle$ and we have found r eigenvectors of U_a. If we find one of these eigenvectors, we can use phase estimation to approximate the eigenvalue, which gives us an approximation of $\theta = k/r$.

Realistically, we do not have one of these eigenvectors. We will do phase estimation anyway. If we want to factor an L-bit number, we will use the phase estimation and quantum Fourier transform with $2L$ qubits. This will measure the eigenvectors and eigenvalue and we will get a state close to $|\zeta_k\rangle$ and phase estimation will give us a good approximation of k/r in the form $d/2^{2L} \approx k/r$ for some d. We can find r from this using some fancy number theory method called *continued fractions.*

We can consider starting in the algorithm in state $|1\rangle$. We have

$$|1\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|\zeta_k\rangle \tag{178}$$

so when we apply phase estimation, we will get a random $|\zeta_k\rangle$ and eigenvalue $e^{-2\pi ik/r}$.

Next lecture, we will talk about the number theory justification that makes this algorithm possible!

24 November 5th, 2021

Announcements

- Midterms are graded!

Today, we will do the number theory associated with factoring. We will provide a gist of the theory that will give us the general idea of the proofs. To go in-depth, we would need another few lectures, which we unfortunately do not have.

24.1 Factoring background

The quadratic sieve is a classical algorithm that takes time $e^{c\sqrt{\log N}\sqrt{\log\log N}}$. The quantum algorithm (Shor's algorithm) takes quantum time $O(\log N^3)$.

Both if these algorithms use the idea that if $N = PQ$ and $(A-B)(A+B) \equiv 0 \mod N$ and $A \equiv \pm 1 \mod N$, we can get one factor from $A + B$ and the other factor from $A - B$.

The best algorithm that does not use this scheme is the number field sieve, which takes time $e^{c\sqrt[3]{\log N}(\log\log N)^{2/3}}$.

24.2 Factoring details

The factoring algorithm gives us $A^2 \equiv B^2 \mod n$. From this we can get the factors $A+B, A-B$. However, these are not necessarily true factors (they may be multiples of factors). We can see this because $(A + B)(A - B) = mN$ for some integer m. Instead, $\gcd(A, N)$ and $\gcd(B, N)$ will give us factors. These are easily computable with the Euclidean algorithm.

24.2.1 Chinese remainder theorem

Another part is picking a for our period finding step. We can show that the probability of a random a working is $\geq 1/2$.

Theorem 24.1 (Chinese remainder theorem)**.** *There is a one-to-one correspondence between integers* $\mod N$ *and pairs of integers* $\mod (P, Q)$

Example 24.2 (CRT)**.** Suppose $N = 15, P = 3, Q = 5$. Then $7 \equiv (1, 2) \mod (P, Q)$.

24.3 Continued fractions

Continued fractions are a way in which we can represent arbitrary rational numbers.

Example 24.3.

$$\frac{12}{41} = \frac{1}{3 + \frac{1}{2+\frac{1}{2+\frac{1}{2}}}} \tag{179}$$

Theorem 24.4. *The convergents of a continued fraction are the closest approximations to the fraction.*

In the $12/41$ example, the first convergent is $1/3$. The second convergent is $1/\left(3 + \frac{1}{2}\right) = 2/7$.

In the last step of the factoring algorithm, we found the phase of p/r. The phase approximation algorithm gave us

$$\frac{p}{r} \approx \frac{d}{2^{2L}}. \tag{180}$$

We can then use continued fractions to get from $d/2^{2L}$ to p/r.

Theorem 24.5. *If $2^{2L} > 2r^2$, then one of the convergents of $d/2^{2L}$ is p/r.*

Therefore, we can guess values of r from enumerating the convergents. One way to quickly guess r is to look at when the continued fraction values jump. This indicates a point where the change in the fraction from each additional term of the continued fraction is very small, so you've probably reached the actual value of p/r at the last term right before that jump.

25 November 8th, 2021

Announcements

- The mean on the midterm was a 55/60 with standard deviation 6
- About one quarter of the class got a perfect score

Today, we will discuss discrete logs.

25.1 Discrete log problem

We note that there are cryptography systems based on discrete log that is much easier than RSA.

Let p prime and g be a generator for the multiplicative group $\mod p, (\mathbb{Z}/p\mathbb{Z})^\times$. Let $h \in (\mathbb{Z}/p/\mathbb{Z})^\times$ We seek to find $x : g^x \equiv h \mod p$.

Definition 25.1 (Generators)**.** g is a *generator* if $\forall h \in [1, p]\exists x : g^x \equiv h \mod p$.

Example 25.2. Consider $p = 31$. We can check that 3 is a generator of the multiplicative group.

25.1.1 Diffie-Hellman key exchange

Supose Alice and Bob want to create a secret that they know but no one else knows. Their only means of communication is over a telephone wire that is being tapped.

One way to encrypt their message is to agree on a prime and a generator p, g. Alice chooses random $x < p$ and Bob chooses a random number $y < p$. Alice sends $g^x \mod p$ and Bob sends Alice $g^y \mod p$. Note that the secret is $g^{xy} \mod p$.

If we know this g^{xy} and use it as a secret key, we can use this as a secret key for a *symmetric* cryptography system, and then we can communicate securely.

The problem. If we know $p, g, g^x, g^y \mod p$, can we find $g^{xy} \mod p$?

Note. If we know discrete log, we can break Diffie-Hellman!

We can find $g^x \mod p$ for large x. We can write x in binary and find

$$g, g^2, g^4, \cdots, g^{2^L} \mod p. \tag{181}$$

We can multiply the associated powers of g in binary to find g^x. Consider $x = 101_2$. Then we have

$$g^x \equiv 1g^{2^0} + 0g^{2^1} + 1g^{2^2} \mod p. \tag{182}$$

There are classical algorithms for discrete log, but we will consider the quantum algorithm that utilizes period-finding.

25.1.2 Quantum algorithm

We will use period-finding. Let U_g be our familiar unitary

$$U_g : |\ell \mod p\rangle \to |\ell g \mod p\rangle. \tag{183}$$

The eigenvectors for this map are the same familiar ones!

$$|\zeta_k\rangle = \frac{1}{\sqrt{p-1}}\left(|1\rangle + e^{2k\pi i/(p-1)}|g\rangle + e^{4k\pi i/(p-1)}|g^2\rangle + \cdots + e^{(p-2)k\pi i/(p-1)}|g^{p-2}\rangle\right) \tag{184}$$

where we have $g^{p-1} \equiv 1 \mod p$.

Theorem 25.3 (Fermat's little theorem)**.** *For p prime and $(a, p) = 1$, we have*

$$a^{p-1} \equiv 1 \mod p. \tag{185}$$

The corresponding eigenvalues are $e^{-2k\pi i/(p-1)}$. Like last time, we have

$$|1\rangle = \frac{1}{\sqrt{p-1}} \sum_{k=0}^{p-2} |\zeta_k\rangle. \tag{186}$$

Then we can apply the phase estimation algorithm to get $\frac{1}{d^{2^L}}$ and we can round it off to $\frac{k}{p-1}$. We require $2^L > 2p$.

Note. We can get $U_g^{2^m}$ by multiplying instead by $g^{2^m}, m \in \mathbb{Z}$.

We can consider

$$U_h|\ell \mod p\rangle = |h\ell \mod p\rangle \tag{187}$$

where $h = g^x \mod p$.

Claim. $|\zeta_k\rangle$ are eigenvectors of U_h.

Proof. Note that $|\zeta_k\rangle = \frac{1}{\sqrt{p-1}} \sum_\ell e^{2k\ell\pi i/(p-1)} |g^\ell\rangle$ and

$$\begin{aligned} U_h|\zeta_k\rangle = U_h \frac{1}{\sqrt{p-1}} \sum_\ell e^{2k\ell\pi i/(p-1)} |g^\ell\rangle &= \frac{1}{\sqrt{p-1}} \sum_\ell e^{2k\ell\pi i/(p-1)} |g^\ell g^x\rangle \\ &= \frac{1}{\sqrt{p-1}} e^{-2\pi i k x/(p-1)} \sum_\ell e^{2\pi i k \ell/(p-1)} e^{2\pi i k x(p-1)} |g^{\ell+x}\rangle \\ &= e^{-2\pi i k x/(p-1)} \underbrace{\frac{1}{\sqrt{p-1}} \sum_\ell e^{2\pi i k(\ell+x)/(p-1)} |g^{\ell+x}\rangle}_{|\zeta_k\rangle} \end{aligned} \tag{188}$$

with eigenvalue $e^{-2\pi i k x/(p-1)}$. □

Then note that doing phase estimation twice gives **both** $kx/(p-1)$ **and** $k/(p-1)$ where the estimates come from U_g, U_h, respectively.

We can then extract x by multiplying kx by the inverse of $k \mod p-1$. Note that we can get kx and x because we already know $p - 1$.

Example 25.4. Let $p = 31$ and suppose we get $\frac{7}{30}, \frac{12}{30} = \frac{2}{5}$. With $g = 3, h = 16$. Then we can find k by multiplying by $7^{-1} \mod 30 = 13$. Note $7 \times 13 = 91 \equiv 1 \mod 30$. Doing this multiplication, we obtain

$$x = 13 \times 12 \mod 30 = 6. \tag{189}$$

Example 25.5. But if $k = 5, kx = 0 \mod 30$, we cannot find x! We require then, that $(k, p - 1) = 1$.

Theorem 25.6. *We find $(k, p - 1)$ relatively prime with probability*

$$\frac{\varphi(p)}{p} \leq \frac{1}{\log\log p} \tag{190}$$

where φ is Euler's totient function.

Note. We can find inverses by using the extended Euclidean algorithm.

The complete algorithm is as follows. Given prime p and generator g, we consider U_g and U_h where $g^x \equiv h \mod p$. We can apply the phase estimation algorithm on both U_g, U_h to obtain estimates

$$\frac{kx}{p-1}, \qquad \frac{k}{p-1} \tag{191}$$

from U_g, U_h, respectively. We can then extract x by multiplying $k^{-1} \times kx \mod p-1$ and we have our discrete log.

26 November 10th, 2021

Today, we will discuss Gover's search algorithm. This algorithm is used to search the solution space of some equation.

26.0.1 The problem

Consider the question of finding three cubes that add to 42. This problem was open for decades. The solution was discovered last year by Andrew Sutherland (at MIT) and Andrew Booker that did **not** use an exhaustive search.

A brute-force search would require checking 10^{34} pairs of numbers. We do this by asking if

$$\sqrt[3]{a^3 + b^3} \in \mathbb{Z}. \tag{192}$$

A quantum computer can do this checking with $\sim 10^{17}$ evaluations of this. This is because quantum computers somehow use diffusion to check all these pairs.

26.1 Grover's algorithm

Consider a solution space of N things and one of them is marked. We consider a Grover oracle O_p

$$O_p|x\rangle = \begin{cases} +|x\rangle, & x \text{ not marked} \\ -|x\rangle, & x \text{ marked} \end{cases} \tag{193}$$

We have the algorithm as follows. We start in the superposition

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle. \tag{194}$$

We then apply the Grover oracle O_p, apply a Hadamard transform $H^{\otimes N}$, apply $2|0\rangle\langle 0| - I$, and apply another Hadamard transform. We repeat this procedure until we are *done*. Note that checking when we are done collapses the state, so we just do this *long enough* until we think it is finished. Then we measure and get the marked item with high probability.

Note. We have

$$(2|0\rangle\langle 0| - I)\,|x\rangle = \begin{cases} +|x\rangle, & |x\rangle = |0\rangle \\ -|x\rangle, & \text{otherwise} \end{cases} \tag{195}$$

which is fairly easy to implement. We note that

$$2H^{\otimes N}|0\rangle\langle 0|H^{\otimes N} - I = 2|\psi\rangle\langle\psi| - I, \qquad |\psi\rangle = \frac{1}{2^{L/2}} \sum_{x=0}^{2^L} |x\rangle. \tag{196}$$

Then we can actually start in superposition

$$|\psi\rangle = \frac{1}{2^{L/2}} \sum_{x=0}^{2^L} |x\rangle \tag{197}$$

which is approximately the same as starting in state space size N.

Using Grover's algorithm can reduce the time in many other classical search algorithms.

26.1.1 Intuition

Note that $2|\psi\rangle\langle\psi| - I$ reflects all amplitudes around their average value. We can see that $\frac{1}{N}\sum_{i=0}^{N-1}\langle i|$ computes the average value of the amplitude and $\sum_{i=0}^{N-1}|i\rangle$ applies this value to every state.

When we apply O_p, we reflect each of the amplitudes around 0 and the other three reflect them around the average amplitude. Intuitively, the marked states start with amplitude $\frac{1}{\sqrt{N}}$. The first reflection takes the marked states to $-\frac{1}{\sqrt{N}}$. The average will still be almost exactly $\frac{1}{\sqrt{N}}$ so the second reflection around the average takes the marked states to $\frac{3}{\sqrt{N}}$. We note that after the kth Grover iteration, the marked states have amplitude $\frac{2k+1}{\sqrt{N}} \implies O(\sqrt{N})$ steps to have probability nearly 1 of finding the marked states.

More formally let there be N states, M of which are marked. Let $|\alpha\rangle$ be the equal superposition of all unmarked states and $|\beta\rangle$ be the equal superposition of all marked states

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}}\sum_{x \text{ unmarked}}|x\rangle, \qquad |\beta\rangle = \frac{1}{\sqrt{M}}\sum_{x \text{ marked}}|x\rangle \tag{198}$$

and

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle. \tag{199}$$

The Grover iteration keeps the computational state in a subspace generated by $|\alpha\rangle, |\beta\rangle$. We can imagine $\psi \in \text{span}\{|\alpha\rangle, |\beta\rangle\}$ and let $\theta/2$ be the angle between $|\alpha\rangle$ and $|\psi\rangle$. Using trigonometry, we have

$$\sin\theta/2 = \sqrt{M/N} \implies \theta \approx 2\sqrt{M/N} \text{ if } M \ll N. \tag{200}$$

Note then, that

$$O_p|\alpha\rangle = |\alpha\rangle, \qquad O_p|\beta\rangle = -|\beta\rangle \tag{201}$$

and

$$(2|\psi\rangle\langle\psi| - I)\,|\psi\rangle = 2|\psi\rangle - |\psi\rangle = |\psi\rangle, \qquad (2|\psi\rangle\langle\psi| - I)\,|\bar{\psi}\rangle = -|\bar{\psi}\rangle \tag{202}$$

where $|\bar{\psi}\rangle$ is the state orthogonal to $|\psi\rangle$. Then this is a reflection across $|\psi\rangle$! The product of these reflections in the plane is a rotation by $+\theta \approx 2\sqrt{M/N}$ degrees. We can repeat this again and get a rotation by $+2\theta$.

We want this state to be rotated to the $|\beta\rangle$ axis, which is angle of approximately $\pi/2$ from the original state. This takes

$$\frac{\pi/2}{2\sqrt{M/N}} \approx \frac{\pi}{4}\sqrt{\frac{N}{M}} \tag{203}$$

iterations.

Note that if we do not know M, we can measure in the $\{|\alpha\rangle, |\beta\rangle\}$ basis and obtain $|\beta\rangle$ with probability $\frac{1}{2}$. If we run Grover's algorithm for a random number of iterations, if iterations are sufficiently large, we have a $\frac{1}{2}$ chance of getting a marked state after measurement.

Then another method of getting a marked state in $O(\sqrt{M/N})$ time is to run Grover's for a random number N between 5 and 10 iterations, then a random number between 10 and 20 iterations, then a random number between 20 and 40 iterations, then a random number between 40 and 80, and so on. We are likely to get a marked state after roughly $\sqrt{N/M}$ iterations.

27 November 12th, 2021

Today, we will discuss the lower bound on the quantum search algorithm (Grover's algorithm).

We recall Gover's algorithm. We want to know how many times the algorithm needs to call the Grover oracle. The original Grover's algorithm can return marked $|x\rangle$ in $O(\sqrt{N})$ steps.

The idea is to start start in state $|\psi\rangle$ and assume another program computes

$$|\psi_k^x\rangle = U_k O_x U_{k-1} O_x \cdots U_1 O_x |\psi\rangle, \qquad |\psi_k\rangle = U_k U_{k-1} \cdots U_1 |\psi\rangle \tag{204}$$

where O_x is Grover's oracle. We can define, where $|\psi_k^x\rangle$ is the final state after the program with O_x ($|\psi_k\rangle$ corresponds to $O_x = I$, no marked states),

$$D_k = \sum_x \left| |\psi_k^x\rangle - |\psi_k\rangle \right|^2. \tag{205}$$

At the end, we have $|\psi_k^x\rangle = |x\rangle$, so intuitively, at the end of the program, $D_k \approx N$. We will also show that D_k cannot grow faster than $O(k^2)$. We also note that D_k is intuitively minimized when $|\psi_k\rangle = \left(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \cdots, \frac{1}{\sqrt{N}}\right)$.

There is a theorem that says we can assume all measurements at the end of a quantum algorithm.

Proof. We will proceed by induction. We get D_{k+1} in terms of D_k. We have

$$D_{k+1} = \sum_x |U_{k+1} O_x |\psi_k^x\rangle - U_{k+1} |\psi_k\rangle|^2 = \sum_x |O_x |\psi_k^x\rangle - |\psi_k\rangle|^2 = \sum_x |O_x |\psi_k^x\rangle - O_x |\psi_k\rangle + O_x |\psi_k\rangle - I |\psi_k\rangle|^2. \tag{206}$$

We will use the fact that

$$\left| |\phi\rangle - |\chi\rangle \right|^2 \leq |\langle \phi | \phi \rangle - \langle \chi | \chi \rangle| + 2 \left| \langle \phi | \chi \rangle \right| \leq |\langle \phi | \phi \rangle + \langle \chi | \chi \rangle| + 2 ||\phi\rangle|||\chi\rangle|. \tag{207}$$

Then

$$D_{k+1} \leq \sum_x |O_x |\psi_k^x\rangle - O_x |\psi_k\rangle|^2 + \sum_x |(O_x - I)|\psi_k\rangle|^2 + 2 \sum_x |O_x (|\psi_k^x\rangle |\psi_k\rangle)| \, |(O_x - I)|\psi_k\rangle|. \tag{208}$$

The first term is

$$\sum_x \left| |\psi_k^x\rangle - |\psi_k\rangle \right|^2 = D_k. \tag{209}$$

The second term is

$$\sum_x |(O_x - I)|\psi_k\rangle|^2. \tag{210}$$

We note

$$O_x - I = 2|x\rangle\langle x| \implies \sum_x |(O_x - I)\,|\psi_k\rangle|^2 = 4 \sum_x ||x\rangle \, \langle x|\psi_k\rangle|^2 = 4 \sum_x |\langle x|\psi_k\rangle|^2 = 4. \tag{211}$$

The third term is the tricky one. We will use the Cauchy-Schwarz inequality $|\boldsymbol{v} \cdot \boldsymbol{w}| \leq \|\boldsymbol{v}\| \|\boldsymbol{w}\|$. We have

$$2 \sum_x \left| |\psi_k^x\rangle - |\psi_k\rangle \right| |(O_x - I)\,|\psi_k\rangle| \leq 2 \sqrt{\sum_x \left| |\psi_k^x\rangle - |\psi_k\rangle \right|^2} \sqrt{\sum_x |(O_x - I)\,|\psi_k\rangle|^2} \leq 2\sqrt{D_k}\sqrt{4} = 4\sqrt{D_k}. \tag{212}$$

We can collect terms and obtain

$$D_{k+1} \leq D_k + 4\sqrt{D_k} + 4 \implies \left(\sqrt{D_{k+1}}\right)^2 \leq \left(\sqrt{D_k} + 2\right)^2 \implies \sqrt{D_{k+1}} \leq \sqrt{D_k} + 2. \tag{213}$$

Then if we consider $D_0 = 0$ because $|\psi_0^x\rangle = |\psi_0\rangle$ and

$$\sqrt{D_{k+1}} \leq 2(k+1) \implies D_{k+1} \leq 4(k+1)^2. \tag{214}$$

Then

$$\begin{aligned} D_{\text{end}} = \sum_x \left||\psi_{\text{end}}^x\rangle - |\psi_{\text{end}}\rangle\right|^2 = \sum_x \left||x\rangle - |\psi_{\text{end}}\rangle\right|^2 &= \sum_x \left(\langle x| - \langle\psi_{\text{end}}|\right)\left(|x\rangle - |\psi_{\text{end}}\rangle\right) \\ &= \sum_x |2 - 2\langle\psi_{\text{end}}|x\rangle| \geq \sum_x |2| - 2\sum_x |\langle\psi_{\text{end}}|x\rangle| \end{aligned} \tag{215}$$

We will use Cauchy-Schwarz again! Consider

$$\sum_x |\langle\psi_{\text{end}}|x\rangle| \leq \sqrt{\sum |\langle\psi_{\text{end}}|x\rangle|^2}\sqrt{\sum |1|^2} = 1 \cdot \sqrt{N} \tag{216}$$

and then

$$D_{\text{end}} \geq 2N - 2\sqrt{N}. \tag{217}$$

Then we note that $2N \leq D_{\text{end}} \leq 4k^2$ where k is the number of oracle calls $\implies k \geq \frac{1}{\sqrt{2}}\sqrt{N}$. Then this is the lower bound for the number of searches using Grover's algorithm. □

27.1 Quantum error correction preview

Suppose we have a qubit $|\psi\rangle$ that cannot be cloned. We can take the qubit and encode it in five single-qubit pieces $|\phi_i\rangle, i \in [1, 5]$, which are all entangled with each other.

We then send the $|\phi_i\rangle$ through a noisy channel. An adversary then comes and makes some arbitrary error (deletion, measurement, applying a unitary) one of these. The idea is that the receiver can take the four that are *not deleted* and recover $|\psi\rangle$ perfectly.

Philosophers may wonder where the quantum information is in deletion step. We note that the information is not encoded in the redundancy of states, but it is also not encoded in any singular $|\phi_i\rangle$.

This is the quantum analog of classical error-correcting code.

28 November 15th, 2021

28.1 Background

in 1948, Claude Shannon published a paper that started information theory. He showed that noisy communication channels have a *capacity* and derived *Shannon's formula* for the capacity of a channel. He showed that an algorithm existed that could transmit information at nearly the channel capacity, but it was exponential in time to implement.

Two years after publication, Richard Hamming discovered the first error-correcting codes. Hamming codes are *linear codes* and have nice properties.

In a bit more detail, the Hamming codes take a four-bit message, encode it into a seven-bit message in a way such that if one of the seven bits is wrong, we can recover the original four bits. The encoding is done by multiplying a message m by a generator matrix G where

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{218}$$

Decoding is done with matrix H with

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{219}$$

where $GH = 0$. If there is a one-bit error in transmission r, we have

$$r = mG + e \implies rH = mGH + eH = eH \implies rH = eH \tag{220}$$

is independent of message and know what the error is. eH is the *syndrome* of the error. Computing e from eH is computationally difficult.

Before Shannon, people used repetition codes where we repeat the bit k times. The encoding matrix in general is

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{221}$$

Note. For a three-bit code, let p be the probability we have an error in one bit and $1 - p$ probability of being correct. Then there will be an error in encoding if and only if at least two encoding bits have errors. The probability is $3p^2(1-p) + p^3 \approx 3p^2$ for p small.

We will first discuss the quantum repetition code, nine-qubit codes that can correct one error, and finally discuss the quantum analog of the Hamming codes.

28.2 Quantum repetition code

It is impossible to clone a qubit, but we can consider a unitary

$$U|0\rangle|00\rangle = |000\rangle, \qquad U|1\rangle|11\rangle = |111\rangle \tag{222}$$

which is a three-qubit code that protects against bit-flip errors. The code maps

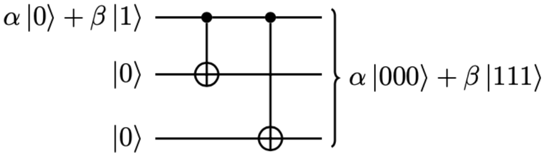
$$\alpha|0\rangle + \beta|1\rangle \to \alpha|000\rangle + \beta|111\rangle. \tag{223}$$

This code corrects one-bit flip (a σ_x error). We measure which bit is different. We do this by projecting the three qubits onto one of the following four subspaces

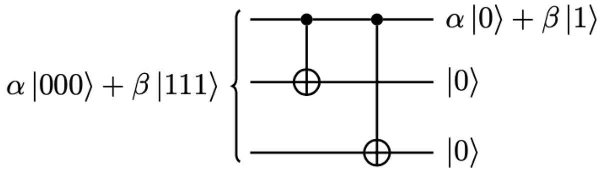
$$|000\rangle\langle 000| + |111\rangle\langle 111| \tag{224}$$
$$|100\rangle\langle 100| + |011\rangle\langle 011| \tag{225}$$
$$|010\rangle\langle 010| + |101\rangle\langle 101| \tag{226}$$
$$|001\rangle\langle 001| + |110\rangle\langle 110| \tag{227}$$

Once we measure which subspace we are in, we can correct the error. For example, if we project into the third subspace, we apply σ_x to the second qubit.

The encoding and decoding circuits are given by


$\alpha|0\rangle + \beta|1\rangle$
$|0\rangle$
$|0\rangle$
$\alpha|000\rangle + \beta|111\rangle$

(a) Encoding circuit


$\alpha|000\rangle + \beta|111\rangle$
$\alpha|0\rangle + \beta|1\rangle$
$|0\rangle$
$|0\rangle$

(b) Decoding circuit

Moreover, the error correction circuit is given by

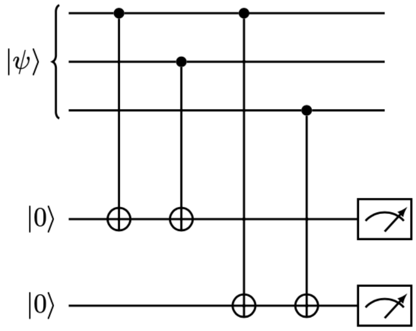


Figure 11: Quantum error correction circuit

If the two measurements (syndrome bits) are $|0\rangle$, we know that $|\psi\rangle$ was in the code subspace. If we measure $|1\rangle|0\rangle$, the second qubit is different. If we measure $|0\rangle|1\rangle$, the third qubit is different and if we measure $|1\rangle|1\rangle$, the first qubit is different. The probability of a bit-flip error on the encoded qubit is still $\sim 3p^2$ for small p.

We have a correspondence

$$|000\rangle, |111\rangle \to |0\rangle|0\rangle, \qquad |001\rangle, |110\rangle \to |0\rangle|1\rangle, \qquad |100\rangle, |011\rangle \to |1\rangle|0\rangle, \qquad |010\rangle, |101\rangle \to |1\rangle|1\rangle. \tag{228}$$

28.2.1 Phase-flip errors

We can consider phase-flip errors (a σ_z error). We can call an encoded $|0\rangle, |1\rangle$ a *logical* $|0\rangle, |1\rangle$, represented by

$$|0_L\rangle = |000\rangle, \qquad |1_L\rangle = |111\rangle \tag{229}$$

If we apply a σ_z to any three encoding qubits, we take $|0_L\rangle \to |0_L\rangle, |1_L\rangle \to -|1_L\rangle$. Then if the probability of a phase error on a single qubit is p, the probably for a phase error on an encoded qubit is $3p + p^3 \approx 3p$ for small p. Thus, this three-qubit encoding decreases the bit flip error but increases the phase flip error.

On the other hand, we recall that Hadamards take $\sigma_x \leftrightarrow \sigma_z$ and we have a code that changes the role of a bit-flip and phase-flip error. The code is

$$|+\rangle \to |+++\rangle, \qquad |-\rangle \to |---\rangle \tag{230}$$

or equivalently,

$$|0\rangle \to \frac{1}{2}\left(|000\rangle + |011\rangle + |101\rangle + |110\rangle\right), \qquad |1\rangle \to \frac{1}{2}\left(|100\rangle + |010\rangle + |001\rangle + |111\rangle\right) \tag{231}$$

Note that we have two codes and we can either protect against on bit-flip errors and make phase-flip errors more likely or the other way around. There is a way to get around this problem.

We can *concatenate* the two codes. This comes from classical coding theory. We can encode using one code and then encode using the other. The convention is to first apply the phase correction code and then the bit error correction code to each of the qubits. Note that this may be because Peter Shor did this in his first paper. It is quite arbitrary.

This works by

$$|+\rangle \to |+++\rangle \to \frac{1}{\sqrt{8}}\left(|000\rangle + |111\rangle\right)^{\otimes 3}, \qquad |-\rangle \to |---\rangle \to \frac{1}{\sqrt{8}}\left(|000\rangle - |111\rangle\right)^{\otimes 3} \tag{232}$$

Note that if we have a bit-flip error, it's corrected by the inner code. If we have a phase-flip error, the inner code turns this into a phase-flip on the logical qubits of the inner code, which gets corrected by the outer code. Then σ_x, σ_z can be corrected.

$\sigma_y = i\sigma_x\sigma_z$ errors can similarly be corrected.

Claim. The nine-qubit code given in this lecture can correct more general one-qubit errors. We will see this in the next lecture.

29 November 17th, 2021

Announcements

- Problem set 9 released this weekend due Friday after the holiday

We will continue our discussion of the nine-qubit code.

29.1 Review

Last time, we started with the three-qubit bit flip correcting code, which takes

$$|0\rangle \to |000\rangle, \qquad |1\rangle \to |111\rangle. \tag{233}$$

Note that this is not a cloning transformation. Noting that $H\sigma_x H = \sigma_z$, we get a three-qubit phase-flip correcting code

$$|+\rangle \to |+++\rangle, \qquad |-\rangle \to |---\rangle. \tag{234}$$

Written in the $\{|0\rangle, |1\rangle\}$ basis, this is

$$|0\rangle \to \frac{1}{2}\left(|000\rangle + |011\rangle + |101\rangle + |110\rangle\right), \qquad |1\rangle \to \frac{1}{2}\left(|001\rangle + |010\rangle + |100\rangle + |111\rangle\right). \tag{235}$$

A bit-flip error σ_x on any qubit results in a bit-flip on the encoded state. We call encoded $|0\rangle, |1\rangle$ *logical* $|0_L\rangle, |1_L\rangle$.

Moreover, we can correct phase-flip σ_z errors as well because

$$\begin{aligned} &|0_L\rangle, \qquad \sigma_z^{(1)}|0_L\rangle, \qquad \sigma_z^{(2)}|0_L\rangle, \qquad \sigma_z^{(3)}|0_L\rangle, && (236)\\ &|1_L\rangle, \qquad \sigma_z^{(1)}|1_L\rangle, \qquad \sigma_z^{(2)}|1_L\rangle, \qquad \sigma_z^{(3)}|1_L\rangle && (237)\end{aligned}$$

are all orthogonal where $\sigma_z^{(j)}$ denotes a σ_z error on qubit j.

We can correct the state by projecting onto one of the four subspaces

$$|0_L\rangle\langle 0_L| + |1_L\rangle\langle 1_L|, \qquad \sigma_z^{(j)}|0_L\rangle\langle 0_L|\sigma_z^{(j)} + \sigma_z^{(j)}|1_L\rangle\langle 1_L|\sigma_z^{(j)} \tag{238}$$

for $j = 1, 2, 3$. We combined the codes by *concatenating them.* We encode using phase-flip and then bit-flip:

$$|+\rangle \to |+\rangle^{\otimes 3} \to \frac{1}{\sqrt{8}}\left(|000\rangle + |111\rangle\right)^{\otimes 3}, \qquad |-\rangle \to |-\rangle^{\otimes 3} \to \frac{1}{\sqrt{8}}\left(|000\rangle - |111\rangle\right)^{\otimes 3} \tag{239}$$

which is the nine-qubit code. We note this code corrects any Pauli error on any qubit. σ_x are corrected by bit-flipping and σ_z are corrected by phase-error correcting code. $\sigma_y = i\sigma_x\sigma_z$ and so we can correct these as well.

29.2 Arbitrary errors

We will now consider arbitrary unitary errors (e.g. a measurement on qubits or a more general type of quantum transformation). We claim that this nine-qubit code corrects all of these errors.

Theorem 29.1. *Any quantum error-correcting code which corrects t or fewer Pauli errors ($\sigma_x, \sigma_y, \sigma_z$ errors) on a subset of t or fewer qubits will also correct an arbitrary quantum operation which is applied to at most t qubits.*

The nine-qubit code corrects any arbitrary single-qubit error follows from the theorem with $t = 1$.

We will first consider an example on the three-qubit phase-flip error correcting code.

Example 29.2. Consider $\alpha|0\rangle + \beta|1\rangle$ encoded in the three-qubit phase-flip correcting code:

$$\frac{1}{2}\alpha\left(|000\rangle + |011\rangle + |101\rangle + |110\rangle\right) + \frac{1}{2}\beta\left(|001\rangle + |010\rangle + |100\rangle + |111\rangle\right). \tag{240}$$

We want to consider what happens when we apply $\begin{bmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{bmatrix}$ to the second qubit. We get

$$\frac{1}{2}e^{-i\theta}\left(\alpha|000\rangle + \alpha|101\rangle + \beta|001\rangle + \beta|100\rangle\right) + \frac{1}{2}e^{i\theta}\left(\alpha|011\rangle + |110\rangle + \beta|010\rangle + \beta|111\rangle\right). \tag{241}$$

We can use the identities $e^{i\theta} = \cos\theta + i\sin\theta, e^{-i\theta} = \cos\theta - i\sin\theta$. Then the above simplifies to

$$\cos\theta\left(\alpha|0_L\rangle + \beta|1_L\rangle\right) - i\sin\theta\sigma_z^{(2)}\left(\alpha|0_L\rangle + \beta|1_L\rangle\right). \tag{242}$$

When we make a measurement, we get that there is no error with probability $\cos^2\theta$ and a σ_z in the second qubit with probability $\sin^2\theta$. When we correct the σ_z error on the second qubit, we recover the original state. This is because

$$\begin{bmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{bmatrix} = (\cos\theta)I - i(\sin\theta)\sigma_z. \tag{243}$$

Then for a general error we have

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} = \frac{1}{2}(\alpha+\delta)I + \frac{1}{2}(\alpha-\delta)\sigma_z + \frac{1}{2}(\beta+\gamma)\sigma_z + \frac{i}{2}(\beta-\gamma)\sigma_y \tag{244}$$

Then if we have one error on any qubit, there is some nonzero probability of finding it, e.g. square amplitudes.

Example 29.3. Suppose we measure qubit 5 in the $\{|+\rangle, |-\rangle\}$ basis and measured $|-\rangle$. This corresponds to projection matrix

$$|-\rangle\langle-| = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \frac{1}{2}(I - \sigma_x). \tag{245}$$

If we apply error correction, we will get I with probability $\frac{1}{2}$ and σ_x with probability $\frac{1}{2}$.

Example 29.4 (Small error on all qubits)**.** We will now consider a small error on all qubits

$$\begin{bmatrix} e^{-2\pi i/200} & 0 \\ 0 & e^{2\pi i/200} \end{bmatrix} = \cos\frac{\pi}{100}I - i\sin\frac{\pi}{100}\sigma_z \approx 0.999I - i(0.03)\sigma_z. \tag{246}$$

Because this is an error on all qubits, we have

$$(0.999I - 0.03i\sigma_z^{(i)})^n = 0.999^n I - 0.03i\sum_{i=1}^{n}\sigma_z^{(i)} + 0.009\sum_{i=1}^{n}\sum_{j=1}^{n}\sigma_z^{(i)}\sigma_z^{(j)}(1-\delta_{ij}) + \cdots \tag{247}$$

So most of the amplitude is in cases with errors on $< \frac{n}{500}$ qubits. So if your error correcting code corrects $\frac{n}{500}$ phase-flip errors, we have a small chance of error.

29.2.1 Some formalism

The more formal proof of the above theorem. We will prove that if an error-correcting code can correct errors described by matrices M_1, M_2, M_3, then it can correct errors described by a linear combination of these. Then we show that any error on t qubits is a linear combination of Pauli errors on t qubits.

The first step comes from linearity of quantum mechanics. If we consider an error correcting circuit, we can apply the principle of delayed measurement to postpone measurements until the end. Instead of measuring error classically and applying Pauli matrices, we measure the error coherently and use controlled Pauli gates to correct erors. This gives us a unitary

$$M_i|\psi\rangle|0^k\rangle \to |\psi\rangle|D_i\rangle \tag{248}$$

where $|D_i\rangle$ a description of the error. Then if we have an error $F = \sum_i M_i$, we can correct it via

$$F|\psi\rangle|0^k\rangle = \sum_i \alpha_i M_i|\psi\rangle|0^k\rangle \to |\psi\rangle \sum_i \alpha_i|D_i\rangle. \tag{249}$$

This shows that error correction *measures the error without measuring the encoded quantum state.* This lets us correct the error without measuring the quantum state.

Now consider a small error on every qubit. Let there be n qubits where the error on the ith qubit is

$$F_i = (1-\epsilon_i)I + \delta_{x,i}\sigma_x^{(i)} + \delta_{y,i}\sigma_y^{(i)} + \delta_{z,i}\sigma_z^{(i)}. \tag{250}$$

We can expand the produce $\prod_i F_i$. Most of the amplitude of the product will be in terms with few Pauli errors, so all we need to do is correct any tensor product of $< \frac{n}{100}$ Pauli errors and if the δ's are small enough, there will be Pauli errors on fewer than $\frac{n}{100}$ qubits. The probability that you make an error that is too large to be corrected is very small.

30 November 19th, 2021

Today, we will discuss the seven-qubit Hamming code. The Hamming code is the simplest CSS code.

30.1 Classical Hamming code

The codewords of the Hamming code are the binary linear combination of the generator matrix

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{251}$$

Note that this is different from the generator matrix we gave in lecture 28.

The Hamming code encodes four bits into seven bits. We take a four-bit message m to get a codeword

$$c = mG. \tag{252}$$

The Hamming code is the simplest non-trivial example of *linear codes.* In linear codes, codewords are binary linear combinations of some generator G. A generalization of these, called *stabilizer codes*, exist, but we will not discuss these today.

30.1.1 Correcting errors

We correct errors using the *parity check* matrix H, which is a generator matrix for the dual space of the code. The dual space of a vector space V is

$$V^\perp = \{\boldsymbol{w} : \boldsymbol{w} \cdot \boldsymbol{v} = 0 \forall \boldsymbol{v} \in V\}. \tag{253}$$

We note that in binary vector spaces, the dual can overlap with the dual. It is still true, however, that $\dim V + \dim V^\perp = n$.

The Hamming code corrects one error because the minimum non-zero codeword has *Hamming weight* 3.

Definition 30.1 (Hamming weight and distance)**.** The *Hamming weight* of a codeword is the number of non-zero coordinates of a codeword. For binary codewords, this is the number of 1's.

The *Hamming distance* $d_H(c_1, c_2)$ is the Hamming weight of $c_1 - c_2$, or the number of coordinates where c_1, c_2 differ.

Theorem 30.2. *Suppose that the minimum non-zero weight of a code is d. The code can correct $t = \lfloor \frac{d-1}{2} \rfloor$ errors and detect $d - 1$ errors.*

Proof. We will show that a word w cannot be within t of two different codewords c_1, c_2. Suppose it is, then

$$d_H(c_1, c_2) \leq d_H(c_1, w) + d_H(w, c_2) \leq 2 \left\lfloor \frac{d-1}{2} \right\rfloor \leq d - 1 \tag{254}$$

but $c_1 - c_2$ is a codeword $\implies$ $d_H(c_1, c_2) \geq d$, a contradiction. So for any word, there is at most one codeword within t of it $\implies$ if a codeword has t or fewer errors, we can correct it to the unique codeword within Hamming distance t.

Moreover, if a codeword has fewer than $d-1$ errors, then the errors cannot take it to another codeword. We can detect that there is an error! □

The parity check matrix for the Hamming code is H^T where

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{255}$$

The parity check matrix is often written as a 7×3 instead of a 3×7 matrix, but this representation is useful for the quantum Hamming code.

We note that the rows of H are the same as the first three rows of G. We have the property $GH^T = 0$, which is important. Then we note that the code $C^\perp \subset C$.

The Hamming code is called a $(7, 4, 3)$ code because it maps 4 bits into 3 bits with minimum weight 3. A code whose generator matrix is H is a $(7, 3, 4)$ code.

Claim. Every non-zero codeword has Hamming weight 4.

The Hamming code can correct one error, for example let the error be $e = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$. Suppose we receive an encoded message with error given by

$$r = mG + e. \tag{256}$$

We can correct the message by multiplying by H^T:

$$rH^T = (mG + e)H^T = eH^T \tag{257}$$

where eH^T is called the *syndrome*. Because e contains a single 1 in the kth position, eH is the kth row of H^T.

Example 30.3 (Hamming codes)**.** Suppose we receive a noisy codeword $r = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$. We can calculate

$$rH^T = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} = eH^T \tag{258}$$

which is the third row of H^T. Thus, the third bit of the codeword has an error. The correct codeword is $\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$, where the third bit is corrected. We can use linear algebra to show that this is the encoding of the message $\begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}$.

We note that $H \subseteq H^\perp = G$.

Definition 30.4 (Weakly self-dual codes)**.** A code C is *weakly self-dual* if $C \subseteq C^\perp$. These codes are important for building quantum error-correcting codes.

30.2 Quantum Hamming code

The quantum Hamming code encodes two bits into one bit. The logical $|0\rangle$ is a superposition of all eight codewords in H

$$|0_L\rangle = \frac{1}{\sqrt{8}} \sum_{c \in H} |c\rangle \tag{259}$$

and the logical $|1\rangle$ is the superposition of the eight codewords that are in G and *not* H. These are codewords of $H \oplus |1111111\rangle$:

$$|1_L\rangle = \frac{1}{\sqrt{8}} \sum_{c \in H} |c \oplus 1111111\rangle. \tag{260}$$

This corrects an arbitrary one-bit error. We will show that this corrects a single σ_x, σ_z error. Moreover, this code treats these two separately, so we can also correct a σ_y.

30.2.1 Bit-flip error

The state is a superposition of elements in the classical Hamming code. We can apply the classical Hamming error correction to each element to correct a single bit-flip error. We can compute the syndrome. If b_i be the ith bit. The syndrome is given by

$$\begin{bmatrix} b_4 \oplus b_5 \oplus b_6 \oplus b_7 & b_2 \oplus b_3 \oplus b_6 \oplus b_7 & b_1 \oplus b_3 \oplus b_5 \oplus b_7 \end{bmatrix}. \tag{261}$$

The circuit is

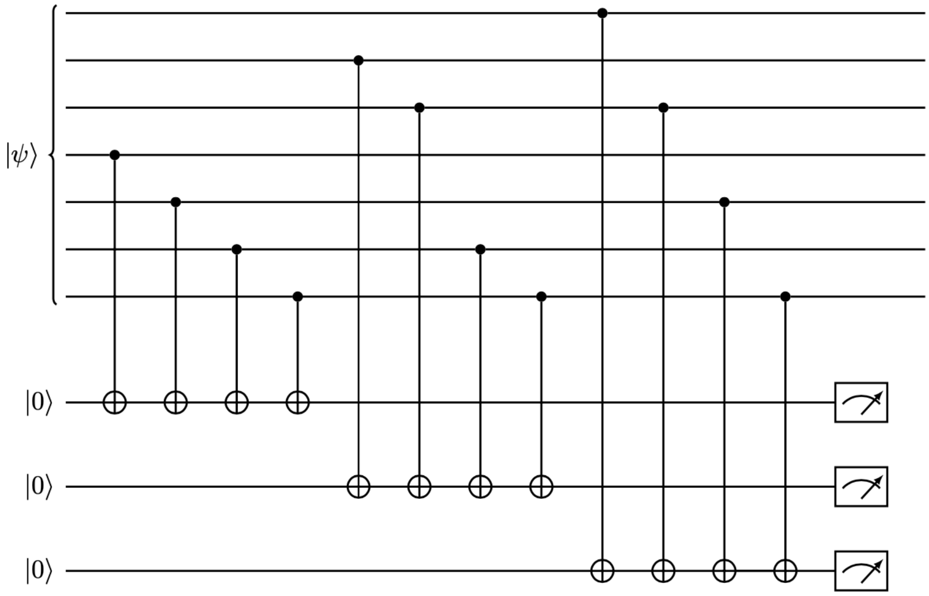


Figure 12: Quantum Hamming syndrome computation circuit

We can then determine the error and correct it. This can be done classically.

30.2.2 Phase-flip error

Recall that the Hadamard makes bit-flip to phase-flip errors. Then we just need to see what happens when we apply a Hadamard to every qubit of the code. It turns out that we get the same quantum code back! Then the circuit for measuring the syndrome for phase-flips is the same as bit-flips except we put seven Hadamard gates on the qubits in the quantum code at the front of the circuit and seven more at the back.

More formally, let us consider applying $H^{\otimes 7}$ to the quantum code. We have

$$|0_L\rangle = \frac{1}{\sqrt{8}} \sum_{x \in H} |x\rangle \implies H^{\otimes 7}|0_L\rangle = \frac{1}{\sqrt{8}} \frac{1}{\sqrt{2^7}} \sum_{x \in H} \sum_{y \in \mathbb{Z}_2^7} (-1)^{x \cdot y} |y\rangle. \tag{262}$$

If $y \in G$, because all elements of G are orthogonal to all elements in H, the phases are $+1$ and the sum evaluates to $8|y\rangle$. If $y \notin G$, half the phases are -1 and the other half are $+1$. Then

$$H^{\otimes 7}|0_L\rangle = \frac{8}{\sqrt{8}} \frac{1}{\sqrt{2^7}} \sum_{y \in G} |y\rangle = \frac{1}{4} \left(\sum_{y \in H} |y\rangle + \sum_{y \in G \setminus H} |y\rangle \right) = \frac{1}{\sqrt{2}} \left(|0_L\rangle + |1_L\rangle \right) \tag{263}$$

and

$$H^{\otimes 7}|1_L\rangle = \frac{1}{\sqrt{8}} \frac{1}{\sqrt{2^7}} \sum_{x \in G \setminus H} \sum_{y \in \mathbb{Z}_2^7} (-1)^{x \cdot y} |y\rangle. \tag{264}$$

If $y \notin G$, the sum vanishes. If $y \in H$, the sum is $8|y\rangle$ and if $y \in G \setminus H$, the sum is $-8|y\rangle$. this is because the inner produce of two vectors in $G \setminus H$ is -1. Then

$$H^{\otimes 7}|1_L\rangle = \frac{1}{\sqrt{2}}\left(|0_L\rangle - |1_L\rangle\right). \tag{265}$$

Then $H^{\otimes 7}|0_L\rangle, H^{\otimes 7}|1_L\rangle$ are states in the quantum Hamming code, and bit-flip errors can be corrected by the error correction procedure. This means phase-flip errors can be corrected by applying $H^{\otimes 7}$, applying bit-flip error correction, and applying $H^{\otimes 7}$ again.

30.3 Applying logical Pauli matrices

We can apply a logical σ_x and logical σ_z. For a logical σ_x, we note that we can turn any codewords in H into a codeworde in $G \setminus H$ and vice versa by adding $|1111111\rangle$, so the corresponding operation to $|0_L\rangle$ is $\sigma_x^{\otimes 7}$. Moreover, applying a logical σ_z is just applying a $\sigma_z^{\otimes 7}$. This is because $\sigma_z^{\otimes 7}$ applies a $+1$ phase to any $|y\rangle$ for $y \in H$ and a -1 phase to any $|y\rangle$ for $y \in G \setminus H$.

31 November 22nd, 2021

Today, we will discuss more Hamming codes and CSS codes.

31.0.1 Review

We reviewed the Hamming code from last lecture. An important detail we may have missed is that a classical binary linear code is a subspace of $\mathbb{Z}_2^n$.

31.1 Quantum Calderbank–Shor–Steane (CSS) codes

A CSS code is a quantum error-correcting code that is derived from two classical codes C_1, C_2 with $C_2 \subseteq C_1$. We call this code $\text{CSS}(C_1 : C_2)$.

Claim (CSS error correction)**.** If C_1 can correct t_1 errors and $C_2^\perp$, the dual code to C_2, can correct t_2 errors, then $\text{CSS}(C_1 : C_2)$ can correct t_1 bit-flip errors and t_2 phase-flip errors.

Note. We can consider a quantum error-correcting code to be a collection of codewords or the subspace generated by these words.

We will first discuss some terms from classical error correcting codes.

Definition 31.1 (Cosets)**.** If $C_2 \subseteq C_1$, a *coset* of C_2 in C_1 is the set of vectors

$$x + C_2 = \{y : y = x + c, c \in C_2\} \tag{266}$$

for some $x \in C_1$. Two cosets $x_1 + C_2$ and $x_2 + C_2$ are either equal or disjoint. They form a partition of the space C_2. Every $x \in C_1$ is contained in some coset. The number of cosets is $\left|\frac{C_1}{C_2}\right|$.

Example 31.2 (Hamming code cosets)**.** In the Hamming code, there are two cosets of H in G, given by

$$H, \qquad H + |1111111\rangle. \tag{267}$$

Definition 31.3 ($\text{CSS}(C_1 : C_2)$)**.** Let C_1, C_2 be two n-bit classical codes with $C_2 \subset C_1$. The CSS code $\text{CSS}(C_1 : C_2)$ is the code with codewords

$$|x + C_2\rangle = \frac{1}{|C_2|^{1/2}} \sum_{c \in C_2} |x + c\rangle \tag{268}$$

for each coset $x + C_2$. The codewords are associated with cosets of C_2 in C_1. Thus, the number of codewords is the number of cosets of C_2 in C_1, which is $\left|\frac{C_1}{C_2}\right|$. The dimension of the code subspace is

$$\dim \text{CSS}(C_1 : C_2) = \log_2 \frac{|C_1|}{|C_2|} = \dim C_1 - \dim C_2. \tag{269}$$

We will now show that $\text{CSS}(C_1 : C_2)$ corrects t_1 bit-flip errors. There is a classical error-correction procedure that will correct t_1 or fewer errors in any codeword c_1 of C_1. The codewords of $\text{CSS}(C_1 : C_2)$ are a superposition of quantum states $|x + c\rangle$, each of which is a codeword of C_1. Applying the classical algorithm in quantum superposition will correct up to t_1 bit-flip errors in the quantum code.

The dual code corrects t_2 phase errors. We recall the the dual W of vector space V is

$$W^\perp = \{x \in V : x \cdot w = 0 \forall w \in W\}. \tag{270}$$

Consider applying the Hadamard transform to codewords $|x + C_2\rangle$. We get superpositions of the codewords of the code $\text{CSS}(C_2^\perp : C_1^\perp)$.

Note. We note $C_2 \subseteq C_1 \implies$ any vector perpendicular to everything in C_1 is perpendicular to everything in $C_2 \implies C_1^\perp \subseteq C_2^\perp$ and $\dim C_1^\perp = n - \dim C_1, \dim C_2^\perp = n - \dim C_2$. We have

$$\dim \mathrm{CSS}(C_1 : C_2) = \dim \mathrm{CSS}(C_2^\perp : C_1^\perp). \tag{271}$$

Now we consider

$$\begin{aligned} H^{\otimes n}|x + C_2\rangle = \frac{1}{\sqrt{2^{\dim C_2}}} H^{\otimes n} \sum_{c_2 \in C_2} |c_2 + x\rangle &= \frac{1}{\sqrt{2^{\dim C_2}}} \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_2^n} \sum_{c_2 \in C_2} (-1)^{(c_2+x)\cdot y} |y\rangle \\ &= \frac{1}{\sqrt{2^{\dim C_2}}} \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_2^n} (-1)^{x \cdot y} \sum_{c_2 \in C_2} (-1)^{c_2 \cdot y} |y\rangle. \end{aligned} \tag{272}$$

Now if $y \notin C_2^\perp$, we have

$$\sum_{c_2 \in C_2} (-1)^{c_2 \cdot y} = \begin{cases} |C_2|, & y \in C_2^\perp \\ 0, & y \notin C_2^\perp \end{cases} \tag{273}$$

If $y \in C_2^\perp$, all elements of the sum are 1 and we get $|C_2|$. If not there is some $d \in C_2 : d \cdot y = 1$ and we can pair elements of C_2 into $(c_2, c_2 + d)$ where the sum of each pair is 0, so the whole sum is 0. Thus we have

$$H^{\otimes n}|x + C_2\rangle = \frac{1}{\sqrt{2^{\dim C_2}}} \frac{1}{\sqrt{2^n}} \sum_{y \in C_2^\perp} (-1)^{x \cdot y} |C_2| |y\rangle = \frac{1}{\sqrt{2^{\dim C_2^\perp}}} \sum_{y \in C_2^\perp} (-1)^{x \cdot y} |y\rangle. \tag{274}$$

We are almost done!

Claim. If y_1, y_2 in the same coset of $C_1^\perp$ in $C_2^\perp$, then $(-1)^{x \cdot y_1} = (-1)^{x \cdot y_2}$ because $y_1 - y_2 \in C_1^\perp, x \in C_1 \implies x \cdot (y_1 - y_2) = 0$.

Let R be a set of representatives of cosets of $C_1^\perp$ in $C_2^\perp$. There is one element of each coset in R. We can group the sum over y above into elements in the representation. Then

$$H^{\otimes n}|x + C_2\rangle = \sqrt{\frac{2^{\dim C_1^\perp}}{2^{\dim C_2^\perp}}} \sum_{y \in R} (-1)^{x \cdot y} |y + C_1^\perp\rangle \tag{275}$$

which shows that the Hadamard transform of a codeword in $\mathrm{CSS}(C_1 : C_2)$ is a superposition of codewords in $\mathrm{CSS}(C_2^\perp : C_1^\perp)$ as desired.

31.2 Shifting a code

We can take a CSS code and shift it in bit space or phase space. A codeword for a code shifted by s in bit space is

$$|x + C_2\rangle = \frac{1}{|C_2|^{1/2}} \sum_{c \in C_2} |s + x + c\rangle. \tag{276}$$

If $s \in C_2$, nothing changes. If $s \in C_1$, this permutes the codewords. If $s \notin C_1$, we get a new code! It still corrects the same number of errors.

A codeword for a code shifted by t in phase space is

$$|x + C_2\rangle = \frac{1}{|C_2|^{1/2}} \sum_{c \in C_2} (-1)^{t \cdot (x+c)} |x + c\rangle. \tag{277}$$

We can also shift in phase space *and* bit space. The ordering here makes a difference. We have

$$|x + C_2\rangle = \frac{1}{|C_2|^{1/2}} \sum_{c \in C_2} (-1)^{t\cdot(x+c)} |s + x + c\rangle. \tag{278}$$

The shifted error correcting codes correct the same number of errors as unshifted ones.

Note. An important fact to know for the proof of security of quantum key distribution is that if you shift both the bit space and phase space by a random vector, and average over resulting codes, we get a uniform distribution with density matrix I_{2^n}.

32 November 24th, 2021

Today, we will discuss more Bell-type quantum “paradoxes."

We will begin witht he Mermin-Peres magic square game and then go into the Kochen-Specker theorem.

32.1 Mermin-Peres magic square game

We have the following game. Consider Alice and Bob, players who are *not* allowed to communicate during the game. The play a game on a tic-tac-toe board. There is a referee that gives both Alice and Bob row and column assignments. They must then fill their respective row/column with $+/-$. Alice and Bob *win* if they agree at an intersection. Alice has an even number of $-$ and Bob has an odd number of $-$.

Note. The product of Alice's signs are positive and Bob's signs are negative.

Example 32.1 (Magic square game)**.** Consider the instructions {Alice: row 3, Bob: column 1} from the referee. Then the winning play is

-		
+		
+	-	-

Table 1: Winning play for Alice, Bob.

There **does not** exist a deterministic *classical* winning strategy for Alice and Bob. If there was a solution, there would be a filling of the grid with an odd number of $-$ in each column and an even number of $-$ in each row. We cannot have this because of the parity of the grid. The maximum probability that Alice, Bob win is $\frac{8}{9}$. The deterministic strategy is given by

+	+	+
+	+	+
-	-	+

Table 2: Winning strategy for Alice, Bob.

where they win if the referee asks them for {Alice: row 3, Bob: column 3}.

32.1.1 Quantum mechanical strategy

If Alice and Bob share two EPR pairs

$$\frac{1}{\sqrt{2}}\left(|00\rangle_{AB} + |11\rangle_{AB}\right) \otimes \frac{1}{\sqrt{2}}\left(|00\rangle_{AB} + |11\rangle_{AB}\right) \tag{279}$$

we can consider the following matrix

$I \otimes \sigma_z$	$\sigma_z \otimes I$	$\sigma_z \otimes \sigma_z$
$\sigma_x \otimes I$	$I \otimes \sigma_x$	$\sigma_x \otimes \sigma_x$
$-\sigma_x \otimes \sigma_z$	$-\sigma_z \otimes \sigma_x$	$\sigma_y \otimes \sigma_y$

Table 3: Quantum mechanical strategy.

where each entry is a Hamiltonian, or a measurement operator.

If Alice gets row i, she measures her qubits with the three observables in the row. Ditto for Bob for the columns.

We require three things.

Commutativity. It is easy to check that all operators in each row (and in each column) commute. For example

$$(\sigma_x \otimes \sigma_z)(\sigma_z \otimes \sigma_x) = \sigma_x\sigma_z \otimes \sigma_z\sigma_x = -\sigma_z\sigma_x \otimes -\sigma_x\sigma_z = (\sigma_z \otimes \sigma_x)(\sigma_x \otimes \sigma_z). \tag{280}$$

This means that Alice and Bob can *simultaneously measure* the three observables.

Identity. We also check that the product of observables in each row is I and the product of observables in each column is $-I$. In the third row, we have

$$\sigma_x\sigma_z\sigma_y \otimes \sigma_z\sigma_x\sigma_y = -i\sigma_y^2 \otimes i\sigma_y^2 = I \tag{281}$$

where we have used $\sigma_x\sigma_z = -i\sigma_y, \sigma_z\sigma_x = \sigma_y$. This means that when we take the three eigenvalues given by the observables in each row, we will always get 1.

Parity. We finally require that Alice and Bob get the same result in each cell after measurement. We can verify that the measurement results are guaranteed to multiply out to $+1$ for Alice and -1 for Bob.

32.2 Kochen-Specker theorem

This theorem relates to a hidden variable quantum mechanics theory.

We recall the spins of a spin-1 particle.

$$J_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad J_x = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad J_y = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & -i & 0 \\ i & 0 & -i \\ 0 & i & 0 \end{bmatrix}, \tag{282}$$

$$J_z^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad J_x^2 = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad J_y^2 = \frac{1}{2}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 1 \end{bmatrix}. \tag{283}$$

Let us suppose that there were hidden variables and the spin $J_{\boldsymbol{v}}^2$ was determined beforehand for every axis. Note that $J_{\boldsymbol{v}}^2 = \{0, 1\}$. Then there exists a coloring of the sphere with red and green that any orthogonal triplet has exactly one green. It turns out they found a set of 115 axes and proved that you cannot color the axes in this way.

The vectors are of the form $\boldsymbol{v} = (\pm 1, \pm 1, 0), (\pm 1, 0, 0), (\sqrt{2}, \pm 1, \pm 1), (\sqrt{2}, \pm 1, 0)$ which correspond to points on a cube. The coloring can be found in the lecture recording.

33 November 29th, 2021

Today, we will discuss the BB84 key distribution protocol, based on quantum error-correcting codes. It was invented by Charlie Bennett and Gilles Brassard in 1984. We will then prove its security using CSS QECC.

33.0.1 Key distribution protocols

The basic idea of a key distribution protocol is that there are two participants, Alice and Bob that want to agree on a secret key that an eavesdropper Eve does not know. We assume that Alice and Bob start the protocol without secret information and must agree on a secret key using a public channel. Classically, we need to base the protocol on a hard problem that Eve cannot solve. In quantum mechanics, however, we do not need to make an hardness assumptions since security comes from the laws of quantum mechanics.

There are some assumptions we need to make to protect against a man-in-the-middle attack. Note that there is no identification information. If Eve inserts herself in the channel, she can pretend to be Bob and Alice to Alice and Bob, respectively. We will assume that Alice and Bob have a classical channel that is *not spoofable* — Eve cannot take over the channel and pretend to be Bob. We will also assume that Alice and Bob have a quantum channel that Eve is allowed to do anything to (consistent with laws of physics).

Because Eve can simply cut this channel, we require that it is unlikely that Alice and Bob think they have shared the secret key while Eve has more than an exponentially small amount of information about the secret key.

The advantage of a quantum key distribution over a classical key distribution is that we don't rely on some hard problem. The disadvantage is that we need a quantum channel between two participants (like an optical fiber).

33.1 BB84 protocol

The protocol is as follows:

1. Alice sends Bob a sequence of qubits in one of four states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$. Consider the sequence
2. Bob measures the sequence in a random basis 0/1 or $+/-$.
3. Alice sends Bob the basis and Bob discards the qubits for which the measurement results do not match.
4. Alice and Bob choose a random sample of the qubits to use to check whether they agree. If they do, they know Eve could not have been measuring many qubits. They then use the remaining qubits into a secret key (like mapping $|0\rangle, |+\rangle \to 0, |1\rangle, |-\rangle \to 1$).

Example 33.1 (BB84 protocol)**.** We can consider the following protocol applied to a sequence of qubits

Alice prepares	$\|0\rangle$	$\|1\rangle$	$\|+\rangle$	$\|+\rangle$	$\|0\rangle$	$\|-\rangle$	$\|-\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$
Bob measures	0/1	0/1	0/1	$+/-$	$+/-$	0/1	$+/-$	0/1	$+/-$	0/1	0/1
and gets	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|+\rangle$	$\|+\rangle$	$\|1\rangle$	$\|-\rangle$	$\|1\rangle$	$\|+\rangle$	$\|0\rangle$	$\|0\rangle$
Alice's basis	0/1	0/1	$+/-$	$+/-$	0/1	$+/-$	$+/-$	0/1	0/1	0/1	0/1
places they agree	$\|0\rangle$	$\|1\rangle$		$\|+\rangle$			$\|-\rangle$	$\|1\rangle$		$\|0\rangle$	$\|0\rangle$
check qubits		?		?						?	
Alice		1		+						0	
Bob		1		+						0	
secret key	0						1	1			0

Table 4: BB84 protocol example.

This protocol works because if Eve measures a qubit, she does not know which basis to measure in. If she chooses the wrong basis, she will disturb the quantum state and Alice and Bob will notice that their check bits disagree.

If the channel is noisy, Alice and Bob's string of bits will disagree. Suppose Alice and Bob have length m strings a, b respectively. Because they tested their bits, they expect ϵm of their bits to differ, where ϵ small. Alice can choose an error correcting code C length m that corrects $\epsilon' m$ bits, $\epsilon' > \epsilon$. So accounting for random fluctuations in noise, the number of places where Alice and Bob's bits differ is less than $\epsilon' m$ with high probability. Alice chooses a random codeword $c \in C$ and sends

$$a \oplus c \tag{284}$$

to Bob. Bob then takes this message and subtracts b from it to get $a \oplus b \oplus c$. This is a string that differs from c in fewer than $\epsilon' m$ positions, so Bob can also apply error correction and get c. Alice and Bob can then share c.

It is possible that Eve has some information about c. To fix this, then, Alice and Bob choose a hash function that maps m bits into $\ell < m$ bits.

33.2 Adapted Lo-Chao protocol

We will now give a protocol where the security proof is relatively simple. Note in the original BB84 protocol, Bob needs quantum memory to store all qubits he receives and is not practical.

The idea behind this protocol is that if Alice and Bob share perfect EPR pairs, then measuring them in the 0/1 basis will give them a secret key. Eve can never determine the outcomes of measurements because Alice and Bob have perfect entanglement.

The protocol is as follows:

1. Alice prepares n EPR pairs
2. Alice chooses a CSS code $\mathrm{CSS}(C_1 : C_2)$ and a translate of it by s in bit space and t in phase space.
3. Alice encodes half of each EPR pair with this code, randomly intersperses test bits which are equally likely to be one of four bases $|0\rangle, |1\rangle, |+\rangle, |-\rangle$ and sends this string to Bob.
4. Bob puts everything into quantum memory.
5. Alice announces the code and strings s, t, which bits were test bits, which bits were code bits, and the values of the test bits.
6. Bob checks the test bits to determine the error rate. He then decodes the EPR pairs, and Alice and Bob measure EPR pair in the 0/1 basis to obtains a secret key.

Note that because Alice sends a random translate of $\mathrm{CSS}(C_1 : C_2)$, the test qubits are equally likely to be in any the four states, the density matrix Eve sees is completely random, i.e. is the identity matrix, so Eve cannot tell which qubits are the code qubits and which qubits are the test qubits.

Moreover, because the rate of noise on the test bits is sufficiently low, the probability the CSS code does not transmit the correct state is ϵ, where ϵ small. Then the state Alice and Bob share after transmission is

$$\sqrt{1-\epsilon}|\phi\rangle^{\otimes n} + \sqrt{\epsilon}|E\rangle \tag{285}$$

where $|\phi\rangle$ is the EPR pair and $|E\rangle$ an arbitrary error state. Then Eve has an exponentially small amount of information about the key after measurement.

33.3 Equivalence of protocols

We can assume that Alice measures her half of the EPR pairs before she encodes the quantum state. Thus we can assume that the quantum state that she sends is a random string of n classical bits which is encoded in $\mathrm{CSS}(C_1 : C_2)$.

When Alice encodes a random string of bits to encode, she is choosing a random coset $x + C_2$ and encoding it. Choosing a random coset x is the same as choosing a random bit string y and taking the coset $y + C_2$.

In the shifted CSS code, it looks like

$$\frac{1}{\sqrt{|C_2|}}\sum_{c_2\in C_2}(-1)^{t\cdot(y+c_2)}|s+y+c_2\rangle. \tag{286}$$

For the secret key, Bob needs to find the coset C_2 that this belongs to. He can find this coset by measuring in the 0/1 basis and subtract s to get $y+c_2$. Bob doesn't need t to find the coset so we can assume Alice never sends him t. Then the density matrix of her message is

$$\frac{1}{|C_2|}\left(\sum_{c_2\in C_2}(-1)^{t\cdot(y+c_2)}|s+y+c_2\rangle\right)\left(\sum_{c_2\in C_2}(-1)^{t\cdot(y+c_2)}\langle s+y+c_2|\right) \\ =\frac{1}{|C_2|}\sum_{c_2\in C_2}|s+y+c_2\rangle\langle s+y+c_2| \tag{287}$$

which is the same as Alice taking a random c_2 and adding it to $s+y$. Because the channel may be noisy, Bob actually gets $s+y+c_2+e$ where e some error. He needs to subtract s and apply classical decoding procedure to get $y+c_2$.

We can now compare the protocols. In BB84, Alice sends Bob a and Bob receives $b=a+e$. Alice then sends Bob $a+c_1$ on the classical channel and Bob subtracts the quantities to obtain c_1+e. Bob then decodes using QECC and obtains $c_1\in C_1$.

In Lo-Chau, Alice sends Bob $s+y+c_2$ where $y+c_2\in C_1$. Bob receives $s+y+c_2+e$. Alice then sends bob s. Bob subtracts and obtains $y+c_2+e$. Bob then uses QECC and obtains $y+c_2\in C_1$.

Thus BB84 is secure because these two protocols are equivalent.

34 December 1st, 2021

35 December 3rd, 2021

36 December 6th, 2021

37 December 8th, 2021