CS378 COURSE NOTES

JEFFREY JIANG

These notes are from the class *Introudction to Quantum Information* taught by Professor Aaronson in Spring 2017

Contents

1. The Rules of Quantum Mechanics

We already know for standard probability, there is a translation into the world of linear algebra.

Definition 1.1. A vector $v \in \mathbb{R}^n$ is a **probability vector** if the entries are all positive, and sum to 1

Definition 1.2. A matrix $S \in \mathcal{M}_{nn}$ is a **stochastic matrix** if all of its columns are probability vectors.

Given two probabilty vectors (for simplicity, say $v, w \in \mathbb{R}^2$), each represents a separate state. If we want to consider the system of both states simultaneously, we use the **tensor product** of the two vectors, where

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$

Note that the tensor product is not commutative. In general $v \otimes w$ is not necessarily the same as $w \otimes v$. Some probability vectors can be factored into a tensor product of two smaller probability vectors, while others can't. If a vector cannot be factored, we say that the vector is **correlated**

We can also take the tensor product of matrices

Example 1.3. Consider the NOT matrix and the identity matrix I

$$\mathsf{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad\qquad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Their tensor product, denoted $\mathsf{NOT} \otimes I$ is then given by the matrix

$$\mathsf{NOT} \otimes I = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

In general, the matrix on the left gives the block form for the tensor product, and the second matrix specifies what lies in each block.

In the general 2×2 case, given matrices A, B, with

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

We have that in block form, $A \otimes B$ is the 4×4 matrix

$$A \otimes B = \begin{pmatrix} aB & bB \\ cB & dB \end{pmatrix}$$

We know probabilities are just numbers in the unit interval $[0, 1]$, but quantum mechanics requires a richer set of matrices and vectors - the world of unitary transformations and probability amplitudes, both of which can be complex valued.

Definition 1.4. The **state** of a **quantum system** is given by a unit vector in $\mathbb{C}^n$ i.e. a state can be expressed by some $v \in \mathbb{C}^n$ such that

$$\langle v, v \rangle = \sum_{i=1}^{n} |\alpha_i|^2 = 1$$

Where $|\alpha|$ denotes the complex modulus of α, given by the formula $\alpha\alpha^*$ and $\langle -, - \rangle$ is the Hermitian inner product on $\mathbb{C}^n$.

Remark. We note that while the standard inner product on $\mathbb{R}^n$ is symmetric, in the complex case we face a minor technicality that

$$\langle v, w \rangle = \langle w, v \rangle^*$$

Upon measurement, we observe the outcome i with probability $p = |\alpha_i|^2$ (The Born Rule).

Definition 1.5. A **qubit** is any two-state quantum system, given by the vector

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

with $\alpha, \beta \in \mathbb{C}$. For notational convenience, this is written in Dirac Notation as $\alpha |0\rangle + \beta |1\rangle$ where $|0\rangle$ and $|1\rangle$ represent the standard basis, sometimes referred to as the **computational basis**

Definition 1.6. Given a column vector $v \in \mathbb{C}^n$, it's **conjugate transpose**, denote $v^\dagger$ is given by transposing v into a row vector and taking the complex conjugate of all its entries. The same hold for matrices as well.

In Dirac notation, we have that the conjugate transpose of a vector $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ is given by the corresponding bra, $\langle\psi| = \alpha^* \langle 0| + \beta^* \langle 1|$. We note for any quantum state, we have that $\langle\psi| \, |\psi\rangle = 1$, and in general, $\langle v| \, |w\rangle = \langle v, w \rangle$, justifying the notation.

Remark. For the more linear algebra inclined, the kets form a vector space V over the field $\mathbb{C}$, and the bras are the corresponding dual space V^*, consisting of the space of linear functionals $f : V \to \mathbb{C}$. In addition, given some basis $\{v_1, \dots v_n\} \subset V$, we get the corresponding dual basis $\{v_1^*, \dots, v_n^*\}$ where $v_i^*(v_j) = \delta_{ij}$.

So probability vectors get their special matrices (Stochastic Matrices) that preserve the 1-norm, i.e. given a probability vector v, after the application of a stochastic matrix S, the resulting vector Sv is still a probability vector. Similarly, qubits have their corresponding linear transformations

Definition 1.7. A complex valued matrix U is **Unitary** if $U^\dagger = U^{-1}$. Equivalently, a matrix is unitary if its columns or rows form an othonormal basis for $\mathbb{C}^n$.

Unitary matrices have the property that the preserve the 2-norm (i.e. Hermtian inner product,), so

$$\langle v| U |w\rangle = \langle v| w\rangle$$

.

So we have the kets $|0\rangle |1\rangle$, which are just orthonormal basis for $\mathbb{C}^2$, we have a few other special vectors we'd like to define

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

Measurement in any orthonormal basis then consists of projection onto the basis vectors. For example, given some orthormal basis $\{|v\rangle, |w\rangle\}$, and some state $|\psi\rangle$, the probability of measure the outcome $|v\rangle$ is given by

$$|\langle\psi| v\rangle|^2$$

Remark. For measurement, a phase shift is not observable. In other words, for $|v\rangle$ and $e^{ix}|v\rangle$, the probabilities for any measurement are the same, so they are physically indistinguishable.

1.1. **Some Special Unitary Matrices.**

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

The Hadamard matrix is a change of basis from the $\{|0\rangle, |1\rangle\}$ basis to the $\{|+\rangle, |-\rangle\}$ basis, and is its own inverse. We have that

$$H|0\rangle = |+\rangle \qquad H|1\rangle = |1\rangle$$
$$H|+\rangle = |0\rangle \qquad H|-\rangle = |1\rangle$$

Likewise, there's a similar change of basis to the $\{|i\rangle, |-i\rangle\}$ basis

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

We also have our "favorite unitary matrix" the CNOT gate

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

1.2. **Distiguishability of States.**

Given any two orthogonal states $|v\rangle$ and $|w\rangle$, these are perfectly distinguishable- we just need to measure in the $\{|v\rangle, |w\rangle\}$ basis! However, we know that $-|v\rangle$ and $|v\rangle$ are not distinguishable at all. In general, given arbitrary states $|v\rangle, |w\rangle$, we can check how distinguishable they are by calculating the inner product

$$|\langle v| w\rangle|^2$$

The inner product is 0 if they are perfectly distinguishable (orthogonal) and 1 if they are not.

Given two states, the best basis to measure them in order to distinguish them is one that shares the same bisector as the two states.

1.3. 2 Qubit States.

Remark. For notational compactness, we will often denote the vector $|v\rangle \otimes |w\rangle$ as either $|vw\rangle$ or $|v\rangle\,|w\rangle$

For a two qubit state, we will need 4 probability amplitudes to specify the state of the system

$$\alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle = \begin{pmatrix}\alpha\\ \beta\\ \gamma\\ \delta\end{pmatrix}$$

If we want to understand the states of the individual qubits in the system, we note that we measure $|0\rangle$ for the first qubit with probability $|\alpha|^2 + |\beta|^2$ and we measure $|1\rangle$ with probability $|\gamma|^2 + |\delta|^2$. Also, we know that if we measure $|0\rangle$ for the first qubit, then the second qubit must be in the state $\alpha\,|0\rangle + \beta\,|1\rangle$. This is the **partial measurement rule**. Some unitary transformations on two qubit states can be expressed as applying one unitary U to one qubit, and applying another unitary T to the other qubit. These transformations are then given by the matrix $U \otimes V$. Like with correlation, if a two qubit state $|\psi\rangle$ is **entangled** if it cannot be factored into a tensor product of two one qubit states $|v\rangle \otimes |w\rangle$. Our favorite example of this is the **Bell Pair**

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

We note that after measuring one qubit, we immediately know the state of the other.

2. Mixed States and Density Matrices

So far, the only states we've discussed are what we call **pure states**, states that can be expressed as superpositions of quantum states. However, there are situations in which we want to introduce classical probability to give us what we call a **mixed state**.

Definition 2.1. Given a collection (somtimes referred to as an **ensemble**) of pure states $\{|\psi\rangle_i\}$ such that we have the state $|\psi_i\rangle$ with probability p_i, we can represent the mixed state as a **density matrix** ρ, given by the formula

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

Remark. The **outer product** of two vectors $|v\rangle$ and $|w\rangle$ is exactly what it looks like, the matrix product of a row vector and a column vector. It is given by the formula

$$|v\rangle\,\langle w| = \begin{pmatrix} v_1w_1^* & v_1w_2^* & \dots & v_1w_n^*\\ \vdots & \vdots & \vdots & \vdots\\ v_nw_1^* & \dots & \dots & v_nw_n^*\end{pmatrix}$$

Some common outer products we end up calculating quite often are

$$|0\rangle\,\langle 0| = \begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix} \qquad |1\rangle\,\langle 1| = \begin{pmatrix}0 & 0\\ 0 & 1\end{pmatrix}$$

$$|+\rangle\,\langle +| = \frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix} \qquad |-\rangle\,\langle -| = \frac{1}{2}\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}$$

So we have a representation for a mixed state as a matrix, but how do we extract information from the denisty matrix ρ? Well in the standard basis $\{|0\rangle \dots |n-1\rangle\}$, we measure the state $|j\rangle$ with probability

$$\Pr(|j\rangle) = \langle j|\,\rho\,|j\rangle = \rho_{jj}$$

So the diagonal encodes the probability of measuring one of the standard basis states.

How do unitary transfomations interact with the density matrix? If we apply the unitary U to our ensemble of states $\{|\psi_i o\rangle\}$, we calculate ρ_U to be

$$\begin{aligned}\rho_U &= \sum_i p_i U |\psi_i\rangle \langle\psi_i| U^\dagger \\ &= \sum_i p_i U |\psi_i\rangle (U |\psi_i\rangle)^\dagger \\ &= U \left(\sum_i p_i |\psi_i\rangle \langle\psi_i| \right) U^\dagger \\ &= U \rho U^\dagger\end{aligned}$$

Then by linearity, we get

We have that these density matrices satisfy the following properties:

(1) ρ is Hermitian, i.e. $\rho = \rho^\dagger$
(2) $\mathrm{Tr}(\rho) = 1$
(3) ρ is positive semidefinite, i.e. all eigenvalues of positive real numbers

Given an arbitrary density matrix ρ, we cal always express it as a mixed state with no cross terms, we do this via the **eigendecomposition** by finding the eigenvectors $\{|\psi_i\rangle\}$ and corresponding eigenvalues $\{\lambda_i\}$. Then

$$\rho = \sum_i \lambda_i |\psi_i\rangle \langle\psi_i|$$

Density matrices then give a very natural way to express single states that part of an entangled bipartite state

$$|\psi\rangle = \sum_i \sum_j \alpha_{ij} |i\rangle |j\rangle$$

via the formula

$$\sum_i \sum_j \sum_k \alpha_{ij} \alpha_{ik}^* |j\rangle \langle k|$$

Example 2.2. Consider the entangled state

$$|\psi\rangle = \frac{|00\rangle + |01\rangle + |10\rangle}{\sqrt{3}}$$

Suppose Alice holds the first (leftmost) qubit and Bob holds the second. How would we calculate Bob's local (sometimes called reduced) density matrix ρ_B? Suppose Alice measures her qubit, and gets the state $|0\rangle$. Then by the partial measurement rule, we know that Bob's qubit is now in the state (without renormalization)

$$|\psi\rangle_{|0\rangle} = |0\rangle + |1\rangle$$

Likewise, we have that if Alice measures $|1\rangle$,

$$|\psi\rangle_{|1\rangle} = |0\rangle$$

We then calculate

$$\begin{aligned}\rho_B &= \left(\frac{1}{\sqrt{3}} |\psi_{|0\rangle}\rangle \right) \left(\frac{1}{\sqrt{3}} \langle\psi|_{|0\rangle} \right) + \left(\frac{1}{\sqrt{3}} |\psi\rangle_{|1\rangle} \right) \left(\frac{1}{\sqrt{3}} \langle\psi|_{|1\rangle} \right) \\ &= \frac{1}{3}(|0\rangle \langle 0| + |0\rangle \langle 1| + |1\rangle \langle 0| + |1\rangle \langle 1|) + \frac{1}{3} |0\rangle \langle 0| \\ &= \frac{1}{3} \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \right) \\ &= \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \end{pmatrix}\end{aligned}$$

Theorem 2.3 (**The No Cloning Theorem**). *There exists no unitary transformation U such that given an* arbitrary *unknown state $|\psi\rangle$ such that $U |\psi\rangle |0\rangle = |\psi\rangle |\psi\rangle$*

Proof. Suppose such a unitary did exist, then we would have that under U,

$$(\alpha |0\rangle + \beta |1\rangle) \otimes |0\rangle \mapsto (\alpha |0\rangle + \beta |1\rangle) \otimes (\alpha |0\rangle + \beta |1\rangle)$$

But then this would imply that

$$U \begin{pmatrix} \alpha \\ \beta \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha^2 \\ \alpha\beta \\ \alpha\beta \\ \beta^2 \end{pmatrix}$$

Which isn't linear! ■

3. Some Quantum Protocols

3.1. **Quantum Money (Weisner Scheme).**

Quantum money, proposed by Weisner, gives a concept for money that is impossible to counterfeit, taking advantage of the No-Cloning Theorem. A quantum bill would then consist of the following

(1) A classical serial number, in the form of a bitstring.
(2) Some quantum serial number, given as a string of states $|\psi_i\rangle$.
(3) A central bank holding a database with a function $f(s)$ that takes classical serial number and maps it to a classical description of the corresponding quantum serial number.

The states making up the quantum serial number comprise of elements in the set $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$, which we recall are complement bases (maximally indistinguishable). When a transaction occurs, the money is then sent to the bank for verification, which takes the bill, consults the function, and measures in the corresponding bases. If the bill matches the description given by the function, then it is deemed valid. Of course, it's possible for fake bills that don't match to be deemed valid as well, but the probability that the entire string is measured correctly decays exponentially by a factor of $\left(\frac{3}{4}\right)^n$ where n is the length of the quantum serial number. That being said, there are some possible attacks on this scheme

Example 3.1 (**Interactive Attack**)**.** Assuming the bank returns the bill back after verification (whether it was correct or not), we have an interactive attack that goes qubit by qubit. Suppose we are given some $|\psi_1\rangle$ that we want to counterfeit. We could first create a qubit $|\phi\rangle = |0\rangle$, then send it to the bank. This may come back as either valid or invalid, but we know that if it came back as valid that $|\psi_1\rangle$ could not have possibly be $|1\rangle$, and likewise if it came back as invalid that $|\psi_1\rangle$ could not have possibly been $|0\rangle$. We repeat the process multiple times in order to deduce to arbitrary precision the valude of $|\psi\rangle$

But suppose the bank won't give back invalid bills, or worse- will immediately arrest you if a measurement goes wrong. Even in this case we have a strategy that works with the probability we get caught arbitrarily small

Example 3.2 (**Elitzur-Vaidman Bomb**)**.** Again given a $|\psi_1\rangle$ we want to copy, we first initialize a qubit $|c\rangle$ to $|0\rangle$. Then after we apply a rotation matrix R_ε to $|c\rangle$, giving us

$$R_\varepsilon |c\rangle = \cos\varepsilon |0\rangle + \sin\varepsilon |1\rangle$$

We then CNOT our qubit $|c\rangle$ with $|\psi_1\rangle$, with $|c\rangle$ as the control and $|\psi_1\rangle$ as the target.

We then send the tampered $|\psi_1\rangle$ back to the bank. The the bill gets rejected with prbability $(\sin^2\varepsilon)$, which we can make arbitrarily small by our choice of ε. We then repeat this process $\frac{\pi}{2\varepsilon}$ times, and after we are done, we measure $|c\rangle$. If $|\psi_1\rangle$ was prepared in the $|0\rangle, |1\rangle$ basis, we will have that $|c\rangle = |0\rangle$. However, if $|\psi_1\rangle$ was prepared in the $|+\rangle |-\rangle$ basis, we have that the CNOT operation leaves the two qubit state unchanged (i.e. $|0\rangle |+\rangle$ and $|0\rangle |-\rangle$ are eigenvectors of the CNOT operation with eigenvalue 1.) In which case after repeating process $\frac{\pi}{2\varepsilon}$ times $|c\rangle$ will be measured as $|1\rangle$.

Remark. We note the bank can foil this by simply returning another copy of the bill or an entirely new bill after verifying the bill, as the new bill would no longer be entangled with $|c\rangle$.

3.2. **BB84 Quantum Key Distribution.**
We know that if Alice wants to send Bob a bitstring s of length n, and they have a preshared bitstring k (a key) also of length n, then Alice can just send the bitstring $e = s \oplus k$ to Bob (s XOR k), and Bob can then apply $k \oplus e$ to recover Alice's original bitstring. This is already to proven to be unbreakable and secure, but the key k can only be used once, and it relies on Alice and Bob already having a shared key. As it turns out, there's a way for Alice and Bob to transmit qubits to each other securely such that they can created a shared key without having to meet in person, which can then be used as a one-time pad for a classical message.

To do this, Alice starts with two random bit strings s_A, b_A of length n. String s_A will tell us which state to put in our qubitstring, and string b_A will tell use which basis to use. We say that 0 corresponds to $|0\rangle$ or $|+\rangle$, and that 1 corresponds to $|1\rangle$ or $|-\rangle$, and in the bitstring b_A, 0 corresponds to the $|0\rangle, |1\rangle$ basis and 1 corresponds to the $|+\rangle, |-\rangle$ basis. For example, consider

$$s_A = 0\ 1\ 1\ 0$$
$$b_A = 1\ 1\ 0\ 1$$

This tells Alice to send the qubitstring

$$|\psi\rangle = |+\rangle\,|-\rangle\,|1\rangle\,|+\rangle$$

Bob then receives $|\psi\rangle$ via the quantum channel, and generates a random bitstring b_B and measures in the corresponding bases. For example suppose Bob randomly generates the bitstring

$$b_B = 0\ 1\ 0\ 1$$

Then measurement could possibly yield

$$|\psi\rangle_B = |0\rangle\,|-\rangle\,|1\rangle\,|1\rangle$$

In which case he only measures the correct qubit if he chose the same basis as Alice. Then over an authenticated classical channel, Alice and Bob compare their bitstrings b_A and b_B to determnine which bases in which they agreed. They then take the corresponding qubits to form their shared key. Of course, this might not be long enough for the message Alice wants to encode, but they can repeate the process to get a key of the appropriate length. But what if there's an eavesdropper Eve on the line? By the no-cloning theorem, the only thing Eve can do to try to learn the key for herself is to measure $|\psi\rangle$, but then this tampers with Alice's key. Then when Alice and Bob compare bases, they can also choose to discard part (usually half) of the shared key they have to ensure they are the same. If any of them differ they know that their key is compromised, and discard the key. In practice, there are also errors in transmitting the qubits, so some might differ even without an eavesdropper, in which case error correction need to be done, and Eve is only detectable if there is a noticable increase in the error rate.

3.3. **SARG04.**
This is another quantum key distribution protocol, but it's ore secure than BB84. In this case, Alice uses bitstrings s_A and b_A just like with BB84 to produce a qubitstring $|\psi\rangle$, which she then sends the to Bob. Like BB84, Bob then generates a random bitstring b_B and measures in the corresponding bases. After this however, Alice then sends Bob a string of pairs

$$|0/1\rangle\,|+/-\rangle$$

for each qubit in the string such that the corresponding qubit in $|\psi\rangle$ is in the pair. Bob can then can deduce in which qubits the measurement was not ambibuous, and sends Alice the locations of these bits. Then the bits in these positions form a shared key. Like BB84, they might have to repeat the process to get sufficiently many shared bits.

Example 3.3. Suppose Alice starts with the bitstrings

$$s_A = 0110101$$
$$b_A = 1001001$$

With which she creates the qubitstring

$$|\psi\rangle = |+\rangle\,|1\rangle\,|1\rangle\,|+\rangle\,|1\rangle\,|0\rangle\,|-\rangle$$

After sending this to Bob, he randomly generates a bitstring b_B, which might be something like

$$b_B = 1101010$$

Then after measuring, Bob might get something like

$$|+\rangle |+\rangle |1\rangle |+\rangle |1\rangle |-\rangle |0\rangle$$

Alice the sends her string of pairs, which might be something like

$$\{|0\rangle, |+\rangle\}, \{|1\rangle, |+\rangle\}, \{|1\rangle, |+\rangle\}, \{|1\rangle, |+\rangle\}, \{|-\rangle, |1\rangle\}, \{|+\rangle, |0\rangle\}, \{|0\rangle, |-\rangle\}$$

Bob then inspects his measured string with what Alice sends and determines if he can unambiguously deduce what Alice sent in $|\psi\rangle$. In this case

(1) Ambiguous– both can be measured as $|+\rangle$
(2) Ambiguous
(3) Ambiguous
(4) Ambiguous
(5) Ambiguous
(6) Bob knows that if Alice sent $|+\rangle$ here that he would have measured $|+\rangle$, however, he measured $|-\rangle$ so he deduces that the intended qubit was $|0\rangle$.
(7) Ambiguous

After this exchange, Bob tells Alice that he now knows what the 6^{th} bit was, so they get the shared bit 0.

As you can see, there's a high failure rate here, so it's possible that they have to do many exchanges to get sufficiently many shared bits

3.4. **Superdense Coding.**

We have that classically, n bits encodes no more than n bits of information, and this is true in the quantum sense too. If Alice sends Bob a qubit, once he measures it, it's just one bit. But in the case that Alice and Bob share entanglement usually in the form of a Bell pair

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

Then Alice can actually send Bob *two* bits of information. To do this, Alice can apply one of the following transformations to her qubit of the Bell pair

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

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad Z \cdot \text{NOT} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

One can verify that applying the transformations gives us 4 mutually orthongonal states, which are therefore perfectly disntinguishable, and can correspond to the 4 possible states of 2 classical bits. To do this, given two bits x, y that Alice wants to send, she follows the rules

(1) If $x = 1$, Alice applies NOT
(2) If $y = 1$, Alice applies Z

After applying the transformations, Alice her bit to Bob so he holds both qubits of the entangled pair. Bob the applies the transformation

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

One can check that under the transformation, based on what Bob was sent, he gets the following after measurement of both qubits in the $|0\rangle, |1\rangle$ basis

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \mapsto |00\rangle$$

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

$$\frac{|01\rangle + |10\rangle}{\sqrt{2}} \mapsto |10\rangle$$

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

Quantum Teleportation.
Say Alice has some qubit

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

and Alice wants to "send" (though in reality Bob is just recreating $|\psi\rangle$, and not receiving the actual physical qubit) $|\psi\rangle$ to Bob using only classical communication and a shared Bell pair. The starting 3 qubit system is then given by

$$a|0\rangle + b|1\rangle \otimes \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Alice first CNOT's $|\psi\rangle$ with her half of the Bell pair, giving us the resulting state

$$\frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|110\rangle + b|101\rangle)$$

We then apply the Hadamard (H) gate to $|\psi\rangle$, giving us

$$\frac{1}{2}(a|000\rangle + a|100\rangle + a|011\rangle + a|111\rangle + b|010\rangle - b|110\rangle + b|001\rangle - b|101\rangle)$$

Then Alice measures both in the $|0\rangle, |1\rangle$ basis and the measurements then tell her the state of Bob's half of the Bell pair, given by the following

$$\begin{aligned}
|00\rangle &\implies a|0\rangle + b|1\rangle \\
|01\rangle &\implies a|1\rangle + b|0\rangle \\
|10\rangle &\implies a|0\rangle - b|1\rangle \\
|11\rangle &\implies a|1\rangle - b|0\rangle
\end{aligned}$$

Alice then sends two classical bits to Bob describing her measurements, and Bob can then apply the appropriate transformation to recreate $|\psi\rangle$ at his location. We note that this actually destroys Alice's half of the Bell pair, as well as destoying her copy of $|\psi\rangle$ so this doesn't violate the No-Cloning Theorem. In addition, we note that Bob's copy of $ket\psi$ is actually his half of the Bell pair, so the entire Bell pair is destroyed in the process. This actually gives us a way of entangling two qubits without them actually interacting, by simply sending an entangled qubit to Bob, thus entangling his half of the Bell pair. If we start with the state where $|B\rangle_A$ and $|B\rangle_B$ correspond to Alice and Bob's halves of the Bell pair, and $|C\rangle$ and $|D\rangle$ and are entangled (note that anyone can hold $|D\rangle$, not necessarily Alice or Bob), with Alice holding $|C\rangle$, we can pictorially represent this as

$$|B\rangle_A \longrightarrow |B\rangle_B$$

$$|C\rangle \longrightarrow |D\rangle$$

After teleportation we have that the Bell pair is destroyed, giving us

$$|C\rangle \downarrow |D\rangle$$

So Bob now holds a qubit entangled with $|D\rangle$.

So far we've only discussed entanglement with 2 qubits, but we can do more. Consider the GHZ-state, with Alice, Bob, and Charlie each holding one part of the state.

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

However, if Alice and Bob get together without Charlie, they find that their bits are just classically correlated, with probability 0.5 that they have $|00\rangle$ and probablity 0.5 that they have $|11\rangle$. THis illustrates the concept of **monogamy of entanglement** if Alice has a qubit maximally entangled with Bob, then the same qubit cannot also be maximally entangled with Charlie.

4. Entropy

The last two protocols utilize entangled Bell pairs as a sort of resource, so we'd probably want to be able to quantify exactly how much entanglement there is in a given state. We first start with a classical construction

Definition 4.1. For a probability vector $v \in \mathbb{R}^n$ the **Shannon Entropy** of v, denoted $S(v)$ is given by the formula

$$S(v) = \sum_{i=1}^{n} v_i \log_2 \frac{1}{v_i}$$

Definition 4.2. For a mixed state given by the density matrix ρ, the **Von Neumann Entropy** of ρ is given by the formula

$$H(\rho) = \sum_{i=0}^{n} \lambda_i \log_2 \frac{1}{\lambda_i}$$

Where λ_i are the eigenvalues of ρ. Note that this is just the Shannon Entropy of the eigenvalues.

Remark 4.3. The Von Neumann Entropy of any pure state $|\psi\rangle \langle\psi|$ is 0, and is invariant under unitary transformation i.e.

$$H(\rho) = H(U\rho U^\dagger)$$

We can then use the Von Neumann Entropy to quantify entanglement

Definition 4.4. Given some entangled state $|\psi\rangle$, with reduced density matrices ρ_1 and ρ_2, we define the **Entanglement Entropy** of $|\psi\rangle$ to be

$$H(\rho_1) = H(\rho_2)$$

Note in order for this to be well defined, the Von Neumann Entropy of the two reduced density matrices must always be the same, which is true.

Theorem 4.5. *A bipartite state*

$$|\psi\rangle = \sum_j \sum_k |j\rangle |k\rangle$$

always be put in ***Schmidt Form***

$$|\psi\rangle = \sum_i |v_i\rangle |w_i\rangle$$

For orthogonal bases $\{|v_i\rangle\}$ *and* $\{|w_i\rangle\}$

Example 4.6. The state

$$\frac{|0\rangle|+\rangle + |0\rangle|-\rangle + |1\rangle|+\rangle}{\sqrt{3}}$$

is not in Schmidt Form because it contains the "cross terms" $|0\rangle|-\rangle$ and $|1\rangle|+\rangle$. However, the state

$$\frac{3|0\rangle|+\rangle + 4|1\rangle|-\rangle}{5}$$

is in Schmidt Form.

Alternatively, if we can reduce $|\psi\rangle$ into **Schmidt Form**, i.e.

$$\sum_i \lambda_i |v_i\rangle |w_i\rangle$$

we can just compute the Shannon Entropy of the probability vector with the λ_i as entries.

We then note that if we calculate the entanglement entropy of the Bell pair

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

We know that the reduced density matrices are both $\frac{I}{2}$, so we calculate

$$H\left(\frac{I}{2}\right) = \frac{1}{2}\log_2 2 + \frac{1}{2}\log_2 2 = 1$$

So the entanglement entropy tells us "how many Bell Pairs of entanglement we have."

5. Quantum Computing Basics

First, some basis tenets of quantum computing

(1) An exponential speed up is the best we can wish for. Quantum computing won't be able to solve problems already proven to be uncomputable (e.g. halting problem).
(2) We should try to use only a polynomial number of gates, each acting on a small number of qubits.
(3) We should take advantage of interference. Classical probabilities don't cancel, but quantum amplitudes do. We should use this to our advantage to get the wrong answers to destructively interfere, and the right answers to interfere constructively.
(4) Entanglement is crucial in taking full advantage of quantum mechanics. Consider an unentangled of n-qubits. This can be written as the tensor product

$$(\alpha_1|0\rangle + \beta_1|0\rangle) \otimes \ldots \otimes (\alpha_n|0\rangle + \beta_n|1\rangle)$$

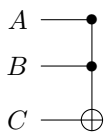
Which requires only $2n$ complex numbers, which can be represented with $4n$ real numbers. This is just $\mathcal{O}(n)$. On the other hand, and entangled state of n-qubits could be written as

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

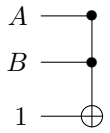
Which has 2^n amplitudes.

5.1. Universal Gate Sets.

Classically, we know that the NAND gate is a universal gate for classical computing. Every logic gate can be expressed as a circuit of NAND gates. Another example of a classically universal gate is the **Toffoli gate** (sometimes referred to as the controlled-controlled-NOT). The best way to illustrate what this does is with a circuit diagram



In other words, a Tofolli gate flips the C bit if $A = B = 1$. To show this is universal, consider



This outputs $\mathsf{NAND}(A, B)$ in the C slot, so this is universal. Note that Tofolli is actually a quantum gate, not just a classical one, since it has the property that it is *reversible*. If we have the output of a Toffoli, we also know what the input is. This is going to be very important when it comes to quantum algorithms. Note that this also proves that quantum computers can do everything a classical computer can do as well. We can do everything with just Toffoli gates. In the quantum case, the definition for what a universal gate set is gets a bit more complex

Definition 5.1. A **universal gate set** is a set of unitary matrices that can be used to approximate any unitary to any desired precision.

We're being a bit vague here, but for our purposes, this is enough. There's also a cool theorem- you probably don't have to know this for now, but it's nice to know

Theorem 5.2 (**Solovay-Kitaev**)**.** *With any universal gate set, we can approximate an n-qubit unitary matrix with precision ε with a $\mathcal{O}\left(polylog\left(\frac{1}{\varepsilon}\right)\right)$ number of gates.*

With that out of the way, how do we know if a gate set is universal? We're actually going to postpone this for a minute. Let's first discuss some red flags as to when a gate set won't be universal

Your Gate Set Probably Isn't Universal If . . .

(1) No interference. For example, $\{\mathsf{CNOT}\}$ is not universal, since all it does is swap amplitudes around. You cannot create interference with CNOT alone.
(2) No entanglement.
(3) Only $\mathbb{R}$-valued gates
(4) Only a stabilizer set (We'll come back to this later). For now, an example of a stabilizer set is the set

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

With that said, here are some examples of universal gate sets

$$\left\{\mathsf{CNOT}, R_{\frac{\pi}{8}}, P\right\}$$
$$\{\text{Toffoli}, H, P\}$$

In addition, it turns out that "almost every" two qubit gate alone will be universal. For the more mathematically inclined, the more precise way to say this is that the set of non-universal 2-qubit gates is a set of measure 0.

5.2. **Complexity.** The next order of business is how we measure the complexity of our algorithms. There are two options here

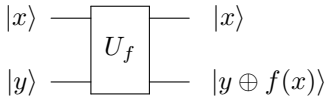
(1) **Circuit Complexity** - The circuit complexity of an arbitrary unitary is the size of the smallest circuit that implements it using only gates from some specified universal set

Remark. This actually proves quite difficult to find/prove that a given circuit is the minimal one, so there's a better way to do this. That being said, we're going to have a heavy preference for unitaries that have a polynomial circuit complexity (recall one of our basic tenets).

(2) **Query Complexity** - We measure the complexity of our algorithm with regards to how many times we have to call an oracle\black box.

The best way to think of this black box is just a unitary that spits out the value of some function, where we have no idea what is going on (hence the name "black box"). A classical black box might take in a bit x, and map via some rule $x \mapsto f(x)$. For the quantum case, you might want to just have $|x\rangle \mapsto |f(x)\rangle$, but

there's a problem here, since f need not be a function represented by a unitary matrix. To do this, we'll need an ancilla quibit to store the answer. We can think of this as given by the circuit



If we prepare $|y\rangle$ as $|0\rangle$, this is a lot like CNOT-ing the answer in to our ancilla bit. There's another trick though. If we prepare $|y\rangle$ as $|-\rangle$, if we run it through our circuit, our utput with be the state

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

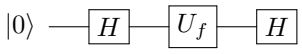
We usually forget about the extra $-$, and say that the oracle just outputs $(-1)^{f(x)} |x\rangle$. This trick is referred to as *phase kickback*, and we'll sometimes call U_f a *phase oracle*.

5.3. **Quantum Algorithms.** We actually know enough to see our first quantum algorithm.

Example 5.3 (**Deutsch's Algorithm**)**. The problem:** We have two unknown bits b_0, b_1, and an oracle $f : \{0,1\} \to \{0,1\}$ where $f(x) = b_x$. We want to find the paritry of these two bits, i.e. $b_0 \oplus b_1$.

Remark. Note that classically, we're going to need 2 queries to f. We're going to need to know both b_0 and b_1. Therefore, the classical query complexity is 2.

The question is, can we do better quantumly? Yes! We're going to be able to do this with just a single query! Consider the circuit (ignoring the ancilla $|-\rangle$ qubit and letting U_f denote our phase oracle).



What does this do? We first apply H to $|0\rangle$, giving us the state

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

Applying U_f, this gives us the state

$$\frac{(-1)^{f_0} |0\rangle + (-1)^{f(1)} |1\rangle}{\sqrt{2}}$$

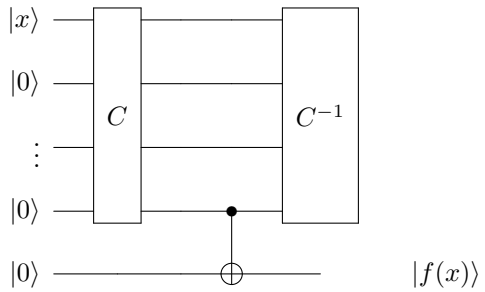
$$= \frac{(-1)^{b_0} |0\rangle + (-1)^{b_1} |1\rangle}{\sqrt{2}}$$

$$= (-1)^{b_0} \frac{|0\rangle + (-1)^{b_0 - b_1} |1\rangle}{\sqrt{2}}$$

$$= \begin{cases} |+\rangle & \text{if } b_0 = b_1 \\ |-\rangle & \text{if } b_0 \neq b_1 \end{cases}$$

Then Hadamarding and measuring will give us $|0\rangle$ if the parity is even and $|1\rangle$ otherwise.

Recall that all unitary matrices are invertible, and that we can think of quantum circuits as just large unitaries acting on a large number of qubits. This introduces a big limitation on what we can do, since all computations must be invertible. This is why we often us ancilla qubits to store answers. Another problems is that intermediate computations can produce *garbage* which might interfere with our other qubits, and either ruin some interference we're trying to exploit, or introduce unwanted interference. How do we deal with this? **Uncomputing**. Consider some computation we do on $|x\rangle$ with some ancilla $|0\rangle$ qubits, we will represent the computation with the unitary C. This might produce some garbage.

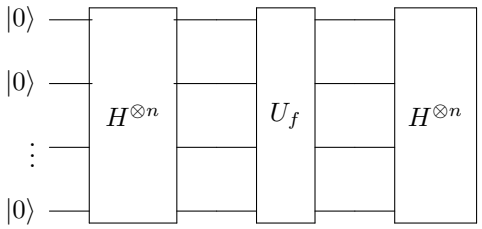


We can just CNOT the answer into another ancilla and do the entire circuit in reverse. This gets rid of all the garbage, and we still have the answer stored away.

Another algorithm,

Example 5.4 (**Deutsch-Jozsa**)**.**
The Problem: Suppose we have a function $f : \{0,1\}^n \to \{0,1\}$, and we are promised that f is either constant or balanced (i.e. half of the outputs are 0 and the other half are 1). We want to determine which. Classically, we're going to need to check $\frac{1}{2}$ of the possible inputs $+1$ to determine the answer with certainty. However, with random sampling we can usually determine the answer with high probability. Nevertheless, quantumly, we can find the solution with certainty using a single query.

The first move, which will become quite familiar, is to Hadamard all the qubits. For notational convenience, let $H^{\otimes n}$ denote the tensor product of n Hadamard gates. The algorithm is the given by the circuit



Note that this essentially an n qubit version of the circuit for Deutsch's Algorithm. Now let's check that the circuit works! For notational convenience again, for bitstrings $s, x \in \{0,1\}^n$, let

$$s \cdot x = \sum_{i=0}^{n-1} s_i \cdot x_i \bmod 2$$

Which is pretty much the standard dot product, but over the finite field $\mathbb{F}_2$. Let's run through the computations

$$H|0\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

$$\text{Applying } U_f \text{ we get } \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle$$

$$\text{Hadamarding everything then givs us} \frac{1}{\sqrt{2}} \sum_{x\in\{0,1\}^n} \sum_{y\in\{0,1\}^n} (-1)^{f(x)}(-1)^{x\cdot y} |y\rangle$$

We then measure all n qubits in the $\{|0\rangle, |1\rangle\}$ basis. If we measure $|0\rangle$, we have that f is balanced, and we measure $|1\rangle$ otherwise.

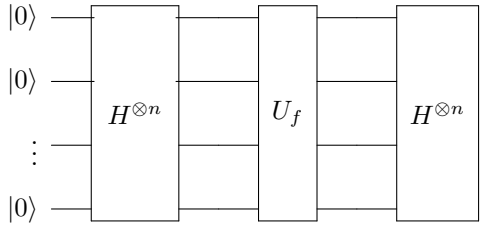
The last two quantum algorithms aren't too impressive in terms of speed-ups. While Deutsch-Jozsa technically is an exponential speed-up, probabalistically the problem isn't very hard. The next algorithm is a better example of how an quantum algorithm can provide a serious improvement over classical computation, sometimes referred to as "quantum supremacy." While the problem seems quite artificial, it's still important in showing that quantum computers can do something that is proven classically hard much faster.

Remark. A useful identity when it comes to Hadamard gates,

$$H^{\otimes n}|x_1 \dots x_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{s \cdot x} |x\rangle$$

Example 5.5 (**Bernstein Vazirani**).
The Problem: Given a function $f : \{0,1\}^n \to \{0,1\}$, we are given that for some bitstring $s \in \{0,1\}^n$, $f(x) = s \cdot x$. We're tasked with determining s. Classically, this will need n queries, where we go bit by bit feeding in an input with all 0 except for 1 in the i^{th} position to determine that i^{th} bit of s. Quantumly, this will be *one query*.

The solution to this problem is given again by the *exact same circuit*.



We note that the first two gates will map

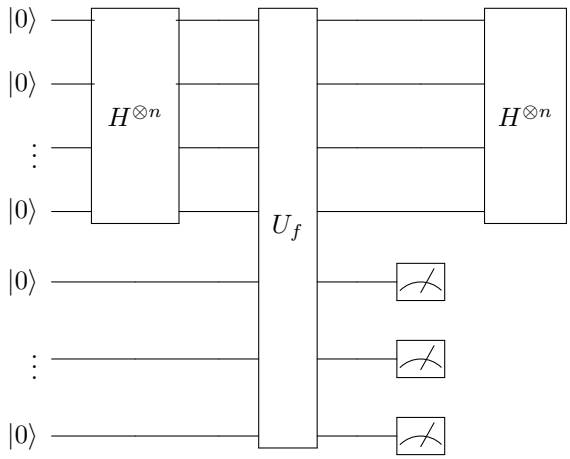
$$|0 \dots 0\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{s \cdot x} |x\rangle$$

Noting the identity, applying $H^{\otimes n}$ to this state yields us $|s\rangle$.

The next algorithm will be slightly more interesting, because it will have some important concepts that will be revisited in Shor's Algorithm- the famous quantum algorithm that will destroy modern cryptography with the ability to factor primes efficiently. That being said, we should do the more elementary algorithm first

Example 5.6 (**Simon's Problem**).
The Problem: Given a function $f : \{0,1\} \to \{0,1\}$, we are given a nonzero bitstring $s \in \{0,1\}^n$ such that $\forall x, y \in \{0,1\}^n, f(x) = f(y) \iff y = x \oplus s$. We are again tasked with finding the secret bitstring s. Note that if we ever find a collision, i.e. bitstrings x, y such that $f(x) = f(y)$, we can recover s but simply taking $x \oplus y$. Classically, we need $2^{n-1} + 1$ queries to find the answer for certain, but we're likely to find a collision with $2^{\frac{n}{2}}$ queries (this is a lot like the birthday "paradox"). Either way, this is an exponentially hard problem for classical computers. The question remains: What can we do quantumly?

The circuit for this problem is going to be slightly different, but more of the same. This time, we're not going to use a phase oracle, but an oracle U_f that maps $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$.



Note that the last few qubits that end up being measured are the answer register. Let's analyze what this circuit is doing. The first part is nothing new. We Hadamard everything to get an equal superposition of all the possible inputs and then store the answers in the answer register. When we measure the answer register $|f(x)\rangle$, by the partial measurement rule, the qubits in the input register collapse to an equal superposition over all possible inputs consistent with what we measured. That is to say, the input register is now an equal superposition

$$\frac{|x\rangle + |x \oplus s\rangle}{\sqrt{2}}$$

For some bitstring x where $f(x)$ is what we measured in the answer register. Then

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

Note that we get constructive interference when $x \cdot z = (x \oplus s) \cdot z$. Otherwise, the amplitudes cancel. In other words, when we measure all the qubits in the $\{|0\rangle, |1\rangle\}$ basis, we measure $|z\rangle$ if and only know $x \cdot z = (x \oplus s) \cdot z$. Noting that $\cdot$ is bilinear, we get

$$\begin{aligned} & x \cdot z = x \cdot z \oplus s \cdot z \\ \implies & s \cdot z = 0 \end{aligned}$$

This gives us one linear equation over $\mathbb{F}_2$. Repeating this process, we get a system of equations $s_i \cdot z_i = 0$. Once we find enough linearly independent equations, we can just solve this using basic linear algebra (row reduction/matrix inversion, etc). One can verify that we will find enough linearly independent equations with high probability just by computing the cardinality of the vector space $\mathbb{F}_2^n$, and finding the cardinality of the subspaces we're looking for.

5.4. **Shor's Algorithm.** We now arrive at the famed prime factoring algorithm. The problem is: Given a composite number $N = pq$ for primes p, q, we want to recover the factors p and q. Since we can check if 2 is a factor by inspection, we're going to assume both p and q are prime. First, we're going to note how we're going to measure complexity in this problem. Note that we can just try all the numbers from 3 to $\frac{N}{2}$, and we'll be able to recover the factors in $\mathcal{O}(N)$ time classically. Instead, we're going let n denote the number of bits needed to store N. Then this becomes a much harder problem classically (we think).

Remark. As famous as the algorithm is, it's not provably faster than a classical computer, since we haven't proven the complexity of prime factorization for classical computers. As of now, we know that we currently can't do better than exponential complexity in n, but that doesn't mean that we can't discover a polynomial algorithm later down the line.

The idea of Shor's Algorithm is to reduce prime factorization to a problem of **Period Finding**. Given a function $f : \mathbb{N} \to \mathbb{N}$, we are promised that there exists some $s > 0$ such that $\forall x, y \in \mathbb{N}$, $f(x) = f(y) \iff s | x - y$ i.e. $y = x + ks$ for some integer k. How is this reduction going to work? Consider the multiplicative group mod N $\mathbb{Z}/N\mathbb{Z}$. Since it is the product of two primes, we can use a formula from Euler's Totient Function φ to find that $|\mathbb{Z}/N\mathbb{Z}| = (p-1)(q-1)$. Then we pick some x such that $\gcd(x, N) = 1$. We then define f where $f(r) = x^r \bmod N$. We note that this function is computable in polynomial time using repeated squaring. Also note that f will be a periodic function. What will knowing the period of f do to help us recover factors? Let's assume we know the period s, and we'll also make the assumption that s is even. Then

$$\begin{aligned} & x^s = 1 \bmod N \\ \implies & x^s - 1 = 0 \bmod N \\ \implies & \left(x^{\frac{s}{2}} + 1\right)\left(x^{\frac{s}{2}} - 1\right) = 0 \bmod N \end{aligned}$$

Then assume that neither $a_1 = \left(x^{\frac{s}{2}} + 1\right)$ nor $a_2 = \left(x^{\frac{s}{2}} - 1\right)$ is a multiple of N. Then we can recover the factors using $\gcd(a_1, N)$ and $\gcd(a_2, N)$, and gcd is computable in polynomial time as well using the Euclidean Algorithm. What happens if our chosen x doesn't happen to satisfy these properties? We'll just try again. I'll state without proof that we can find such an x with approximately $\frac{3}{8}$ probability. Now we've reduced the factoring problem to finding the order of the function f that we've created.

Having reduce the problem, the question remains: How do we find the period of the function f? First, choose $Q > N$ such that $Q \approx N^2$. For simplicity, we're just going to pick the fist power of 2 greater than N^2. Then using Hadamard's and our function, we can prepare the state

$$\frac{1}{\sqrt{2}} \sum_{r=0}^{Q-1} |r\rangle |f(r)\rangle$$

Then if we measure the answer register, the input register collapses into the answers consistent with what answer we measured. This will the the superposition (with some normalization factor)

$$\frac{1}{\sqrt{L}} (|r\rangle + |r+s\rangle + |r+2s\rangle + \ldots)$$

Then we're going to apply a Fourier transform. Let

$$\omega = e^{\frac{2\pi i}{Q}}$$

Then the Quantum Fourier Transform of dimension Q is the unitary matrix F where

$$F_{Q_{jk}} = \omega^{jk}$$

This can be implemented with a polynomial number of gates using a method similar to the classical Fast Fourier Transform. We then apply F_Q to our state, giving us

$$\frac{1}{\sqrt{QL}} \sum_{j=0}^{Q_1} \sum_{\ell=0}^{L-1} \omega^{(r+\ell s)j} |j\rangle$$

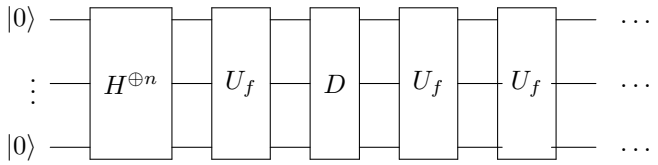
and then measure the qubits. In the case that $s \mid Q$ and, we get that we measure $|j\rangle$ only when js is a multiple of Q. Then $j = \frac{KQ}{s}$. If we repeate the process, we can find multiple j_i, and then taking their gcd will give us $\frac{Q}{s}$, which we can use to find s, since we know Q.

In the case where s does not divide Q, then $\frac{Q}{s}$ is no longer and integer, but the Fourier Transform will ensure that we measure a nearest integer to $\frac{Q}{s}$ with high probability. We can then use a continued fraction algorithm to find some multiple of $\frac{Q}{s}$. Using continued fractions, we can find $\frac{k}{s}$ for some integer k. If we repeat this process multiple times and reduce each $\frac{k_i}{s_i}$ into lowest terms, we can take the least common multiple of their denominators to find s with high probability.

5.5. **Grover's Algorithm.** **The Problem:** Given a function $f : \{1, \ldots, N\} \to \{0, 1\}$,

(1) Does there exist an x such that $f(x) = 1$?
(2) If so, find an x such that $f(x) = 1$.

You can think of this as an unstructured search problem. For example, given some unorganized database, we might want to see if some object is in the database, and if so, to find it. This also gives a way to solve your favorite NP-complete problem by simply performing this search over all possible inputs. Classically, we're going to need $\mathcal{O}(N)$ queries, because we'll never know if the one we didn't check was the x we were looking for, so we'll need to check them all. However with Grover's Algorithm, we're going to be able to do this with $\mathcal{O}(\sqrt{N})$ queries. We're going to make some simplifying assumptions here. first, we're going to assume that there is a unique x^* such that $f(x^*) = 1$. We'll call this element the *marked item*. The case with multiple marked items will be similar. Also, we're going to assume that $N = 2^n$ for some n. If not, we can just tack on extra elements onto our domain that evaluate to 0. Let's draw out the circuit now. We're going to use a phase oracle in this problem



You might be wondering what this D operator is. It's referred to as the *Grover Diffusion Operator*, and its action is usually referred to as *inversion about the average*. We'll see what this means in a moment. As

a $2^n \times 2^n$ matrix, D is given by

$$D_{ij} = \begin{cases} \frac{2}{N} - 1 & i = j \\ \frac{2}{N} & i \neq j \end{cases}$$

For an arbitrary state $|\psi\rangle = \sum_{x=0}^{N-1} \alpha_x |x\rangle$, let μ denote the average amplitude, i.e.

$$\mu = \frac{\sum_{x=0}^{N-1} \alpha_x}{N}$$

Then when D acts on $|\psi\rangle$, the amplitude α_x is "flipped about the mean", i.e. $\alpha_x \mapsto (2\mu - \alpha_x)$. Geometrically, you can think of the iterated process given by the circuit as fixing a 2 dimensional subspace and rotating our given state towards the basis state specified by the marked item. Then after about approximately $\frac{\pi\sqrt{N}}{4}$ iterations, we should measure our desired state with high probability. Note that we need to be careful with how many iterations we do this. If we run it too many times, we're going to overshoot our target, and keep rotating past. However, if we miss we can keep querying until we come back around again. The success probability is approximately a sinusoid in the number of queries. In the case where there are k marked items, we're going to need $\mathcal{O}\left(\sqrt{\frac{N}{K}}\right)$ queries to find any number of them.

It's been proven that Grover's algorithm is optimal. This is a bit unfortunate, since it erases any hopes of any brute force quantum algorithm to solve NP-complete problems without exploiting some specific structure of the problem itself. That being said, a polynomial speed up is still nothing to scoff at.

6. A Brief Aside into Quantum Complexity Theory

First, some classical concepts in complexity theory

(1) P is the set of decision problems solvable in polynomial time
(2) NP is the set of decision problems where potential answers can be verified in polynomial time
(3) NP-hard problems are problems in which a solution to this problem would give us a solution to any other NP problem
(4) NP-complete problems are those that are both NP and NP-complete.

The quantum complexity class we care the most about at the moment is BQP, otherwise know as **Bounded Error Quantum Polynomial Time**, which is the quantum analogue of P. At the moment, relations between BQP and the other complexity classes are not that well known- we often don't know the true relations between classical classes either! What we do know is that $\mathsf{P} \subset \mathsf{BQP}$ and that factoring is in BQP. We also know that $\mathsf{BQP} \subset \mathsf{EXP}$ and that $\mathsf{BQP} \subset \mathsf{PSPACE}$.

7. Hamiltonians and the Adiabatic Algorithm

It's a strange time to first be introducing Hamiltonians, but since we're doing computer science, most of the things we do are discrete rather than continuous, so we've mostly ignored Hamiltonians. For our intents in purposes, a **Hamiltonian** $\mathcal{H}$ is just a Hermitian matrix. Recall that that this means $\mathcal{H} = \mathcal{H}^\dagger$.

The Shrödinger Equation is given by the formula

$$i\hbar \frac{\partial |\psi\rangle}{\partial t} = \mathcal{H} |\psi\rangle$$

We don't really care about the Planck constant $\hbar$, so we'll just rescale to natural units where $\hbar = 1$. This gives us the differential equation

$$i \frac{\partial |\psi\rangle}{\partial t} = \mathcal{H} |\psi\rangle$$

This is a system of differential equations (one for each component of $|\psi\rangle$). For those who know some differential equations, the solution to this is given by the *matrix exponential*

$$|\psi(t)\rangle = e^{-i\mathcal{H}t} |\psi(0)\rangle$$

One might wonder what it means to take e to the power of some matrix. In this case, we just use the power series definition of the exponential. Given some square matrix A, let

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

While not immediately obvious from the definition, this series converges. That being said, it's not very clear how one would actually go about computing the damn thing! If our matrix is diagonalizable (which in most cases it is), we have a handy trick. First note that if A is a diagonal matrix, then e^A is a diagonal matrix where

$$e^A_{ij} = \begin{cases} e^{A_{ij}} & i = j \\ 0 & i \neq j \end{cases}$$

Then for some diagonalizable matrix A, we know $A = UDU^{-1}$ for some diagonal matrix D. Then

$$e^A = \sum_{k=0}^{\infty} \frac{\left(UDU^{-1}\right)^k}{k!} = U\left(\sum_{k=0}^{\infty} \frac{D^k}{k!}\right)U^{-1} = Ue^DU^{-1}$$

So given a matrix, diagonalize it, exponentiate the diagonal matrix, and put everything back together to find the matrix exponential.

We now claim that for any Hermitian matrix $\mathcal{H}$, we have that $e^{i\mathcal{H}t}$ is a unitary matrix. Why must this be the case? First, we're going to claim that all the eigenvalues of $\mathcal{H}$ are real. To see why this is true, let $|v\rangle$ be an eigenvector of $\mathcal{H}$ with eigenvalue λ. Then we know

$$\langle v| \mathcal{H} |v\rangle = \lambda$$
$$\langle v| \mathcal{H}^\dagger |v\rangle = \overline{\lambda}$$

Noting that $\mathcal{H}^\dagger = \mathcal{H}$, this gives us that $\overline{\lambda} = \lambda$, which implies that $\lambda \in \mathbb{R}$. One fact that we'll use without proof is that any Hermitian matrix is diagonalizable via unitaries, i.e.

$$\mathcal{H} = UDU^\dagger$$

For some unitary matrix U. Then when we compute the matrix exponential, note that e^{itD} will be a unitary matrix since all the eigenvalues of $\mathcal{H}$, which are the elements on the diagonal of D are real. Therefore, $UDU^\dagger$ is also unitary, being a product of unitary matrices. Note that any eigenvector of $\mathcal{H}$ with eigenvalue λ will also be an eigenvector of $e^{i\mathcal{H}t}$ with eigenvalue $e^{i\lambda t}$, but the converse need not be true.

Definition 7.1. Given a Hamiltonian $\mathcal{H}$, let $\{\lambda_i\}$ be the set of eigenvalues, which we call **energies** ordered from smallest to largest and let $\{|v_i\rangle\}$ be the set of corresponding eigenvectors. Then the smallest energy λ_1 is called the **ground state energy** with $|v_1\rangle$ being the **ground state**. The other energies λ_i with eigenvectors $|v_i\rangle$ are then called the i^{th}-excited energy/state respectively.

Given two Hermitian matrices A, B, we know their sum $A + B$ is also a Hermitian matrix. One might think that $e^{A+B} = e^Ae^B$, like it is with complex numbers, but unfortunately, this isn't true in general. However, it is true when A and B commute. One can interpret e^{A+B} as two Hamiltonians acting of a system simultaneously. However, there is a way for us to understand what e^{A+B} is like without having to compute the exponential, which is helpful if both A and B are very complicated matrices. If we want to simulate e^{A+B} to some precision ε, we can use the formula

$$e^{A+B} \approx \underbrace{e^{\varepsilon A}e^{\varepsilon B}e^{\varepsilon A}e^{\varepsilon B}\ldots}_{\frac{1}{\varepsilon} \text{ times}}$$

This process is called *Trotterization*. You can think of it as turning on the Hamiltonian A for a very short period of time, turning it off, turning on B for a short amount of time, and so on.

The reason we're talking about Hamiltonians is because they provide a potential way to encode answers to problems, which is the motivation behind the adiabatic algorithm we're going to discuss. As an example, we're going to consider the 3-SAT problem.

Example 7.2 (**3-SAT**).
The Problem: Given a finite collection of clauses $(x_i \vee x_j \vee x_k)$, we want to know if there exists assignment of true or false to every x_i such that all the clauses evaluate to true.

We want to encode the answer to this problem in the ground state of some Hamiltonian $\mathcal{H}$. To do this, lets first establish how we're going to represent this problem as a quantum state. Let $\{x_i\}$ be the set of all literals that appear in our collection of clauses, and let n denote the cardinality of this set. Then we can represent an assignment of literals $f : \{1 \dots n\} \to \{0, 1\}$

$$f(i) = \begin{cases} 1 & x_i \text{ is assigned true} \\ 0 & \text{otherwise} \end{cases}$$

with the vector $|\psi_f\rangle = |f(1), f(2), \dots f(n)\rangle$. Let C_i denote the i^{th} clause. Then we can define define a Hamiltonian H_i acting on the three qubits x, y, z, that compose this clause where

$$H_i |xyz\rangle = \begin{cases} |xyz\rangle & \text{clause } i \text{ is unsatisfiable} \\ 0 & \text{otherwise} \end{cases}$$

In other words, $|xyz\rangle$ is an eigenvector with energy 1 if C_i is unsatsifiable and with energy 0 if C_i is satsifiable. Then let $\mathcal{H} = \sum H_i$. Then if our collection of clauses is unsatisfiable, the ground state will be an eigenvector with eigenvalue 0 encoding the solution to our problem.

Now we can discuss the **Adiabatic Algorithm**. First, lets discuss the theorem that inspired the algorithm. The actual theorem is somewhat technical, so we'll be a bit vague.

Theorem 7.3 (**Adiabatic Theorem**). *Given the ground state for some Hamiltonian $\mathcal{H}_i$, and $\mathcal{H}_\rangle$ is continuously changed to another Hamiltonian $\mathcal{H}_f$. Then, provided the transition was sufficiently slow, the state will end in the ground state of $\mathcal{H}_f$.*

Given this information, it's pretty easy to see what the Adiabatic Algorithm will do. We first design some Hamiltonian $\mathcal{H}_f$ that encodes the answer to the problem, and prepare another Hamiltonian $\mathcal{H}_\rangle$ that is relatively simple and has a ground state we can easily prepare. Then we continuously change $\mathcal{H}_i$ to $\mathcal{H}_\{$, and then measure our ground state the find the answer to our problem. While this sounds simple, there are some problems. For example, one of the main questions is just what is "sufficiently slow?"

Let $\mathcal{H}_t$ denote the Hamiltonian we have at time t. WLOG take $t \in [0, 1]$, with $\mathcal{H}_0 = \mathcal{H}_i$ and $\mathcal{H}_1 = \mathcal{H}_f$. Then let $\lambda_0(t)$ be the ground state energy of $\mathcal{H}_t$ and $\lambda_1(t)$ the 1^{st} excited state energy. Then let

$$g = \min_{t \in [0,1]} |\lambda_0(t) - \lambda_1(t)|$$

Then for the transition to be sufficiently slow, the time needed to run the Adiabatic Algorithm is $\sim \mathcal{O}\left(\frac{1}{g^2}\right)$. Note that if $g = 0$, then the algorithm won't even work, since we won't be able to avoid the ground state jumping to the first excited state.

8. Quantum Error Correction

First, let's talk about Classical Error Correction. In the early days of computing, there were questions regarding what to do if certain bits were corrupted due to noisy signals or some other factors. This might lead to a bit reading 0 to be flipped to 1. Their solution was to encode a logical bit at 3 bits, and using the logical bits as the bits in a computation. The logical bits $\bar{0}$ and $\bar{1}$ are encoded where

$$\bar{0} = 000$$
$$\bar{1} = 111$$

Then to detect an error, we just need to check if all three bits are the same. To correct the error, we change the flipped bit to the value of the majority of the bits. This will correct any single bit flip. The reason this works is because the set of bitstrings we can reach from 000 and 111 via single bit flips are disjoint.

Qunatumly, things get a bit more complicated. First of all, we have an entire continuum of possible errors. For example, if we had the state $|0\rangle$, this might rotate a little bit to the state $\sqrt{1-\varepsilon^2}\,|0\rangle + \varepsilon\,|1\rangle$. Fortunately this one has a relatively simple solution. If we keep measuring the qubit, it will collapse to $|0\rangle$ with high probability. Another more troubling problem is the fact that bit flips aren't the only problem we need to deal with. There also is the problem of a phase flip, where the relative phase between $|0\rangle$ and $|1\rangle$ change. To illustrate the problem, suppose we adopted the classical approach where

$$|\bar{0}\rangle = |000\rangle$$
$$|\bar{1}\rangle = |111\rangle$$

Note that in this encoding,

$$|\bar{+}\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}}$$
$$|\bar{-}\rangle = \frac{|000\rangle - |111\rangle}{\sqrt{2}}$$

Then a phase flip error on either $|\bar{+}\rangle$ or $|\bar{-}\rangle$ will flip the state to the other basis vector. In other words, a phase flip in the $\{|0\rangle, |1\rangle\}$ basis is a bit flip error in the $\{|+\rangle, |-\rangle\}$ basis and vice versa. This will prove to be a useful observation shortly. Note there there is another code that protects against phase flip errors,

$$|\bar{0}\rangle = |+++\rangle$$
$$|\bar{-}\rangle = |---\rangle$$

but this encoding is susceptible to bit flip errors. How do we do both? Note that if we can correct a bit flip or phase flip error, we can correct any single-qubit-error, since such an error would be some combination of a phase and a bit flip.

8.1. **Shor's 9-Qubit Code.** The idea behind Shor's error correction code is to merge the codee protecting against phase flips with the code that protects against bit flips. This is done using the encoding

$$|\bar{0}\rangle = \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3}$$
$$|\bar{1}\rangle = \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3}$$

Note that each term in the tensor product is protected against bitflips, since it uses the classical protocol for correcting for bitflips. This also protects against phase flips, since we can check the relative phases between each of the tensor product terms and compare them to correct a phase flip. So this code effectively detects bitflips using the computational basis and also detects phase flips by checking for bitflips in the Hadamard basis.

Theorem 8.1 (**Quantum Fault Tolerance Theorem**). *If every qubit fails at each step of a computation with independent probability ε. Then if we assume the ability to apply gates to qubits in parallel, measure and discard bad qubits, and introduce new qubits initialized to $|0\rangle$, in addition to reliable classical computation, then we can do a* BQP *problem for ε sufficiently small*

Working with logical qubits with the 9-qubit code is quite cumbersome. There's another encoding that uses 5-qubits that's much easier to follow, but requires the development of the **Stabilizer Formalism**

8.2. **The Stabilizer Formalism.**

Definition 8.2. A **Stabilizer Circuit** is a circuit that is entirely composed of gates from the set

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

A **Stablizer State** is any state that is reachable starting from $|0\ldots0\rangle$ and applying some stabilizer circuit.

One can check just by playing around with stabilizer circuits that they only seem to generate a discrete set of states, which explains why we mentioned earlier that a stabilizer set wouldn't be a universal gate set. For example, all the 1-qubit stabilizer states are

$$\{|0\rangle, |1\rangle |+\rangle, |-\rangle, |i\rangle, |-i\rangle\}$$

As it turns out, the n-qubit stabilizer states form a subspace of all the n-qubit states. In addition, all the amplitudes are uniform, so if there is ever a superposition, it will be an equal superposition of all possible n-qubit states.

Definition 8.3. A unitary U stabilizes a state $|\psi\rangle$ if $|\psi\rangle$ is an eigenvector of U with eigenvalue 1.

Note that if U and V stabilize $|\psi\rangle$, then UV, VU, U^{-1}, V^{-1} also stabilize U. Therefore, the set of matrices that stabilize a given state $|\psi\rangle$ is a group, often denoted $\text{stab}(|\psi\rangle)$

The **Pauli Matrices** are the matrices

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

Note that these matrices seem to correspond exactly with the possible single-qubit errors. The Pauli matrices are quite prevalent in quantum mechanics, and also appear in group theory. They satisfy several identities

$$X^2 + Y^2 + Z^2 = I$$

$$XY = iZ \qquad YX = -iZ$$

$$YZ = iX \qquad ZY = -iX$$

$$ZX = iY \qquad XZ = -iY$$

Also note that all these matrices are unitary and Hermitian. What states do these matrices stabilze? $\pm I$ stabilizes everything and nothing respectively. $\pm X$ stablilizes $|+\rangle$ and $|-\rangle$ respectively, $\pm Z$ stabilizes $|0\rangle$ and $|1\rangle$ respectively, and $\pm Y$ stablizies $|i\rangle$ and $|-i\rangle$ respectively.

Definition 8.4. Given an n-qubit state $|\psi\rangle$, the **Stabilizer Group** of $|\psi\rangle$, denoted $\text{stab}(|\psi\rangle)$, is the group of all tensor products of Pauli Matrices that stabilize $|\psi\rangle$.

Remark. For notational convenience, we're going to let AB denote the tensor product of A with B, since we're not going to be multiplying many matrices in the near future.

Example 8.5. We'll compute some easy stabilizer groups here

$$\text{stab}(|0\rangle) = \{I, Z\}$$

$$\text{stab}(|+\rangle) = \{I, X\}$$

$$\text{stab}(|0\rangle \otimes |+\rangle) = \{II, IX, ZI, ZX\}$$

$$\text{stab}\left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right) = \{II, XX, -YY, ZZ\}$$

$$\text{stab}\left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right) = \{II, -XX, ZZ, YY\}$$

We're going to use the following without proof

Theorem 8.6. *The stabilizer states on n-qubits arethe states with stabilizer groups of order 2^n, which has a generating set of order n, which uniquely determines the stabilizer state.*

Theorem 8.7 (**Gottesman-Knill**)**.** *Stabilizer circuits can be simulated with a classical computer in polynomial time.*

There's actually a way to implement this, which you can program yourself if you wish. The idea is to identify states with a set of generators for their stabilzer group, and then use those generators to make a matrix, which we call a **tableau representation** this matrix will take the form

$$(\pm \mid X \mid Z)$$

We'll mostly ignore the signs here, which will lose some information, but isn't too big of a deal. The X and Z blocks will be $n \times n$ matrices, where n is the number of qubits in the state we are trying to simulate. Then an entry X_{ij} is 1 if that spot should be either an X or Y matrix, and is 0 otherwise. Similarly, an entry Z_{ij} is 1 if that spot is either a Z or a Y, and is 0 otherwise. For example, the tableau for the state $|0000\rangle$ is

$$\left(\begin{array}{c|cccc|cccc} + & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ + & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ + & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ + & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

Corresponding to the generating set $\{ZIII, IZII, IIZI, IIIZ\}$. Now we have a way to represent a state, how do we update the state after we apply a stabilizer gate?

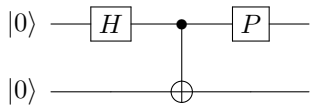
(1) **Hadamard on qubit** i - Swap the i^{th} column of X with the i^{th} column of Z
(2) **Phase on qubit** I - Bitwise xor the i^{th} column of the X matrix into the i^{th} column of the Z matrix
(3) CNOT **with qubit** i **controlling qubit** j - Bitwise xor the i^{th} column of X into the j^{th} column of X. In addition, Bitwise xor the j^{th} column of Z into the i^{th} column of Z

The rules listed above sometimes miss when the sign flips. but it's not too big of a deal. The matrices end up encoding a lot of information without keeping track of signs. Some things to note,

(1) Measuring qubit i in the computational basis will have a determinate outcome if and only if the i^{th} column of the X matrix is all 0s.
(2) The number of basis states our state is a superposition over is 2^k where k is the rank of the X matrix.

Let's run through a simple example.

Example 8.8. Consider the circuit



Our initial state is $|00\rangle$, which has the tableau

$$\left(\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

Applying H, to the first qubit, we swap the first column of the X and Z matrices, giving us the tableau

$$\left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

Then we have a CNOT with the first qubit controlling the second qubit. This results in the tableau

$$\left(\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array}\right)$$

Finally, applying the phase gates gives us our final state, with corresponding tableau

$$\left(\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{array}\right)$$

You can compute the states you get from running the circuit in the standard way with Dirac notation and verify that the state is indeed stabilized by the Pauli matrices specified in the stabilizer formalism.