Lecture Notes for CS 150 - Quantum Computer Science

Instructor: Saeed Mehraban
Tufts University

January 15, 2026

Contents

Chapter 1

From digital computation to quantum computation

In this chapter, we define quantum computation as a generalization of classical digital computation. We begin by introducing classical bits of information and their quantum counterpart, known as "qubits". We then explain a basic transformation on qubits known as the Hadamard transform and discuss nontrivial features of quantum processes, such as reversibility and the uncertainty principle, through the lens of this example. This is the subject of section 1.1. Reversability is one of the inherent features of quantum mechanical laws. One may wonder whether computation is possible using reversible gates. In section 1.2 we explain that this is indeed possible and formulate arbitrary classical computations using reversible means. Finally, in section 1.3 we introduce quantum computations as a generalization of classical reversible computation.

1.1 Quantum v.s. classical bits

In this section we introduce quantum bits as a generalization of classical bits.

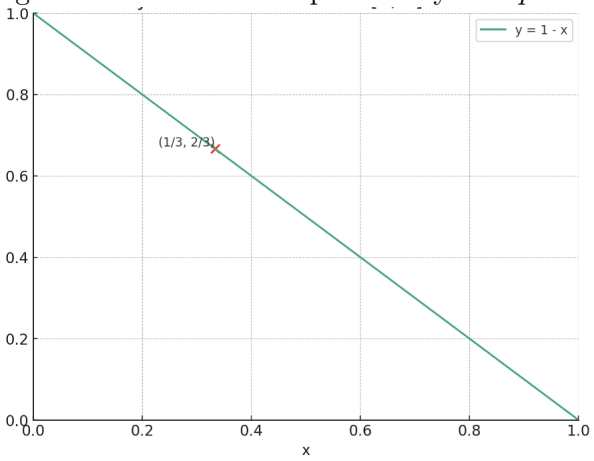
1.1.1 Classical bits

One bit of classical information is represented by a Boolean variable x which can take values 0 or 1. Such a bit may encode the answer to a Yes/No question or may mean an operations was successful or not, etc. In general, binary data are represented using strings of bits. A string is a concatenation of zeros and ones, e.g., 001010100.

A digital computer is a machine that takes as input a sequence of bits, goes through certain (mechanical) steps, and converts the input string into an output string. For instance consider the task of deciding whether a number represented in binary is even or odd. Suppose we encode 0 to mean the number is even and 1 to mean it is odd. Starting with a binary input, the output is 0 if the least significant bit of the number is 0 and is 1 if it is 1. For instance, the procedure takes the even number 110 as input and outputs 0.

Classical bits can be realized using physical architectures. We can define different configurations of a physical system to correspond to different binary values. For example: a switch can be open or closed or a magnet may be oriented upwards or downwards. At the level of implementation, we may store a bit on a physical tape, the state of a transistor or some other means. Each position on the tape stores a deterministic value 0 or 1.

Figure 1.1: The classical probability line $q = 1 - p$



THE DIRAC NOTATION: The Dirac notation is extensively used to represent quantum states as vectors in a Hilbert space, which is a **complete, normed**[1] complex inner-product space. Basically we represent a vector v using a ket notation $|v\rangle$. We will predominantly use this notation when we discuss quantum computing. In order to get prepare for quantum computing, for now we can use this notation to represent classical bit-strings. For instance we represent 101 by $|1\rangle|0\rangle|1\rangle = |101\rangle$. In some cases, if our bit-string is encoding a natural number, as in using the string 101 to encode the value 5, it may be more convenient to write $|5\rangle$. As another piece of notation, we can write $|0\rangle_a |1\rangle_b$ to mean register a holds 0, and register b holds 1.

1.1.2 Classical probability

Quantum mechanics is inherently probabilistic. We can view quantum probability as a generalization of classical probability. Let us briefly review classical probability. Consider the example where we have a register that contains the bit 0 with probability 2/3 and 1 with probability 1/3. We can represent this state as a probability vector $\begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix}$. More generally, we can represent it using the vector $|p\rangle = \begin{pmatrix} p \\ q \end{pmatrix}$, for $q = 1 - p$ and $0 \leq p \leq 1$. We can visualize any such vector as some point on the line $q = 1 - p$ (see fig. 1.1).

We also can consider larger (discrete) probability spaces. For example: $|\mathbf{p}\rangle = \begin{pmatrix} p_1 \\ \ddots \\ p_N \end{pmatrix}$, with $p_i \geq 0$ such that $\sum_i p_i = 1$.

1.1.3 Quantum bits

We now present quantum bits as a generalization of probabilistic classical bits. Physically, we can encode a quantum bit within the degrees of freedom of a physical system such as electron spin (up or down), or photon polarization (clockwise or counter clockwise). Let's define a quantum bit mathematically. We use the Dirac vector notation. We represent one quantum bit as a complex vector in a two-dimensional Hilbert space $\mathbb{C}^2$. The quantum bit corresponding to state 0 with a basis vector $|0\rangle$ and quantum bit corresponding to 1 with a vector $|1\rangle$ (which is

[1] These terms have mathematical definitions, but we wont use them much in this course.

orthogonal to $|0\rangle$). In other words:

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

A quantum bit may be in a *superposition* (linear combination) of the basis states:

$$\begin{aligned}|\psi\rangle = \begin{pmatrix}\alpha\\\beta\end{pmatrix} &= \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix},\\ &= \alpha|0\rangle + \beta|1\rangle.\end{aligned} \tag{1.1}$$

where $|\alpha|^2 + |\beta|^2 = 1$ We also call α and β the complex amplitudes for the state vector $|\psi\rangle$. *Notice that while a classical probabilistic bit corresponds to a point on a line, we can think of a quantum bit as corresponding to a point on the unit circle. (see Figure 2).*

Example 1. *As an example, consider the following two quantum states:*

$$|+\rangle := \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}, |-\rangle := \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}.$$

We can have more quantum states by using complex numbers, e.g., $|\pm i\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm i|1\rangle)$. *This means that our picture of a quantum bit being a point on the unit circle is not exactly correct, since our state is described by more than 2 real parameters. It turns out that a quantum bit is actually a point on the surface of a 3-dimensional sphere (see figure 3).*

MEASUREMENT: When we measure the quantum state $|\psi\rangle = \begin{pmatrix}\alpha\\\beta\end{pmatrix} = \alpha|0\rangle + \beta|1\rangle$ in the standard basis we obtain the value 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$. Recall that for a classical probabilistic bit, the probability of measuring $0, 1$ is p, q, respectively. In contrast, for a quantum bit, the squared moduli give us the probabilities. Note that the probabilities corresponding to the state with a *global phase* θ, i.e. $\alpha(e^{i\theta})|0\rangle + \beta(e^{i\theta})$, are exactly the same as the original $|\psi\rangle$. For this reason, we say that a global phase applied to any state vector is physically unobservable.

Exercise 1. *Can we distinguish* $|+\rangle$ *from* $|-i\rangle$ *in the standard basis?*

HIGHER-DIMENSIONAL STATES: The same way we defined higher-dimensional versions of classical probabilistic bits, we can define states corresponding to higher dimensions, usually called a *qudit*, corresponding a d-dimensional ector. In particular, a qudit $|\psi\rangle \in \mathbb{C}^d$ can be written as $|\psi\rangle = \begin{pmatrix}\alpha_0\\\alpha_1\\\dots\\\alpha_{d-1}\end{pmatrix} = \sum_{j=0}^{d-1}\alpha_j|j\rangle$, with $\sum_i |\alpha_i|^2 = 1$.

CONJUGATE VECTORS: We can define a conjugate vector $\langle\psi| = (\alpha_0^*, \alpha_1^*, \dots)$. Based on this definition $\langle\psi,\psi\rangle = \sum_i |\alpha_i|^2$ which is $= 1$ for a quantum state. We can also define an inner product between two different states by $\langle\psi,\phi\rangle = \sum_i \psi_i\phi_i^*$. We can see that

$$\langle 0,1\rangle = 0, \qquad \langle +,-\rangle = 0.$$

Figure 1.2: The amplitude circle $\alpha^2 + \beta^2 = 1$

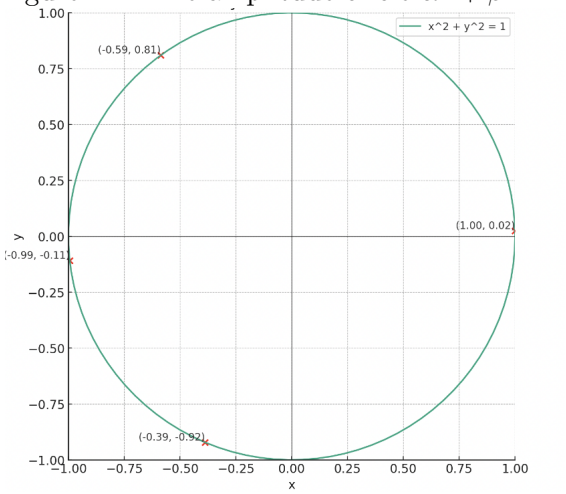
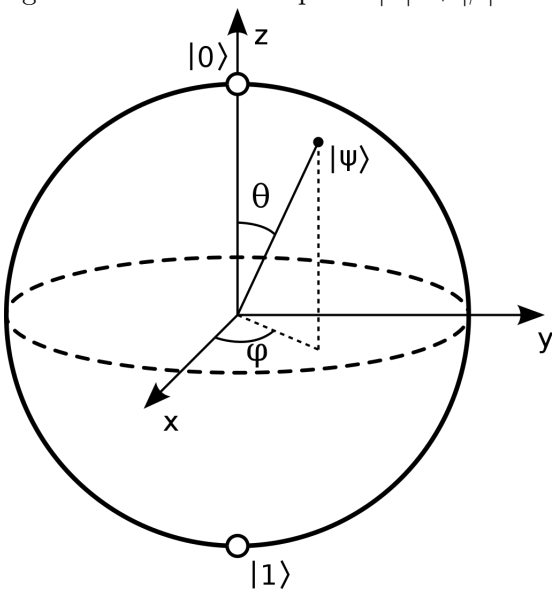


Figure 1.3: The Bloch sphere $|\alpha|^2 + |\beta|^2 = 1$



Quantum states

d level quantum states are unit vectors in a Hilbert space $\mathbb{C}^d$. A quantum state $|\psi\rangle \in \mathbb{C}^d$ can be written as $|\psi\rangle = \alpha_0 |0\rangle + \ldots + \alpha_{d-1} |d-1\rangle$ satisfying $\langle\psi|\psi\rangle = |\alpha_0|^2 + \ldots + |\alpha_{d-1}|^2 = 1$.

1.1.4 Change of basis

So far we saw that a quantum bit is a generalization of the probability distribution for a two-level system. It is represented by a two-dimensional complex vector $\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$, for $\alpha, \beta \in \mathbb{C}$, that has unit norm $|\alpha|^2 + |\beta|^2 = 1$. A quantum bit can be implemented using various physical systems. That can be polarization of light or the number of photons in an optical mode. One important example of a quantum bit is the internal spin of an electron. Electron spin can be up or down (rotating clockwise or counter-clockwise) in any direction: top-down, left-right, front-back or any other direction. Let us denote the spin up and down states using $|0\rangle$ and $|1\rangle$, respectively. Similarly define right-spin and left-spin as $|+\rangle$ and $|-\rangle$, respectively. How are these two pairs of basis states related to each other?

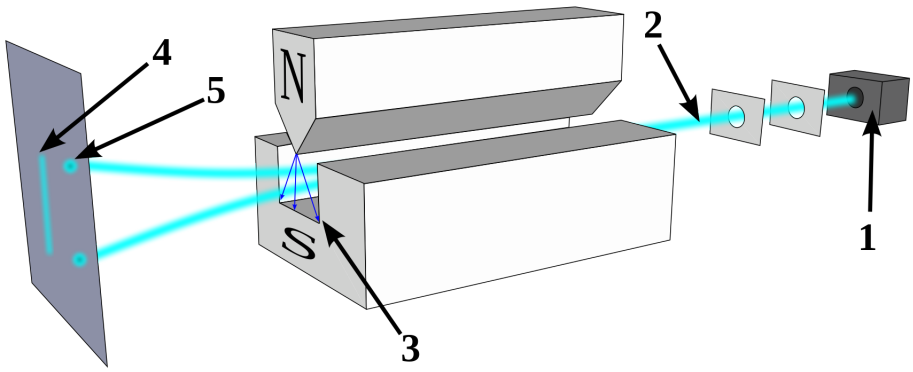
In this section, we use heuristic reasoning to deduce how the states $|+\rangle$ and $|-\rangle$ are related to $|0\rangle$ and $|1\rangle$. We start by noting that each set of basis vectors $\{|0\rangle, |1\rangle\}$, $\{|+\rangle, |-\rangle\}$ describe the same quantum system, i.e. the electron spin. This means $|+\rangle = \alpha |0\rangle + \beta |1\rangle$ and $|-\rangle = \gamma |0\rangle + \delta |1\rangle$. They are furthermore orthonormal basis, i.e., $\langle 1|0\rangle = \langle 0|1\rangle = 0$ and $\langle 0|0\rangle = \langle 1|1\rangle = 1$; similarly $\langle +|-\rangle = \langle -|+\rangle = 0$ and $\langle +|+\rangle = \langle -|-\rangle = 1$. $\alpha^*\delta + \beta^*\gamma = 0$, furthermore $|\alpha|^2 + |\beta|^2 = 1$ and $|\gamma|^2 + |\delta|^2 = 1$. Let's assume $\alpha, \beta, \gamma, \delta$ are all real numbers. We can show that (up to a global sign) our solution to these equations takes a form like:

$$|+\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |-\rangle = \beta|0\rangle - \alpha|1\rangle,$$

Here $\alpha^2 + \beta^2 = 1$. How can we find α and β? To do this, we need to evaluate the inner product $\alpha = \langle 0|+\rangle$. We recall that $|\langle 0|+\rangle|^2$ is the probability of observing the state $|+\rangle$ to be $|0\rangle$. Can we design an experiment to deduce this? The Stern-Gerlach experiment, which we talked about during the first lecture, exactly performs this measurement. In the Stern-Gerlach experiment, we prepare two opposite magnets (oriented top-down) designed in a way that a spin-up electron will move upwards and a spin-down electron will move down. Please see Figure 1.4 (For more information you can read the Wikipadia article `https://en.wikipedia.org/wiki/Stern%E2%80%93Gerlach_experiment`). Classical physics predicts that if the orientation of the electron spin is an angle other then up or down then the electron beam should land somewhere between the top or down position at the screen (see arrow number number 4 in the Figure). Particularly, if the spin is in the left-right direction, it should not be deflected at all. The surprising outcome of the Stern-Gerlach experiment is that we only see a top point or a bottom point (see arrow number 4 in the Figure) from the experiment. That means the value of spin is quantized and once we measure it, it takes a value up or down. What is the state of the electron before measurement? It is a "superposition" of top and down. Superposition is a fundamental concept in quantum physics. What if I rotate the experimental setup 90 degrees and measure the left-right spin? Surprisingly, we get two points again. One on the leftmost side of the screen and one on the rightmost side. So, in a way, the state of each electron before measurement is in a superposition of being up or down, and at the same time, it is in a superposition of being left and right. We do not have a classical counterpart for such an observation.

The electron beam in this experiment is in a highly complex mixture of spins in various directions. What if we do an experiment to make sure all electron beams are in the top direction? See Figure 1.1.4. We compose two Stern-Gerlach (SG) experiments. In the first experiment we

Figure 1.4: The Stern-Gerlach experiment. Arrow number 4 is the classical prediction. Arrow number 5 is the actual outcome of the quantum experiment. Image Source: Wikipedia.



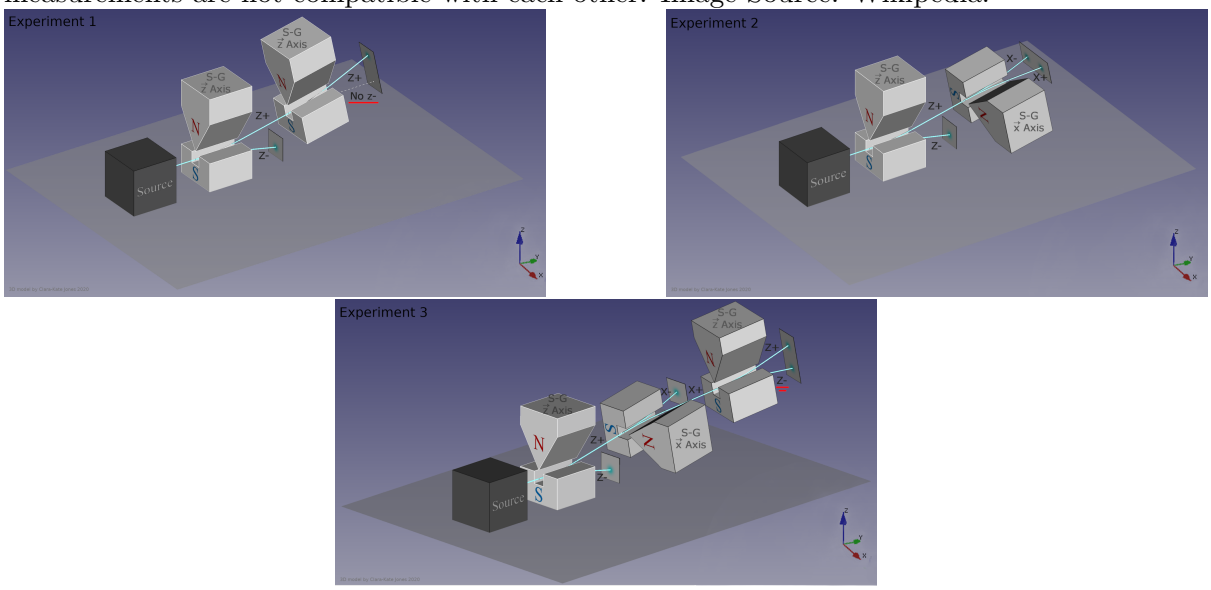
compose two top-down experiments. We see that in the first experiment, electrons in top directions are selected, and as we pass the resulting beam through the second experiment, we only see spin-up electrons. We don't see spin-down electrons! This is consistent with our intuition. In a way, the beam going through the second experiment is deterministically in the $|0\rangle$ state. What if we pass this second beam through a left-right experiment? This is the subject of the second experiment in Figure 1.1.4. Classically, we expect that the beam should not be deflected at all. This is because the electron spin has 90 90-degree angle from the N-S magnet in the second SG experiment. To our surprise, we see that we obtain both left spins and right spins, with probability 1/2 each. (The third experiment is very similar to the second experiment; we only exchange the order of the two SG measurements). From this experimental observation, we deduce that when the electron is in a spin-up state, it has a 1/2 probability of being measured in spin-left or spin-right states! We hence deduce that $|\langle 0|+\rangle|^2 = \frac{1}{2} = \alpha^2$. Therefore $\alpha = \beta = \frac{1}{\sqrt{2}}$.

The uncertainty principle: Using the SG experiment (second or third experiments in Figure 1.1.4), we arrive at one of the most important concepts in quantum physics: the uncertainty principle.

"If the electron spin is fully deterministically in the up direction, its spin value in the left-right direction is completely uncertain. If we measure the spin in the left-right direction, half of the time, we observe spin left, and the other half of the time, we observe spin right! Similarly, if we prepare an electron spin in the right-spin direction and measure the spin in the top-down direction, we get a completely uncertain outcome. Half of the time, we obtain spin up, and the other half, we obtain spin down."

In this situation, we say that the top-down and left-right spins are **incompatible observables**. There are several other examples of incompatible observables in quantum physics. One of the first examples of the uncertainty principle was the uncertainty principle was about the position

Figure 1.5: Three Stern-Gerlach experiments. In the first experiment we measure the top-down direction twice. In the second experiment we first measure top-down then left-right, and in the third experiment we measure left-right then top-down. While in the first experiment the outcome is deterministically a top spin in the other two experiments the outcome is probabilistic mixture of each direction. In other words, the left-right measurement and the top-down measurements are not compatible with each other. Image Source: Wikipedia.



and velocity of particles. That means if we learn the position of a particle with high accuracy, we lose the information about the velocity and vice versa. Heisenberg explained this phenomenon using the following intuition. If we want to measure the position of a particle. We have to interact with it using, say, a photon. If we want to learn the position with high accuracy, we will have to alter the information about velocity. So we can't learn both.

Remark 1. *This situation is similar to the uncertainty of variables that are Fourier transforms of each other in signal processing and Fourier analysis. This is not a coincidence. You can read more about the role of the Fourier transform for quantum observables in standard quantum physics textbooks. We won't delve deeper into this notion in this course.*

Measuring a Qubit in an Arbitrary Orthonormal Basis: We have so far explained measurement of a quantum state in the basis $\mathcal{C} = \{|0\rangle, |1\rangle\}$, usually referred to as the computational basis. We can, however, measure a qubit in any complete orthonormal basis for the Hilbert space $\mathbb{C}^2$. For instance, let $|A\rangle = \alpha |0\rangle + \beta e^{i\phi} |1\rangle$ be an arbitray qubit with real valued α, β satisfying $\alpha^2 + \beta^2 = 1$; $\phi \in [0, 2\pi)$ is an arbitrary phase. Let $|B\rangle = -\beta |0\rangle + \alpha e^{i\phi} |1\rangle$.

Exercise 2. *Show that $\mathcal{A} = \{|A\rangle, |B\rangle\}$ corresponds to an orthonormal complete basis for $\mathbb{C}^2$.*

Now suppose $|\psi\rangle \in \mathbb{C}^2$ is an arbitrary quantum state. We can decompose $|\psi\rangle$ into the $\mathcal{A}$ basis using the following result:

Exercise 3. *Show that*

$$|A\rangle \langle A| + |B\rangle \langle B| = I.$$

This is known as the completeness relationship. As a matter of fact, we can show for any orthonormal complete basis $\mathcal{A}_d = \{|A_1\rangle, \ldots, |A_d\rangle\}$ for $\mathbb{C}^d$,

$$|A_1\rangle \langle A_1| + \ldots + |A_d\rangle \langle A_d| = I$$

Prove this.

Using the tool in this exercise, we can show

$$\begin{aligned} |\psi\rangle &= I\,|\psi\rangle = (|A\rangle\langle A| + |B\rangle\langle B|)\,|\psi\rangle \\ &= \langle A|\psi\rangle\,|A\rangle + \langle B|\psi\rangle\,|B\rangle \\ &=: \gamma_A\,|A\rangle + \gamma_B\,|B\rangle\,. \end{aligned} \tag{1.2}$$

Where $\gamma_B := \langle A|\psi\rangle$ and $\gamma_B := \langle B|\psi\rangle$ are innerproducts, and are called amplitudes of $|\psi\rangle$ in the $\mathcal{A}$ basis. The probability of measuring $|\psi\rangle$ in the $\mathcal{A}$ basis and obtaining label A and B are $|\gamma_A|^2$ and $|\gamma_B|^2$, respectively.

General formulation of quantum measurements: We note that the outer product operators $\Pi_A := |A\rangle\langle A|$ and $\Pi_B := |B\rangle\langle B|$ are positive semi-definite, i.e., $\Pi_A \geq 0, \Pi_B \geq 0$ and $\Pi_A + \Pi_B = I$. The probability of obtaining label $j \in \{A, B\}$ is $|\gamma_j|^2 = \langle\psi|\,\Pi_j\,|\psi\rangle$. This is not a coincidence. As a matter of fact, the most general formulation of quantum measurement is according to the following definition.

General Formulation of Quantum Measurement

Given an arbitrary Hilbert space $\mathcal{H}$, any quantum measurement can be formulated using a *positive operator valued measure* (POVM) defined as a set of operators

$$\mathcal{M} = \{M_1, \ldots, M_T\}, \quad T \geq 1,$$

such that

$$M_j \geq 0 \quad \text{and} \quad \sum_j M_j = I.$$

By measuring a quantum state $|\psi\rangle$ according to the POVM $\mathcal{M}$, we obtain a probability distribution $P_1, \ldots, P_T$ with

$$P_j = \langle\psi|\,M_j\,|\psi\rangle\,.$$

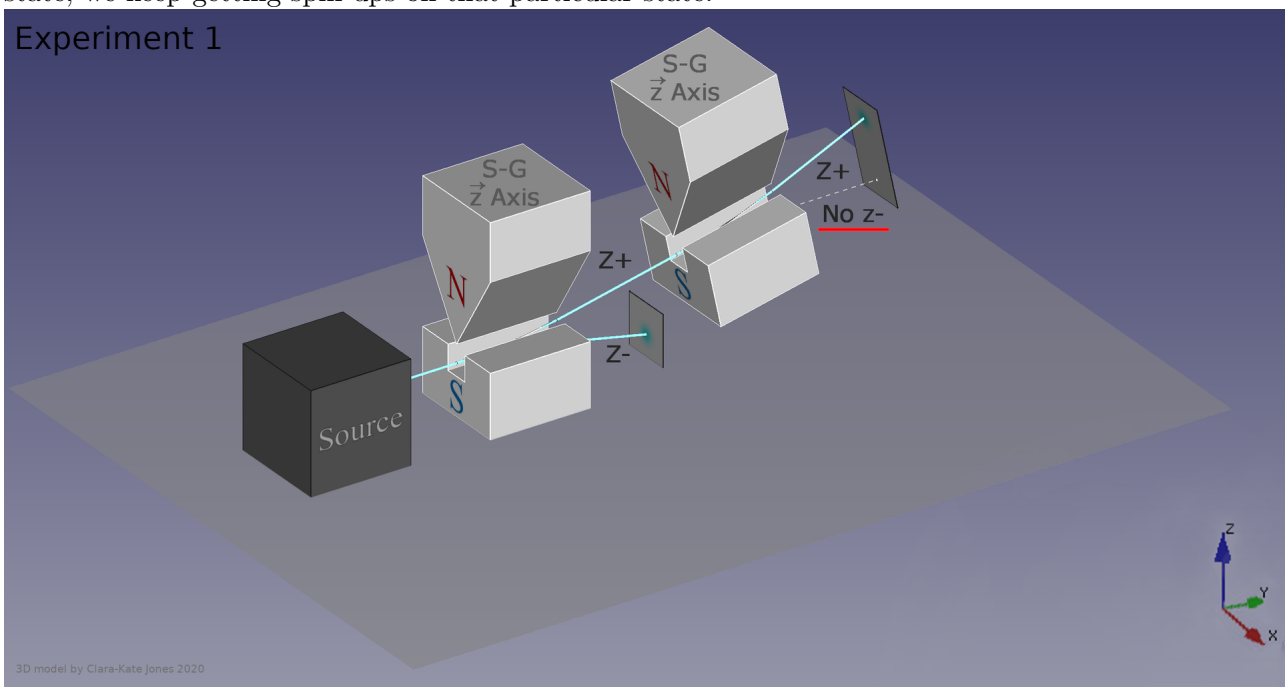
Exercise 4. *Show that P_j corresponds to a probability distribution.*

A special case of POVM is a *projective valued measurement* (PVM) corresponding to M_j being projectors. Measurement in an arbitrary orthonormal basis corresponds to a PVM.

1.1.5 What happens after a quantum measurement?

Suppose we prepare a quantum state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ in an equal superposition of $|0\rangle$ and $|1\rangle$. If we measure the quantum state in $|0\rangle, |1\rangle$ basis, we obtain a probabilistic outcome, but if we measure the same state in the $|+\rangle, |-\rangle$ basis, we obtain a deterministic result. This suggests that the state of the quantum system was not really $|0\rangle$ or $|1\rangle$, but rather a superposition of the two; it was the state $|+\rangle$. What happens to the quantum state after measurement? Recall the Stern-Gerlach experiment. Suppose we measure the electron spin on an up-down basis and filter out the electrons that have been measured to be down. If we measure the upper beam on an up-down basis one more time, we obtain spin-up one more time. If we continue doing this, we will continue to experience spin-ups. See future 1.6. In other words, once we measure an arbitrary quantum state on any given basis and obtain a certain outcome, the state of the quantum system collapses to that outcome. Previously, we told you that quantum mechanics is a reversible theory. We have to correct this statement: quantum mechanics is a reversible theory "before measurement." Measurement is an irreversible process. We should note that there are several schools of though regarding interpretations of quantum mechanics. The above interpretation of quantum measurements is based on a school of thought known as the Copenhagen school.

Figure 1.6: Two consecutive spin up-down measurements. Once we obtain spin-up in a quantum state, we keep getting spin-ups on that particular state.



Schrödinger's cat

Would the above interpretation of quantum measurements lead to paradoxical outcomes? Schrödinger's cat is a thought experiment in quantum mechanics, proposed by the Austrian physicist Erwin Schrödinger in 1935. It illustrates what he saw as the problem of the Copenhagen interpretation of quantum mechanics when applied to everyday objects.

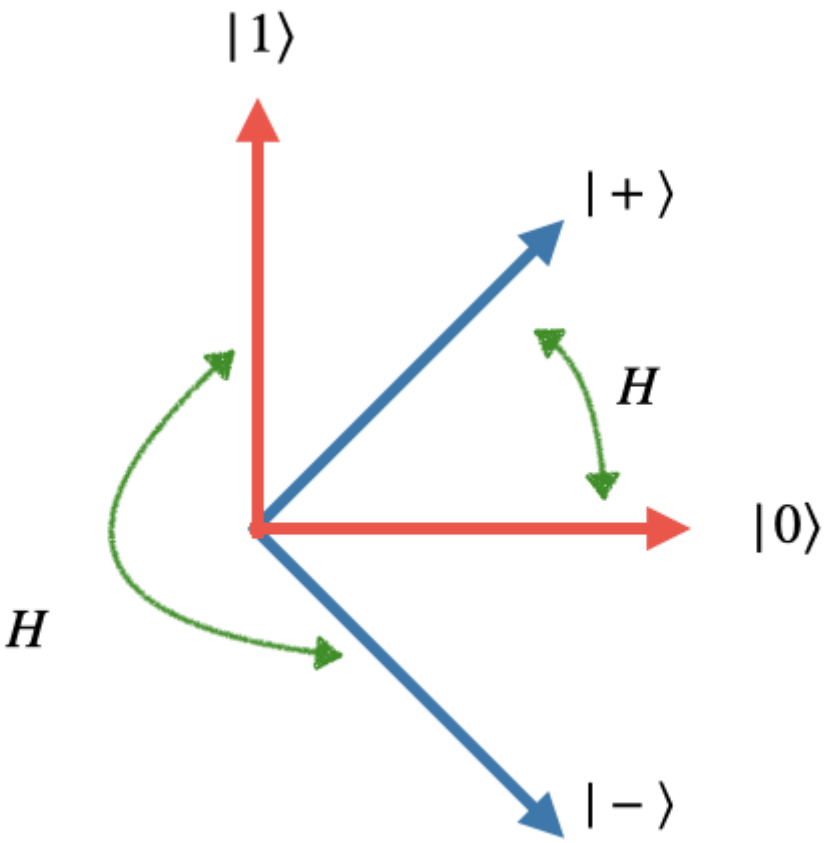
In the thought experiment, a cat is placed in a sealed box with a radioactive atom, a Geiger counter, a hammer, and a vial of poison. If the radioactive atom decays, the Geiger counter triggers the hammer to break the vial, which would kill the cat. As long as the box remains closed, the state of the radioactive atom is in a superposition of triggering and not triggering the hammer. Hence, the cat is simultaneously alive and dead in a superposition of states. Once the box is opened, the cat is observed to be either alive or dead, not both. Suppose we keep the box closed for many years, and then open it to find that the cat is dead. When did the cat really die?

This paradox is often used to illustrate the peculiarities of quantum mechanics and the concept of superposition, where particles can exist in multiple states simultaneously until they are observed. The thought experiment was not intended to be practical, but rather to illustrate the potential issues and interpretations of quantum mechanics in understanding real-world objects and systems.

1.1.6 Hadamard transform

In the previous section, we saw that by rotating the SG experiment 90 degrees, we change the basis of measurement from $\{|0\rangle, |1\rangle\}$ to $\{|+\rangle, |-\rangle\}$. Similarly, suppose I start with the quantum state $|0\rangle$. If I keep the direction of the experiment fixed and rotate the electron, I have mapped the state of the electron from $|0\rangle$ to $|+\rangle$. Similarly, if I start with $|1\rangle$ and rotate it 90 degrees,

Figure 1.7: The Hadamard transform.



the state of the electron becomes $|-\rangle$ (up to an unobservable minus sign). We saw that classical computations have matrix representations. Is there a matrix representation for the operation that performs the map $|0\rangle \mapsto |+\rangle$ and $|1\rangle \mapsto |-\rangle$? The answer is yes, and this map is called the **Hadamard transform**. It is the matrix:

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

See Figure 1.14 for a geometric interpretation of the Hadamard operation. Furthermore, the Hadamard operation is **self-adjoint**, meaning that $H = H^\dagger$ where $H^\dagger$ is the conjugate-transpose. (Recall that we define $(A^\dagger)_{ij} = A^*_{ji}$). In other words, $HH^\dagger = I$. In linear algebra, we refer to such matrices as *unitary*. The Hadamard matrix also satisfies $H^{-1} = H$. That means it is its own inverse. Recall our discussion about reversible computations. We can see from this example that the Hadamard operation is reversible.

We can view the Hadamard transform as a coin-flip operation. Because when we apply it to $|0\rangle$, we obtain $|+\rangle$, which has $1/2$ probability of being in $|0\rangle$ or $|1\rangle$. How is this quantum coin flip different from a classical coin flip? If we flip a coin and obtain heads, then there is no way to go back and see if the initial state of the coin was heads or tails. However, if we use the Hadamard operation twice, we will get back to where we started! It is easy to check that $H^2 = I$.[2]

[2]Linear operations which are their own inverses, are called *involutory*. Note that matrices that are both unitary and Hermitian are always involutory. Why is this true?

Remark 2. *We note that by rotating the Stern-Gerlach experiment, we obtain a change of basis according to the matrix* $R = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$*, which differs from the Hadamard matrix. It maps* $|0\rangle$ *to* $|+\rangle$ *but it maps* $|1\rangle$ *to* $-|-\rangle$*. We introduced Hadamard because of its fundamental role in the rest of this course.*

Exercise 5. *Show that the Hadamard transform is unitary.*

1.1.7 The principle of unitarity

We saw that the Hadamard transform is unitary. What is special about unitarity? Unitaries are linear transforms that preserve the inner product between complex vectors. As a result, quantum states are mapped to quantum states that satisfy the correct probability rule (i.e., the amplitudes square to 1) since they map unit vectors onto unit vectors. They are precisely the family of linear maps that preserve the norm of a vector. As a matter of fact, unitarity is one of the main postulates in the formulation of quantum mechanics:

Unitarity

For a closed quantum system, the most general transformation describing the evolution of a quantum state is a unitary matrix. A matrix U is unitary if $U^\dagger = U^{-1}$. In particular, starting with the initial state of a closed system $|\phi_0\rangle$, for any unitary matrix U, there exists a quantum experiment which maps $|\phi_0\rangle$ to $|\phi_1\rangle = U|\phi_0\rangle$.

Exercise 6. *For a quantum state* $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$*. Let* $H|\psi\rangle = \alpha'|0\rangle + \beta'|1\rangle$*. Show that* $|\alpha'|^2 + |\beta'|^2 = 1$*.*

More generally, a matrix U is called unitary if for any pair of vectors $|\psi\rangle$ and $|\phi\rangle$, $\langle\phi,\psi\rangle = \langle U\phi, U\psi\rangle$. We can show that this criterion is satisfied if $U^\dagger U = I$. For a complex matrix $A \in \mathbb{C}^{M\times N}$ with entries $A_{i,j}, 1 \le i \le M, 1 \le j \le N$, $A^\dagger \in \mathbb{C}^{N\times M}$ with entries $A^\dagger_{i,j} = A^*_{j,i}$.

To see this, we need to define some notation that will be useful throughout this course. For a vector $|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} \in \mathbb{C}^N$ the dual vector is denoted by $\langle\psi| = \begin{pmatrix} \alpha_1^* & \dots & \alpha_N^* \end{pmatrix}$. One useful aspect of this notation is that the inner product of two vectors

$$|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} \quad \text{and} \quad |\phi\rangle = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_N \end{pmatrix}$$

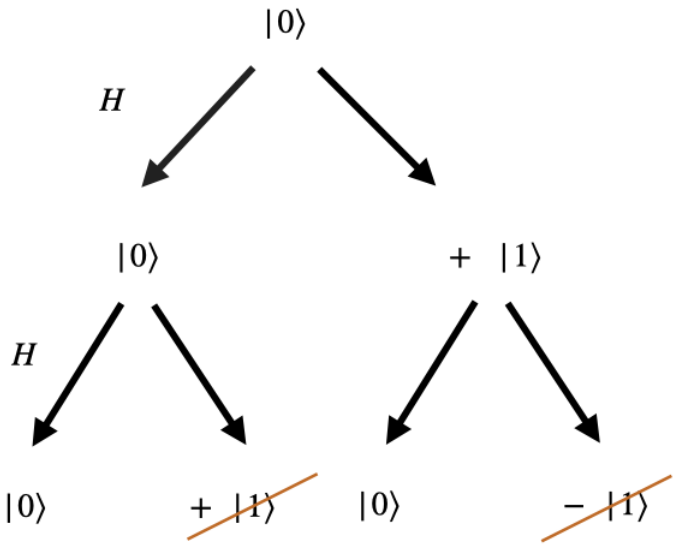
is given by

$$\langle\psi,\phi\rangle = \begin{pmatrix} \alpha_1^* & \dots & \alpha_N^* \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_N \end{pmatrix} = \langle\psi| \cdot |\phi\rangle .$$

We can show that for any matrix $A \in \mathbb{C}^{M\times N}$ and vector $|v\rangle \in \mathbb{C}^N$, $(A|v\rangle)^\dagger = \langle v| A^\dagger$. To see, let $|y\rangle = A|v\rangle$. Then we can expand $y_i = \sum_{j=1}^N A_{i,j} v_j$. Therefore, $y_i^* = \sum_{j=1}^N v_j^* A^*_{i,j} = \sum_{j=1}^N v_j^* A^\dagger_{j,i}$. Therefore $\langle y| = \langle v| A^\dagger$.

Now, using this observation, we want to understand what unitarity means. In particular, $\langle U\phi, U\psi\rangle = \langle\phi|U^\dagger U|\psi\rangle$. If $\langle U\phi, U\psi\rangle = \langle\phi,\psi\rangle$ for all ϕ and ψ, then we have no choice other than to choose $U^\dagger U = I$. We can easily see that this is equivalent to $U^\dagger = U^{-1}$. So, this immediately implies that quantum operations are reversible.

Figure 1.8: The interference phenomenon for two applications of Hadamard.



Example: Here are two examples for 2×2 and 3×3 unitary matrices

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

and

$$V = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{i\frac{2\pi}{3}} & e^{i\frac{4\pi}{3}} \\ 1 & e^{i\frac{4\pi}{3}} & e^{i\frac{2\pi}{3}} \end{bmatrix}$$

INTERFERENCE PHENOMENON: We discussed the strange phenomenon of two quantum bit flips being equivalent to having done nothing. We can explain this using the interference phenomenon. See Figure 1.8. After one application of Hadamard, we obtain a superposition of $|0\rangle$ and $|1\rangle$. If we apply H another time, we obtain four superposition terms $|0\rangle$, $|1\rangle$, another $|0\rangle$ and $-|1\rangle$. $|1\rangle$ and $-|1\rangle$ cancel each other out. This is the same interference phenomenon we observe in classical waves. In other words, one of the immediate differences between quantum and classical computation is the existence of negative signs in the matrix representation of computations.

1.1.8 Single-qubit rotations:

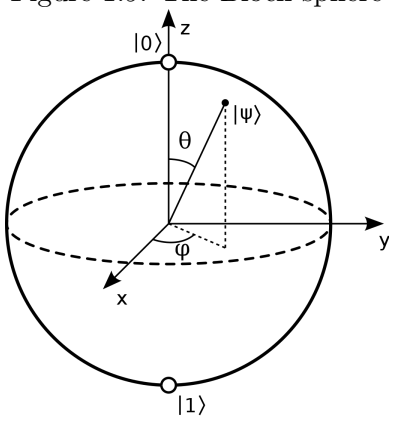
Recall that a vector gives the state of a single two-level system in $\mathbb{C}^2$. So we can represent it by:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, |\alpha|^2 + |\beta|^2 = 1$$

we can use the following angle parameters to represent this qubit

$$|\psi\rangle = \cos(\frac{\theta}{2})\,|0\rangle + e^{i\phi}\sin(\frac{\theta}{2})\,|1\rangle$$

Figure 1.9: The Bloch sphere



The following operator induces a rotation in θ (around Y axis). Rotation operators are special classes of unitary operations.

$$R_Y(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}$$

the following creates a rotation around Z

$$R_Z(\phi) = \begin{pmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{pmatrix}$$

the following creates a rotation around X axis.

$$R_X(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}$$

We can show that $R_i(\theta)$ performs a rotation by angle θ around the i-th axis. Let us study these operators a bit deeper. To generate θ rotation around X, (Y or Z) axis, we can apply the operator $e^{-i\frac{\theta}{2}Z}$ (same for X and Y). Here for an operator A, e^A is given by the Taylor series $e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots$. Suppose A has spectral decomposition $A = \sum_i \lambda_i |i\rangle\langle i|$, then $f(A) = \sum_i f(\lambda_i)|i\rangle\langle i|$. In particular

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1|$$

therefore

$$e^{-i\frac{\theta}{2}Z} = e^{-i\frac{\theta}{2}}|0\rangle\langle 0| + e^{i\frac{\theta}{2}}|1\rangle\langle 1| \tag{1.3}$$

$$= \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \tag{1.4}$$

To derive the formula for $R_X(\theta)$, we use the formula $HZH = X$. Therefore for any integer l, $(HZH)^l = X^l$. Plugging these into the Taylor series we can show $e^{-i\frac{\theta}{2}X} = He^{-i\frac{\theta}{2}Z}H$. Therefore

$$R_X(\theta) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1.5}$$

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} e^{-i\frac{\theta}{2}} & e^{-i\frac{\theta}{2}} \\ e^{i\frac{\theta}{2}} & -e^{i\frac{\theta}{2}} \end{pmatrix} \tag{1.6}$$

$$= \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} e^{-i\frac{\theta}{2}} & e^{-i\frac{\theta}{2}} \\ e^{i\frac{\theta}{2}} & -e^{i\frac{\theta}{2}} \end{pmatrix} \tag{1.7}$$

$$= \begin{pmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}. \tag{1.8}$$

Exercise 7. *Prove that*

$$e^{-i\frac{\theta}{2}Y} = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} = R_Y(\theta)$$

(Hint: $Y = SXS^{-1}$)

How can we describe rotation around an arbitrary direction $\hat{n} = (n_x, n_y, n_z)$ on the Bloch sphere? Let $\sigma = (X, Y, Z)$ and the inner product $\hat{n} \cdot \sigma = n_x X + n_y Y + n_z Z$. When $\hat{n} = (1, 0, 0)$ we obtain X, for $\hat{n} = (0, 1, 0)$ we obtain Y and for $\hat{n} = (0, 0, 1)$ we obtain Z. It turns out rotation by θ around $\hat{n}$ axis is given by $R_{\hat{n}}(\theta) = e^{-i\frac{\theta}{2}\hat{n}\cdot\sigma}$. In general, an arbitrary single-qubit quantum operation can be captured in the form $e^{i\alpha}R_{\hat{n}}(\theta)$ for some unit vector $\hat{n}$.

Exercise 8. *Show that $(\hat{n}\cdot\sigma)^2 = I$. Using this observation show that $R_{\hat{n}} = \cos(\frac{\theta}{2})I + i\sin(\theta)(\hat{n}\cdot\sigma)$. We recommend reading Section 4.2 of Nielsen Chuang for more information.*

It turns out that an arbitrary single qubit unitary operation can be decomposed as $U = e^{i\alpha}R_Z(\gamma)R_Y(\beta)R_X(\gamma)$ for suitably selected $\alpha, \beta, \gamma, \delta$.

1.2 Reversible operations on classical bits

It was known since the early days of quantum mechanics that quantum mechanics is reversible by nature. The question, hence, was whether we can perform computation using reversible elements. For instance the AND gate is not reversible, since it has 2 inputs and only 1 output. In the 1980's Ed Fredkin and Toffoli answered this question by proposing a reversible model based on billiard balls that could simulate arbitrary classical computations. We will build up an understanding of reversible computation in this section.

We first start with boolean functions which encode input-output pairs for any classical problem we can imagine, then classical circuits which implement boolean functions using digital logic gates, then formally introduce reversible classical circuits implementing arbitrary classical circuits with no information loss. Finally, we show that reversible classical circuits can be implemented using matrix calculations. We show how computation can be performed by evolving vectors using matrix transformations. We will use these concepts as a foundation to quantum computations on quantum bits which we will present in the next section.

1.2.1 Boolean logic

Let's first introduce Boolean logic.

Boolean function: A Boolean function is a function that takes a value in $\{0,1\}^*$ and outputs a value in $\{0,1\}^*$. For example the XOR function is the following function

$$XOR(x) = \begin{cases} 0 & x \text{ has even number of ones} \\ 1 & \text{otherwise} \end{cases}$$

is a Boolean function. We often fix an integer n and work with functions $f : \{0,1\}^n \to \{0,1\}$. In other words $f(x) = x_0 + \ldots + x_{n-1} \mod 2$. Other important Boolean functions are the AND, OR and NOT gates. The NOT gate simply flips the input bit. The AND gate outputs 1 only if all inputs are 1, and the OR gate outputs 1 if at least one of the bits is set to 1. Usually, we assume AND and OR gates take 2 input bits. But they can also be defined over larger number of input bits.

Figure 1.10: The truth table for functions NOT, AND and OR

x	$\bar{x}$
0	1
1	0

$f(x) = NOT(x)$

x	y	$AND(x,y)$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	$OR(x,y)$
0	0	0
0	1	1
1	0	1
1	1	1

Importantly, the input-output pairs for any classical problem we can imagine (or compute) can be fully represented as a boolean function. That is, a boolean function can be any mapping of classical input information to classical output information.

TRUTH TABLE: For a Boolean function f the *truth table* for f is a table which describes input and outputs pairs. In particular, each input string $x \in \{0,1\}^n$ corresponds to a row which describes x and $f(x)$. You can see the example of the truth table for AND, OR and NOT in Figure 1.10. A truth table has 2^n rows since there are 2^n possible input bitstrings.

Exercise 9. *How many different Boolean functions with n-bit inputs and 1-bit outputs?*

Answer: There are 2^n rows and each row can output either 0 or 1. Thus, there are $2^{(2^n)}$ possible truth tables.

BOOLEAN GATE-SET: An important observation is that arbitrarily large Boolean circuits can be constructed using basic Boolean functions that take one or two inputs. A gate set is a fixed set of Boolean functions that take a certain (say $1, 2, 3, \ldots$) number of input bits and output a certain number of output bits. We use gate-sets to produce larger gate-sets.

BOOLEAN CIRCUIT: A Boolean circuit is a diagram that represents compositions of gates from a gate set. You can think of it as a directed acyclic graph. Each edge encodes a Boolean value and each node is a gate. See for example the circuit producing XOR in Figure 1.11 (From here).

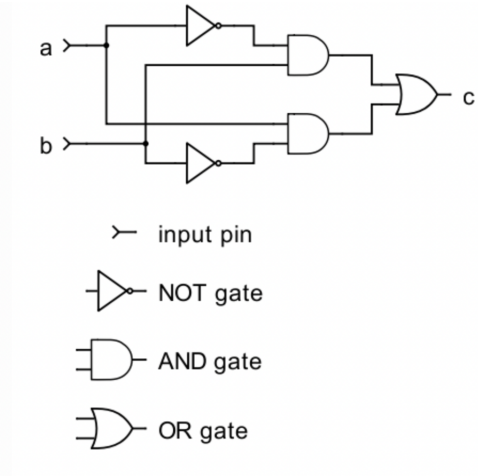
UNIVERSAL GATE-SET: A gate set is universal if, by composing gates from this gate set, we can generate every Boolean function.

Exercise 10. *Show that $\{AND, OR, NOT\}$ is a universal gate set.*

Exercise 11. *Show that $NAND$ by itself is a universal gate set. Prove the same result for NOR.*

Exercise 12. *How many gates is sufficient to produce an arbitrary function?*

Figure 1.11: Circuit for XOR .



1.2.2 Simple Boolean gates

Now let's study a few more Boolean gates. We already saw, AND, OR, NOT, and XOR gates.

Exercise 13. *Is* AND*reversible? How about* OR*? How about* XOR*?*

1.2.3 Universal classical computation via reversible gates

We say a Boolean function f is reversible, if there exists another Boolean function f^{-1} such that $f^{-1} \circ f$ is the identity operation. Clearly in order to have a reversible gate f needs to have equal numbers of inputs and outputs.

Example 2. *The* NOT*gate is reversible. The* SWAP*gate simply swaps the order of two bits, i.e.,* $\mathrm{SWAP}(x, y) = (y, x)$*. Controlled-NOT (*CNOT*), which copies the first bit and applies NOT to the second bit if and only if the first bit is 1. We call this first bit the control bit. In particular* $\mathrm{CNOT}(x, y) = (x, x \oplus y)$*. In all three example each gate is self inverse.*

Exercise 14. *Prove* $SWAP = CNOT_{12}CNOT_{21}CNOT_{12}$*.*

We already know that universal gate sets for non-reversible classical computation exist. But we would like a universal gate set for "reversible" classical computation. It turns out that thre exist no universal gate set solely composed of 2*-bit reversible gates. In the following exercise you will see that the controlled* NOT*gate is not universal.*

Exercise 15. *Show that the the controlled NOT gate is linear meaning* $\mathrm{CNOT}(x \oplus y) = \mathrm{CNOT}(x) \oplus \mathrm{CNOT}(y)$*, where for* $x, y \in \{0,1\}^m$*,* $x \oplus y = (x_1, \ldots, x_m) \oplus (y_1, \ldots, y_m) = (x_1 \oplus y_1, \ldots, x_m \oplus y_m)$*. Use mathematical induction to prove that any circuit composed of* CNOT *and* NOT *corresponds to a linear function. Use this observation to argue that* CNOT *and* NOT *does not correspond to a universal gate set. Finally show that there exist no universal reversible gate set using single and two qubit reversible gates.*

Next we define the 3*-bit Toffoli gate and show that this gate is universal by itself.*

TOFFOLI GATE: *The Toffoli gate* $\mathrm{TOFFOLI} : \{0,1\}^3 \to \{0,1\}^3$ *takes in* 3 *bits and outputs three bits according to* $\mathrm{TOFFOLI}(x, y, z) = (x, y, (x \wedge y) \oplus z)$*. In some ways, Toffoli is the controlled-controlled-NOT gate (i.e., it performs NOT controlled on "two" bits being set to* 1*s).*

Now we show that Toffoli gate is universal for classical computation by showing how to simulate AND, NOT, OR using a Toffoli gate.

- ***AND:*** $T(x, y, 0) = (x, y, x \wedge y)$
- ***NOT:*** $T(x, 1, 1) = (x, 1, \bar{x})$
- ***OR:*** $T(\bar{x}, \bar{y}, 1) = (\bar{x}, \bar{y}, NOT(\bar{x} \wedge \bar{y})) = (\bar{x}, \bar{y}, x \vee y)$

Another reversible gate which is universal for classical computation is the Fredkin gate, which acts as a Controlled-SWAP:

$$F(x, y, z) = \begin{cases} (x, y, z) & x = 0 \\ (x, z, y) & x = 1. \end{cases}$$

Exercise 16. *Prove the Fredkin gate is Universal.*

1.2.4 Matrix representation of classical operations

Let's start with a simple operation. What is the matrix representation of the identity function? It maps $0 \to 0$ *and* $1 \to 1$*. Its matrix is given by*

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

What is the matrix representation of the NOT operation?

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

What is the matrix representation of a constant matrix (a matrix that always outputs 0 *or* 1*)?*

$$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$$

Note this matrix is singular.
What is the matrix representation of AND?

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

We would like a systematic way of finding such representations. We introduce the "bra" notation:

$$\langle 0| = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \langle 1| = \begin{pmatrix} 0 & 1 \end{pmatrix}.$$

Using this notation, we can find the representation for AND by thinking about how the gate maps inputs to outputs. In particular,

$$|0\rangle (\langle 00| + \langle 10| + \langle 01|) + |1\rangle \langle 11| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}, \tag{1.9}$$

$$= \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{1.10}$$

How about OR?

$$|1\rangle (\langle 10| + \langle 01| + \langle 11|) + |0\rangle \langle 00| = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}, \tag{1.11}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}. \tag{1.12}$$

How about XOR?

$$|1\rangle\left(\langle 10|+\langle 01|\right)+|0\rangle\left(\langle 00|+\langle 11|\right)=\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}0 & 1 & 1 & 0\end{pmatrix}+\begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 1\end{pmatrix}, \tag{1.13}$$

$$=\begin{pmatrix}0 & 1 & 1 & 0\\1 & 0 & 0 & 1\end{pmatrix}. \tag{1.14}$$

Now let's do something more complicated. SWAP is given by

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

and CNOT is

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

We can also write the matrix form for the Toffoli gate

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

Exercise: *Write a matrix form for the Fredkin gate.*

Remark 3. *What do all of these matrices have in common?*

Answer: *Sum of columns go to* 1 *and each entry is either* 0 *or* 1*. Equivalently, sum of squares of the columns add to* 1*.*

Fact 1. *Classical reversible matrices are permutation matrices.*

1.3 From reversible operations to unitary operations

Up to this point, we saw that arbitrary classical computations can be captured using a matrix formalism. In particular, we can use a vector $|x_0\rangle$ to encode the initial state of a computation. We then use a matrix A to represent the process of the computation: the final state of the computation becomes $|x_1\rangle = A|x_0\rangle$. We then measure $|x_1\rangle$. For instance, recall the controlled-NOT operation is according to the matrix

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

we encode the two-bit states using vectors according to

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \quad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, \quad |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \quad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}.$$

Suppose the initial state is $|11\rangle$*. After the application of* $CNOT$ *we obtain:*

$$CNOT|11\rangle = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} = |10\rangle.$$

Fact 2. *Classical reversible matrices are permutation matrices.*

Matrix formulation of quantum computations–and quantum mechanics in general–is very similar: The initial state of a system, e.g., $|\psi_0\rangle$*, is evolved according to a matrix evolution like* U *to obtain a final state* $|\psi_1\rangle = U|\psi_0\rangle$*. In what follows, we explain the three main postulates of quantum formalism.*

1.3.1 Unitary evolution:

We already explained the principle of unitarity in the previous chapter. Reversible operations are a special subclass of unitary operations which do not create any superposition. As a matter of fact all the reversible operations you saw so far, SWAP, CNOT and Toffoli in particular, are quantum maps and we will use them (with precisely the matrix representations you saw) in the forthcoming sections of these notes.

1.3.2 Summary of quantum formulation and comparison with classical computations

As we discussed the matrix formulation of quantum mechanics is based on unitary mapping between unit complex vectors and performing measurements. We also saw that classical states can also be represented via vectors and operations can also be represented via matrices. In particular the state $0 \leq j \leq N-1$ *of a classical process can be represented using the ket notation as the unit vector* $|j\rangle \in \mathbb{C}^N$ *and maps between states by a matrix like* $A : \mathbb{C}^N \to \mathbb{C}^M$ *(i.e.* $A \in \mathbb{C}^{M\times N}$*) such that each* $A_{i,j} \in \{0,1\}$ *and* $\sum_{i=1} A_{i,j} = 1$*. We note the stochastic matrices are generalizations of classical deterministic processes where* $0 \leq A_{i,j} \leq 1$ *and* $\sum_i A_{i,j} = 1$*. In other words the columns of* A *are normalized with respect to the* L_1 *distance. We can view this matrix formulation of quantum mechanics as a generalization of classical matrices to complex matrices where rows and columns are normal with respect to the* L_2 *distance, and furthermore they are orthogonal to each other:* $\sum_j |A_{i,j}|^2 = \sum_j |A_{i,j}|^2 = 1$ *and* $\sum_j A^*_{ij}A_{lj} = \sum_j A^*_{ji}A_{jl} = 0$ *for all* i, k*. Reversible operations are special cases of unitary matrices where* $A_{i,j} \in \{0,1\}$*.*

	Classical computation	*Stochastic computation*	*Quantum Computation*
States:	$\|v\rangle \in \{0,1\}^m, \sum_i v_i = 1$	$\|p\rangle \in \mathbb{R}^m_{\geq 0}, \sum_j p_j = 1$	$\|\psi\rangle \in \mathbb{C}^m, \sum_i \|\psi_i\|^2 = 1$
Evolutions:	$A \in \{0,1\}^{m\times n}, \sum_i A_{ij} = 1$	$P \in \mathbb{R}_{\geq 0}{}^{m\times n}, \sum_i P_{ij} = 1$	$U \in \mathbb{C}^{m\times m}, U^\dagger U = I$
Output:	$\|w\rangle = A\|v\rangle$	$\|q\rangle = P\|p\rangle$	$\|\phi\rangle = U\|\psi\rangle$
Measurements:	*The nonzero element of* $\|w\rangle$	*Sample* j *with prob* p_j	*Sample* j *with prob* $\|\psi_j\|^2$

1.3.3 Remark on Measurement and Post-Measurement States

Suppose

$$|\psi\rangle_{ABC} = \sqrt{\tfrac{1}{5}}\,|001\rangle_{ABC} + \sqrt{\tfrac{2}{5}}\,|010\rangle_{ABC} - \sqrt{\tfrac{2}{5}}\,|011\rangle_{ABC}$$

is a quantum state consisting of three parts. We wish to measure the first two qubits and compute the probability of obtaining the outcome 01 *on qubits A and B, as well as the corresponding post-measurement state on system C.*

We can rewrite the state as

$$|\psi\rangle = |01\rangle_{AB}\big(\sqrt{\tfrac{2}{5}}\,|0\rangle_C - \sqrt{\tfrac{2}{5}}\,|1\rangle_C\big) + \dots$$

where "..." denotes the terms corresponding to other outcomes on qubits AB.

The probability of measuring AB to be 01 *is given by the squared norm of the unnormalized post-measurement vector:*

$$\big\|\sqrt{\tfrac{2}{5}}\,|0\rangle_C - \sqrt{\tfrac{2}{5}}\,|1\rangle_C\big\|^2 = \tfrac{4}{5}.$$

The (normalized) post-measurement state on C is therefore

$$\frac{1}{\sqrt{2}}\big(|0\rangle_C - |1\rangle_C\big) = |-\rangle_C.$$

Now suppose we wish to measure the first qubit of

$$|\phi\rangle_{AB} = \sqrt{\tfrac{1}{3}}\,|00\rangle - \sqrt{\tfrac{2}{3}}\,|11\rangle$$

in the $\{|+\rangle, |-\rangle\}$ *basis, where*

$$|0\rangle = \tfrac{1}{\sqrt{2}}\big(|+\rangle + |-\rangle\big), \quad |1\rangle = \tfrac{1}{\sqrt{2}}\big(|+\rangle - |-\rangle\big).$$

Substituting, we obtain

$$|\phi\rangle_{AB} = |+\rangle\big(\sqrt{\tfrac{1}{6}}\,|0\rangle - \sqrt{\tfrac{2}{6}}\,|1\rangle\big) + |-\rangle\big(\sqrt{\tfrac{1}{6}}\,|0\rangle + \sqrt{\tfrac{2}{6}}\,|1\rangle\big).$$

Hence, the probability of obtaining the + outcome on the first qubit is

$$\tfrac{1}{6} + \tfrac{2}{6} = \tfrac{1}{2},$$

and the post-measurement state (unnormalized) corresponding to the + outcome is

$$\sqrt{\tfrac{1}{3}}\,|0\rangle - \sqrt{\tfrac{2}{3}}\,|1\rangle.$$

An equivalent procedure is to apply a Hadamard gate to the first qubit and then measure it in the computational basis, interpreting outcome 0 *as "+" and outcome* 1 *as "−". This equivalence follows from the following general principle.*

Proposition 1 (Change of Measurement Basis)**.** *Let* $\mathcal{V} = \{|v_j\rangle\}_j$ *be an orthonormal basis related to the computational basis* $\{|j\rangle\}$ *by a unitary change of basis V, i.e.* $|v_j\rangle = V|j\rangle$*. Then, for any quantum state* $|\psi\rangle$*,*

$$\Pr(\textit{measure } v_j \textit{ in basis } \mathcal{V}) = |\langle v_j|\psi\rangle|^2 = |\langle j|V^{-1}|\psi\rangle|^2.$$

In words, measuring $|\psi\rangle$ *in the basis* $\mathcal{V}$ *is equivalent to applying* V^{-1} *to* $|\psi\rangle$ *and measuring in the computational basis.*

1.4 Quantum Computation

Quantum computing consists of three steps: Initialization, applying a quantum circuit, and measurement. The standard model of quantum computing is based on quantum circuits. We start with all zeros states, apply a quantum circuit from a universal gate set, and measure the very first qubit. If it was 1 *accept otherwise reject.*

1.4.1 Basic quantum gates

Similar to classical computing, in which we decompose a large computation into a composition of small gates, we can decompose an arbitrary unitary matrix into smaller gates.

- *Pauli gates*

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

- *Hadamard gate*

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

- *The T gate. (Becomes important later).*

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

- *The following is known as the phase gate*

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

- *The following relationships hold between these operators*

$$\{ X, Y \} = \{ X, Z \} = \{ Y, Z \} = 0$$

$$\frac{1}{2}[X, Y] = iZ, \frac{1}{2}[Y, Z] = iX, \frac{1}{2}[Z, X] = iY$$

$$HXH = Z, HZH = X, S^2 = Z.$$

- *Other examples include classical reversible circuits:*

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

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

1.4.2 Universal quantum gate sets

Similar to classical computing, we can talk about a set of universal gates.

Definition 1. *A quantum gate set is universal if it can approximate any unitary operation within arbitrary precision. More precisely, G is universal if for any $N = 2^n$ and any $U \in \mathcal{U}(N)$ and any $\epsilon > 0$ there exist a finite sequence $g_1, \ldots, g_T \in G$ such that $\|U - g_T \ldots g_1\|_\infty \leq \epsilon$.*

In the definition above, we measured the distance between the unitary using the infinity or operator norm $\|\cdot\|_\infty$ defined as the largest singular value. We could define the distance using different metrics; however, in finite dimensions, universality in one norm implies universality in other norms. The choice of metric would matter if we talked about efficiency, which is not what universality is concerned with. Here we have defined $\mathcal{U}(2^n)$ to be the unitary group over n qubits. We note that for practical purposes, it is sufficient if our gateset captures the unitary up to a global phase. As a result, instead of $\mathcal{U}(N)$, we often use the special unitary group $\mathcal{SU}(N)$, which is the group of unitary operations with a unit determinant.

Recall that there are 2^{2^n} Boolean functions over n bits. We note that the space of n qubit unitary matrices is even more gigantic. So, to approximate an arbitrary unitary matrix, we may need an exponentially long quantum circuit. Can we produce an arbitrary quantum operation exactly? The answer is no. Because arbitrary operations include arbitrary real numbers. But for all plausible applications, an approximation is sufficient.

- *CNOT and arbitrary rotation*
- *Clifford + T; where Clifford = {CNOT, H, S}*
- *Hadamard, Toffoli*

Simple example for a universal gate set: *Consider* 1 1*-dimensional quantum states: all quantum states that are a scalar multiple of one vector, e.g., $|0\rangle$. Such a quantum state is only a phase $e^{i\phi}|0\rangle$. Consider a phase quantum gate that applies a phase $G = e^{i\theta}$. If θ is a rational multiple of π (e.g., $2\pi, \pi/2, \pi/7$, etc.), by applying G any number of times, we are not able to produce arbitrary phases. Based on a well-known theorem in mathematical analysis called Gibbs equidistribution, if θ is an irrational multiple of π, then we can estimate any angle up to arbitrary precision by repeating G.*

The Solovay-Kitaev Theorem

Universality is a statement about the capability of compiling one gate set into another; it does not, however, capture the efficiency of such compilation. Due to a seminal result by Solovay and Kitaev, we know that one can compile a quantum circuit based on one gate set into another gate set with minimal overhead. In particular, let G and G' be two universal gate sets for a single qubit. There is a constant c such that for any gate U in G, there is a sequence of gates in G' with length $\log^c(1/\epsilon)$, which approximates U. The polylogarithmic bound is significant because if we cover the unitary group with balls of radius ϵ, we need at least polynomially many balls in $1/\epsilon$, and one expects the compiling bound to grow polynomially. This polynomial-to-polylogarithmic improvement result enables us to efficiently compile quantum computation over one gate set into another. A similar polylogarithmic bound holds for multi-qubit compilations.

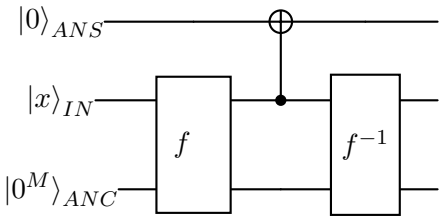
1.4.3 Uncomputing

Due to the reversibility of quantum mechanics in quantum computations, the number of output qubits is equal to the number of input qubits. As a result, the output state in a quantum computation is fundamentally a multi-qubit state. Furthermore, the use of ancillary qubits is inevitable. Suppose in a quantum computation (intended to solve a decision problem) we start with input $|x\rangle |0^M\rangle$ *input followed by several ancilla qubits and perform a unitary computation and obtain the output quantum state* $|\psi_{out}\rangle = \sqrt{1-\epsilon_x}\,|B(x)\rangle\,|\phi(x)\rangle + \sqrt{\epsilon_x}\,|\phi'(x)\rangle$, *for* $\epsilon_x \ll 1$ *and some some normalized quantum state* $|\phi'(x)\rangle$. *Upon measuring the first qubit with high* $(1-\epsilon_x)$ *probability, we obtain the answer* $B(x) \in \{0,1\}$, *and we can toss out the remaining qubits. Now, suppose we want to use this procedure as a subroutine within another quantum computation. The main issue is that now the input to the subroutine is some superposition over input states, i.e., a quantum state like* $\sum_x \alpha_x |x\rangle$, *as a result, the output of the subroutine will be*

$$\sum_x \alpha_x(\sqrt{1-\epsilon_x}\,|B(x)\rangle\,|\phi(x)\rangle + \sqrt{\epsilon_x}\,|\phi'(x)\rangle)$$

Now, suppose we want to measure the contents of the answer register. Due to the entanglement between the answer register and the remaining qubits, the reduced density matrix over the answer qubit can be extremely mixed, and you may not learn anything useful. This is an interesting turn of events, as we usually think that entanglement is a major source of speedup, and having more entanglement will always help. However, this example shows us that in this case, entanglement will have a destructive effect.

How can we decouple the answer qubit from the rest of the qubits? In reversible classical computation, uncomputing is the result that for any reversible classical computation $f(x, 0^M) = (B(x), y)$, *where* $B(x)$ *is the bit we want to learn and* 0^M *are ancillae, we can define another reversible computation* g *such that* $g(0, x, 0^M) = g(B(x), x, 0^M)$. *To perform this we use the following circuit:*



In this procedure we first implement f, *then use CNOT to copy the answer bit to a register reserved for this purpose. We then undo this procedure by applying* f^{-1}. *This procedure is called uncomputing. It turns out that we can perform the same procedure to perform uncomputing. To do this we just replace* f *with* U, *the unitary used to implement the quantum computation. In particular,*

Theorem 1. *Let* U *be the quantum circuit for a quantum computation which on input* x *reveals output bit* $B(x)$ *with probability at least* $1-\epsilon$. *The uncomputing procedure resulted from replacing* f *with* U *inf section 1.4.3 applied to the input* $\sum_x \alpha_x |x\rangle$ *results in the output* $|\phi'_{out}\rangle$ *such that*

$$\| \, |\phi'_{out}\rangle - \sum_x \alpha_x |x\rangle\,|B(x)\rangle\,|0^M\rangle \, \| \leq O(\sqrt{\epsilon})$$

Proof. We know

$$U(|x\rangle\,|0^M\rangle) = \sqrt{1-\epsilon_x}\,|B(x)\rangle\,|\phi(x)\rangle + \sqrt{\epsilon_x}\,|\phi'(x)\rangle$$

Therefore by applying CNOT and U^{-1} we get

$$|\phi'_{out}(x)\rangle = \sqrt{1-\epsilon_x}\,|B(x)\rangle\, U^{-1}(|B(x)\rangle\,|\phi(x)\rangle) + \sqrt{\epsilon_x}\,|\phi''(x)\rangle$$

Where $|\phi''(x)\rangle = U^{-1}CNOT(|0\rangle\,|\phi'(x)\rangle)$. We now observe that $\|\,|\phi''(x)\rangle\,\| = 1$. Furthermore

$$U^{-1}(|B(x)\rangle\,|\phi(x)\rangle) = (1-\epsilon_x)^{-1/2}\,|x\rangle\,|0^M\rangle - \sqrt{\frac{\epsilon_x}{1-\epsilon_x}}U^{-1}(|\phi'(x)\rangle)$$

Therefore

$$|\phi'_{out}(x)\rangle = |B(x)\rangle\,|x\rangle\,|0^M\rangle - \sqrt{\epsilon_x}\,|B(x)\rangle\, U^{-1}(|\phi'(x)\rangle) + \sqrt{\epsilon_x}\,|\phi''(x)\rangle$$

Now if we consider the full superposition

$$\begin{aligned} |\phi'_{out}\rangle &:= \sum_x \alpha_x\,|\phi'_{out}(x)\rangle \\ &= (\sum_x \alpha_x\,|B(x)\rangle\,|x\rangle)\,|0^M\rangle + |\phi'''\rangle \end{aligned} \tag{1.15}$$

where $\|\,|\phi'''\rangle\,\| \le \max_x \sqrt{\epsilon_x}$. □

1.4.4 Probability of error

Due to the probabilistic nature of quantum formalism, we inevitably have to define a model of computation that succeeds only a fraction of times in producing the correct answer; in other words, we have a probabilistic model of computation. We say that the given quantum algorithm succeeds in performing reliable computation if it gives the correct answer at least $2/3$ *of the time. The choice of* $2/3$ *is arbitrary. We can always amplify the probability of success to any number (e.g.,* 0.9999*) by taking the majority's vote: we repeat the computation multiple times and choose the majority of the answers as the final answer. How does the majority vote help us? Imagine a coin that produces outcome* 0*,* $2/3$ *of the time. If we flip this coin* n *times, it produces* k *heads with probability* $\binom{n}{k}(2/3)^k(1/3)^{n-k}$*.*

Exercise: *Show that the probability that more than* $n/2$ *of the outcomes is bit* 1 *is exponentially small.*

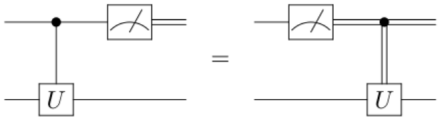
Lemma 1. *We can perform a majority vote by only one measurement in the end.*

Proof. We prepare multiple copies of the quantum experiment and utilize a reversible implementation of the majority vote to store the result in a single bit. To be specific, let $|\phi_{out}\rangle$ be the quantum state at the output of a single copy of the circuit. Let p_0 and p_1 be, respectively, the marginal probability distribution over the answer qubit revealing 0 or 1. Then $|\phi_{out}\rangle = \sqrt{p_0}\,|0\rangle\,|\phi_0\rangle + \sqrt{p_1}\,|1\rangle\,|\phi_1\rangle$, where $|\phi_j\rangle$ is a normalized quantum state over the rest of the qubits. We now prepare N independent copies of $|\phi_{out}\rangle$, i.e.,

$$|\phi_{out}\rangle^{\otimes N} = \sum_{x\in\{0,1\}^N} \sqrt{p_{x_0}\cdots p_{x_N}}\,|x\rangle \otimes |\phi_x\rangle$$

where $|x\rangle = |x_1 \ldots x_N\rangle$ and $|\phi_x\rangle = \bigotimes_{i=1}^N |\phi_{x_i}\rangle$ Now let MAJ be the reversible implementation of the N-bit majority using M ancilla qubits and uncomputing the answer in the end, i.e.,

Figure 1.12: Principle of deferred measurements



$MAJ(|0\rangle |x\rangle |0^M\rangle) = |MAJ(x)\rangle |x\rangle |0^M\rangle$. Therefore, upon applying MAJ to $|0\rangle\otimes|\phi_{out}\rangle^{\otimes N}\otimes|0^M\rangle$ we obtain the quantum state

$$|\phi'_{out}\rangle = \sum_{x\in\{0,1\}^N} \sqrt{p_x}\, |MAJ(x)\rangle\, |\tilde{\phi_x}\rangle$$

where we have defined $p_x = p_{x_1} \dots p_{x_N}$, and $|\tilde{\phi_x}\rangle = |\phi_x\rangle \otimes |x\rangle \otimes |0^M\rangle$. Therefore the probability of obtaining 1 from the marginal probability distribution over the answer bit of $|\tilde{\phi}'_{out}\rangle$ is

$$q_1 = \sum_{x:MAJ(x)=1} p_x$$

which is the same value as taking the majority vote classically.

□

Without loss of generality, we can always assume you start with $|0 \dots 0\rangle$.

1.4.5 Extra features

Lemma 2 (Deferred measurements)**.** *You can always push the measurements to the end.*

Proof. We can eliminate intermediate measurements using the gadget in Figure 1.12. □

Lemma 3. *We can simulate arbitrary quantum computations with gates composed of real numbers only.*

Proof. We can show that by merely doubly the dimension of the Hilbert space we can simulate aribtrary quantum computations using complex amplitudes using a formulation using real amplitudes only (i.e. orthogonal matrices instead of unitary matrices). In particular, we map a quantum state $|\psi\rangle = \sum_{j=0}^{N-1} \alpha_j |j\rangle \in \mathbb{C}^N$, with a quantum state $|\psi\rangle = \sum_{j=0}^{N-1} \alpha_j^{(re)} |j^{(re)}\rangle + \sum_{j=0}^{N-1} \alpha^{(im)} |j^{(im)}\rangle \in \mathbb{R}^{2N}$ where $\alpha^{(re)} = \Re(\alpha_j)$ and $\alpha^{(im)} = \Im(\alpha_j)$, are the real and imaginary parts of α_j, and $\{|j^{(re)}\rangle, |j^{(im)}\rangle\}$ correspond to an orthonormal basis for $\mathbb{R}^{2N}$. Next we replace any unitary matrix in $\mathbb{C}^{N\times N}$ with an orthonormal matrix $O \in \mathbb{R}^{2N\times 2N}$ according to the following rule. We replace each complex entry of U_{jk} with the 2×2 block $U_{jk}^{(re)} I + U_{jk}^{(im)} S$ where

$$S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

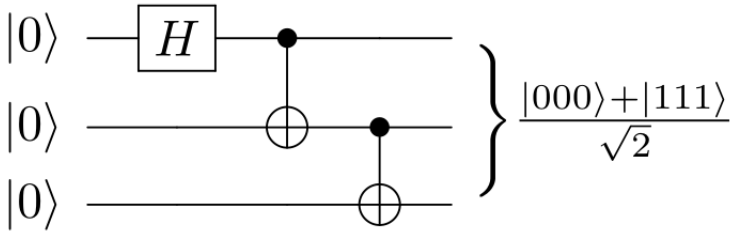
In other words, we replace i with S and 1 with the identity I. □

1.4.6 Multi-qubit systems:

Tensor product of operations

Next, we talk about the tensor product of operations on different parts of a system.

Figure 1.13: Quantum Circuit for the GHZ state



- *Tensor product of operators:* $A \otimes B$.

$$A \otimes B = \begin{pmatrix} & .. & \\ .. & A_{ij}B & .. \\ & .. & \end{pmatrix}$$

  - $(A + A') \otimes B = A \otimes B + A' \otimes B$
  - $(\alpha A) \otimes B = \alpha(A \otimes B)$
  - $(A \otimes B)(A' \otimes B') = AA' \otimes BB'$
  - *Example* $(X \otimes X)(|0\rangle\langle 1| \otimes H)$

- *Tensor product of operation* $(A \otimes B)(|a\rangle \otimes |b\rangle) = (A|a\rangle \otimes B|b\rangle)$

Example 3. *Apply* $H \otimes H$ *to* $|11\rangle$. *Then, apply CNOT. Then SWAP. Then apply the phase gate on the first qubit.*

Example 4. *Consider the three-qubit state* $|000\rangle$. *If we apply Hadamard to the first qubit, we obtain:* $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |00\rangle$. *Now if we apply* $CNOT_{1,2}$ *we obtain* $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes |0\rangle$. *If we next apply* $CNOT_{2,3}$, *we obtain* $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$. *This state is known as the GHZ state. See Figure 1.13*

1.4.7 Hadamard test

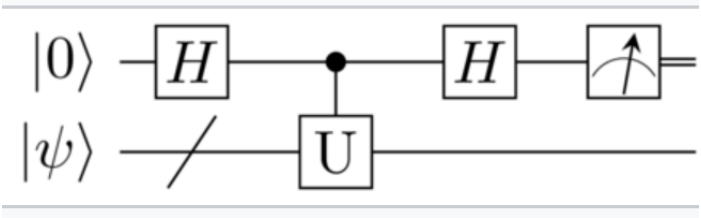
Hadamard test is an important quantum subroutine to measure amplitudes of quantum experiments. Suppose we have a quantum experiment according to the unitary matrix U. *We also have a quantum state* $|\psi\rangle$ *stored in a quantum memory. Our objective is to estimate the amplitude* $\langle \psi, U\psi \rangle$. *Similar to CNOT, we can implement controlled-U operations. See Figure 1.14. In particular, if the control bit is set to* $|0\rangle$, *then the gate applies identity to the second bit, and if the control bit is set to* $|1\rangle$, *it applies* U. *Using the tensor product notation, the controlled U operations is* $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$.

Now we analyze the circuit in Figure 1.14. We start with $|0\rangle|\psi\rangle$. *After the application of the first Hadamard we obtain* $\frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes |\psi\rangle$. *Next we apply controlled U and obtain* $\frac{|0\rangle|\psi\rangle+|1\rangle U|\psi\rangle}{\sqrt{2}}$. *Next we apply Hadamard another time and obtain:*

$$\frac{|0\rangle|\psi\rangle + |1\rangle|\psi\rangle + |0\rangle U|\psi\rangle - |0\rangle U|\psi\rangle}{2} = |0\rangle(\frac{I+U}{2})|\psi\rangle + |1\rangle(\frac{I-U}{2})|\psi\rangle$$

What is the probability of obtaining 0 *from this experiment? We now tell you an important rule. Suppose we have quantum state that can be written as* $|0\rangle|\phi_0\rangle + |1\rangle|\phi_1\rangle$. *We can show that the probability of obtaining* 0 *is* $\langle \phi_0, \phi_0 \rangle$ *and the probability of obtaining* 1 *is* $\langle \phi_1, \phi_1 \rangle$.

Figure 1.14: The Hadamard circuit



Exercise 17. *Prove this!*

Therefore with probability $\frac{1}{4}\langle\psi|(I+U^\dagger)(I+U)|\psi\rangle = \frac{1}{2}(1+Re(\langle\psi|U|\psi\rangle))$ *we obtain* 0 *and with probability* $\frac{1}{4}\langle\psi|(I-U^\dagger)(I-U)|\psi\rangle = \frac{1}{2}(1-Re(\langle\psi|U|\psi\rangle))$ *we obtain* 1*. We claim that if we output* $X=+1$ *upon measuring* 0 *and* $X=-1$ *upon measuring* 1*, then the expectation value of* X *will be* $Re(\langle\psi|U|\psi\rangle)$*. To see this let* $\alpha = Re(\langle\psi|U|\psi\rangle)$*. Then* $Pr(X=0) = \frac{1+\alpha}{2}$ *and* $Pr(X=1) = \frac{1-\alpha}{2}$*. Therefore* $E(X) = Pr(X=0) - Pr(X=1) = \alpha$*. To recover* α*, we repeat the experiment a few times to obtain* $X_1, \ldots, X_k$*. We then compute* $\frac{X_1+\ldots+X_k}{k}$ *and output it as an estimation to* α*. From basis probability theory we know that the probability of error decays as* $O(\frac{1}{\sqrt{k}})$*. So, in order to make the error* ϵ*, we need to repeat the experiment at least* $\Omega(1/\epsilon^2)$*.*

This experiment yields the real part of the amplitude $\langle\psi|U|\psi\rangle$*. To obtain the imaginary part we can design a similar experiment. We leave it as an exercise for your practice.*

Exercise 18. *Design a quantum algorithm to estimate the imaginary part of a quantum amplitude.*

Multi-qubit Pauli strings

An important class of multi-qubit operations are Pauli strings. A Pauli string is a tensor product of Pauli operators such as $X\otimes X$*,* $X\otimes Y$*,* $I\otimes Y$*, etc. In general, any operator like* $P_1\otimes\ldots\otimes P_n$ *for* $P_i\in\{I,X,Y,Z\}$ *is a Pauli string.*

Recall the Clifford gate set is the group of operators that can be generated by $S, H, CNOT$*. We know that this gate set generates the Pauli string. To se this we note that* $Z=S^2$*,* $X=HZH$ *and* $Y=iZX$*. We can show that the set of Pauli strings (up to a global phase) is closed under conjugation by Pauli strings. We can see this using the following exercise.*

Exercise 19. *The conjugation of a matrix* A *by another matrix* B *is according to* BAB^{-1}*. In this exercise, we want to study the conjugation of Pauli strings by Clifford circuits. Let* P *be the tensor product of any two Pauli strings and let* C *be any Clifford operations. Show that, up to a global phase,* CPC^{-1} *is a Pauli string. (Hint: You only need to show this for* XI, ZI, IX, IZ*, because these four operators generate all Pauli strings. Can you see why that is enough?)*

Remark 4. *Mathematically speaking, the Clifford group is the normalizer of the Pauli group. The Pauli group is the group of Pauli strings under multiplication. The Pauli group* $\mathcal{P}$ *includes four copies of each Pauli string, one for either of the four global phases* $1, -1, i$ *or* $-i$*. In other words,* $\mathcal{P} = \{e^{im\pi/2}P_1\otimes\ldots\otimes P_n : P_i\in\{I,X,Y,Z\}, m=0,1,2,3\}$*. We previously discussed in class that according to a well-known theorem of Gottesman and Knill, the Clifford circuits are classically simulable. The proof of this theorem heavily relies on the Clifford group being the normalizer of the Pauli group. We will discuss more details about this theorem in future lectures.*

1.4.8 No cloning theorem

Now that we learned about multi-qubit systems, we can ask a basic question: can we use quantum operations to produce multiple copies of a quantum state? One of the main features of classical computing is the capability to copy quantum states. Can we copy quantum states? In particular, is there a quantum operation to clone an arbitrary quantum state $|\psi\rangle$: $|\psi\rangle|0\rangle \to |\psi\rangle|\psi\rangle$? It turns out the answer is no. For instance, suppose you have a quantum state, and you don't know if it is the state $|0\rangle$ or $|+\rangle$. How can we copy this quantum state if it is either of the two incompatible bases? There is a simple reason to see why quantum operations are not capable of cloning quantum states. The reason is simply that cloning is a nonlinear operation, and quantum mechanics is a linear theory.

We can show the no-cloning theorem in a different way. Suppose there existed a cloning quantum operation U such that $U(|\psi\rangle \otimes |0\rangle) = (|\psi\rangle \otimes |\psi\rangle)$. For any other state $|\phi\rangle$ we have $U(|\phi\rangle \otimes |0\rangle) = (|\phi\rangle \otimes |\phi\rangle)$. What is the inner product of the two sides of each equation? $(\langle\psi| \otimes \langle 0|)U^\dagger U(|\phi\rangle \otimes |0\rangle) = (\langle\psi| \otimes \langle\psi|)(|\phi\rangle \otimes |\phi\rangle)$. Hence $\langle\psi, \phi\rangle = |\langle\psi, \phi\rangle|^2$. Which is a contradiction (for instance, let $|\psi\rangle = |0\rangle$ and $|\phi\rangle = |+\rangle$). However, cloning is possible only on an orthonormal basis (which is not more powerful than classical computations).

1.5 Non-classical correlations

If a quantum state is the tensor product of two quantum states, we call the quantum state separable. In general, quantum states are not separable. We call such a quantum state "entangled". For instance, let $|\psi_1\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|\psi_2\rangle = \gamma|0\rangle + \delta|1\rangle$. The quantum state $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ is a separable quantum state. If we expand $|\psi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$. In general, let α_x be the amplitude corresponding to $|x\rangle$. We see that if these amplitudes satisfy $\alpha_{00}\alpha_{11} \neq \alpha_{01}\alpha_{10}$, then the quantum state is entangled. There is a sense to say a quantum state is maximally entangled. Consider the example of EPR pairs below.

- *EPR pair* $|\Phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

Exercise: *Show that the EPR pair is not separable.*

EPR pairs satisfy the following surprising relationships

- *EPR pair* $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|++\rangle + |--\rangle)$
- $|00\rangle + |11\rangle = |\theta\uparrow, \theta\uparrow\rangle + |\theta\downarrow, \theta\downarrow\rangle$

Here

- $|\theta\uparrow\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle$
- $|\theta\downarrow\rangle = -\sin\theta|0\rangle + \cos\theta|1\rangle$

In general for any square matrix M we have $(M \otimes I)|\Phi\rangle = (I \otimes M^T)|\Phi\rangle$.

1.5.1 No signaling and density matrix

We saw that we can write the EPR state (shared between Alice and Bob) in two ways $\frac{|00\rangle_{AB} + |11\rangle_{AB}}{\sqrt{2}} = \frac{|++\rangle_{AB} + |--\rangle_{AB}}{\sqrt{2}}$. Imagine Alice wants to use this equivalence of representations to send a bit of

information to Bob (who is far away). If Alice wants to send a 0 bit, she will measure her quantum state in the $\{|0\rangle, |1\rangle\}$ basis, and if she wants to send bit 1, she makes a measurement in the $\{|+\rangle, |-\rangle\}$ basis. In the former case, the state at Bob's disposal will be a uniform mixture of $|0\rangle$ and $|1\rangle$, and in the latter it will be a uniform mixture of $|+\rangle$ and $|-\rangle$. Now, if Bob could distinguish the two cases apart, he could decode Alice's bit (in other words, Alice would succeed in sending a bit to Bob instantly). However, we claim that the two ensembles are statistically indistinguishable. To see this, we need to define the notion of a density matrix.

Definition 2. *For an ensemble of quantum states $\mathcal{E} = \{(p_1, |\psi_1\rangle), (p_2, |\psi_2\rangle), \ldots, (p_N, |\psi_N\rangle)\}$ (i.e., we either have quantum state $|\psi_1\rangle$ with probability p_1, or $|\psi_2\rangle$ with probability p_2, and so on), we define the corresponding density matrix.*

$$\rho = p_1 |\psi_1\rangle \langle\psi_1| + \ldots + p_N |\psi_N\rangle \langle\psi_N|$$

We note that the definition above states $|\psi_j\rangle$ need not be orthonormal and N might be smaller, equal, or larger than the dimension of the Hilbert space.

Exercise 20. *Show that density matrices satisfy the following features*

1. *Given a density matrix ρ, $Tr(\rho) = 1$ and $\rho \geq 0$.*
2. *Convex combination of density matrices is a density matrix, i.e., given matrices $\rho_1 \geq 0, \ldots, \rho_N \geq 0$ with unit trace and probabilities $p_1 + \ldots + p_N = 1$, $\rho = p_1\rho_1 + \ldots + p_N\rho_N \geq 0$ and $Tr(\rho) = 1$.*

We can show that

Lemma 4. *If we measure the ensemble $\mathcal{E}$ according to a POVM $\{M_x\}$, then the probability of obtaining label x is $Tr(M_x\rho)$.*

Proof. Upon measuring $\mathcal{E}$ with probability p_j we have $|\psi_j\rangle$ which reveals x with probability $\langle\psi_j| M_x |\psi_j\rangle = Tr(|\psi_j\rangle \langle\psi_j| M_x)$. Therefore, the total probability of obtaining x is $Tr(\rho M_x)$. □

In other words, if the density matrix for two ensembles is the same, then there is no quantum experiment to distinguish them. We now show that the ensemble $\mathcal{E}_1 = \{(1/2, |0\rangle), (1/2, |1\rangle)\}$ is indistinguishable from $\mathcal{E}_2 = \{(1/2, |+\rangle), (1/2, |-\rangle)\}$, implying that Bob cannot receive a signal from Alice. Let ρ_j be the density matrix for $\mathcal{E}_j$. Then

$$\rho_1 = \frac{1}{2}(|0\rangle \langle 0| + |1\rangle \langle 1|) = \frac{1}{2}(|+\rangle \langle +| + |-\rangle \langle -|) = \rho_2$$

Therefore $\rho_1 = \rho_2 = I/2$. This special density matrix is known as the maximally mixed state. Information theoretically, it contains no bit of information. As a matter of fact, the uniform mixture of states from any orthonormal basis is a maximally mixed state.

Exercise 21. *Let $|\psi\rangle_{AB}$ be a bipartite quantum state, and $\mathcal{M}_A = \{M_1^A, \ldots, M_N^A\}$ a POVM on system A. According to quantum formalism we can show that upon implementing $\mathcal{M}_A$ and obtaining label y the quantum states on the B part will collapse to $|\psi_y\rangle_B = \frac{(M_y^A \otimes I_B)|\psi\rangle_{AB}}{\|(M_y^A \otimes I_B)|\psi\rangle_{AB}\|}$. Now let ψ be any quantum state. Show that there is no quantum experiment B can perform to distinguish between the two scenarios: (1) A has performed POVM $\mathcal{M}_A$, (2) A hasn't done anything.*

1.5.2 Quantum teleportation

One of the main implications of entanglement is quantum teleportation: We can destroy a quantum state at one point in space and recreate it somewhere else if we are allowed to send a few classical bits. Here is how it is done. First, observe that the following states are a complete basis for two quantum bits.

$$|\Phi_\pm\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle), \quad |\Psi_\pm\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |01\rangle).$$

We can show that these quantum states form a complete basis by:

$$|\Phi_+\rangle\langle\Phi_+| + |\Phi_-\rangle\langle\Phi_-| + |\Psi_+\rangle\langle\Psi_+| + |\Psi_-\rangle\langle\Psi_-| = I$$

It is useful to note

$$|00\rangle = \frac{1}{\sqrt{2}}(|\Phi_+\rangle + |\Phi_-\rangle), \quad |11\rangle = \frac{1}{\sqrt{2}}(|\Phi_+\rangle - |\Phi_-\rangle),$$

$$|01\rangle = \frac{1}{\sqrt{2}}(|\Psi_+\rangle + |\Psi_-\rangle), \quad |10\rangle = \frac{1}{\sqrt{2}}(|\Psi_+\rangle - |\Psi_-\rangle).$$

The teleportation protocol is as follows: Alice wishes to send a quantum bit $|\psi\rangle$ to Bob; A be a register at Alice's side and B be a register at Bob's, where he wishes to receive the quantum state. She stores $|\psi\rangle$ in a separate register C and shares an EPR state ($|\Phi_+\rangle_{AB}$) with Bob. Alice measures (destroys) the registers AC in according to POVM Q. Let $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. With probability 1/4 the content of ABC will be either of the following:

$$|\Phi_+\rangle_{AC} \otimes (\alpha|0\rangle + \beta|1\rangle)_B, \quad |\Phi_-\rangle_{AC} \otimes (\alpha|0\rangle - \beta|1\rangle)_B,$$

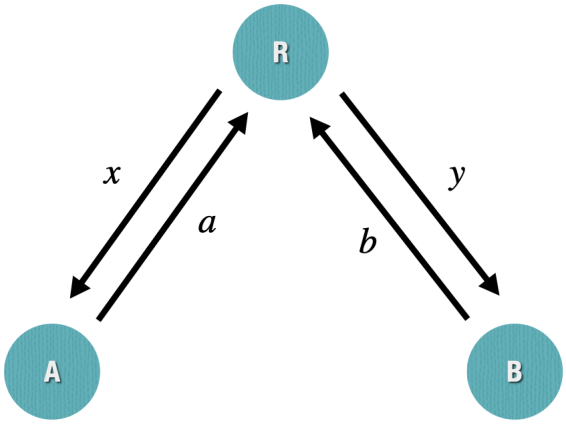
$$|\Psi_+\rangle_{AC} \otimes (\alpha|1\rangle + \beta|0\rangle)_B, \quad |\Psi_-\rangle_{AC} \otimes (\alpha|1\rangle - \beta|0\rangle)_B,$$

We observe that in either of these cases Bob has received $|\psi\rangle$ up to some error. The last step is for Bob to correct this error. Here is how he can do it. Alice selects two classical bits b_X, b_Z. Upon measuring $|\Phi_+\rangle_{AC}$ she sets $b_X = 0, b_Z = 0$; upon measuring $|\Phi_-\rangle_{AC}$ she sets $b_X = 0, b_Z = 1$; upon measuring $|\Psi_+\rangle_{AC}$ she sets $b_X = 1, b_Z = 0$; upon measuring $|\Psi_-\rangle_{AC}$ she sets $b_X = 1, b_Z = 1$. She sends b_X, b_Z to Bob. Bob corrects the error by applying $Z^{b_Z}X^{b_X}$. In conclusion, by sending two classical bits, and sharing an entangled state, Alice can send 1 quantum bit to Bob.

1.5.3 Positive Operator Valued Measures:

The set of projectors $Q = \{|\Phi_+\rangle\langle\Phi_+|, |\Phi_-\rangle\langle\Phi_-|, |\Psi_+\rangle\langle\Psi_+|, |\Psi_-\rangle\langle\Psi_-|\}$ partition the space of two-qubit states into four equal partitions. Mathematically we say Q forms a Positive Operator-Valued Measure (POVM). A POVM is a set of PSD matrices $\Pi_1, \ldots, \Pi_k$ such that $\Pi_1 + \ldots + \Pi_k = I$. POVM is a generalization of probability vectors to matrices. If the state of a quantum system is $|\psi\rangle$ and define $p_i := \langle\psi|\Pi_i|\psi\rangle$, then $p_i \geq 0$ and $\sum_i p_i = 1$. So, for any quantum state, we can define a probability vector. In quantum mechanics, we can assign a POVM for any measurement. For instance, the POVM for measuring a qubit in the computational basis is $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$. Similarly $Q = \{|\Phi_+\rangle\langle\Phi_+|, |\Phi_-\rangle\langle\Phi_-|, |\Psi_+\rangle\langle\Psi_+|, |\Psi_-\rangle\langle\Psi_-|\}$ is the POVM corresponding to measuring in the $|\Phi_\pm\rangle, |\Psi_\pm\rangle$ basis.

Figure 1.15: The description of the CHSH game



1.5.4 CHSH game

We saw that quantum superposition has a probabilistic behavior. For instance, if we measure the $|+\rangle$ *in the* $|0\rangle, |1\rangle$ *basis, we obtain* 0 *or* 1*, each with probability* 1/2*. What is the value of the quantum state before we measure this quantum state? Local realism is the idea that physical systems have definite properties independent of measurement and that these properties can influence the outcomes of measurements in a local manner. All classical computations are based on the local realism framework. The CHSH inequality, named after its discoverers Clauser, Horne, Shimony, and Holt, is a fundamental result in quantum mechanics that demonstrates a violation of local realism.*

Instead of the inequality, we will present a certain nonlocal quantum mechanical "game" that is inspired by the CHSH inequality. We show how the predictions of this game violate local realism. This game has been performed experimentally, confirming the violation of local realism. The outcome of this experiment indicates that the nature of correlations in quantum mechanics is fundamentally different from our classical intuition. We note that while this experiment violates "local" realism, there might be "non-local" hidden variables that justify the probabilistic nature of quantum mechanics. However, the experiment does refute local hidden variables.

Game:

- *R gives A and B two bits* x, y
- *they don't interact*
- *they output* a *and* b *back to R.*
- *they win if* x *AND* $y = a \oplus b$*.*

Exercise 22. *Show that classical algorithms can win with at most* 75%*.*

Exercise 23. *Show that if one player measures the EPR pair in* θ *basis and another in* ϕ *basis, then they output the same value w.p* $\cos^2(\theta - \phi)$*.*

Figure 1.16: Alice and Bob strategies

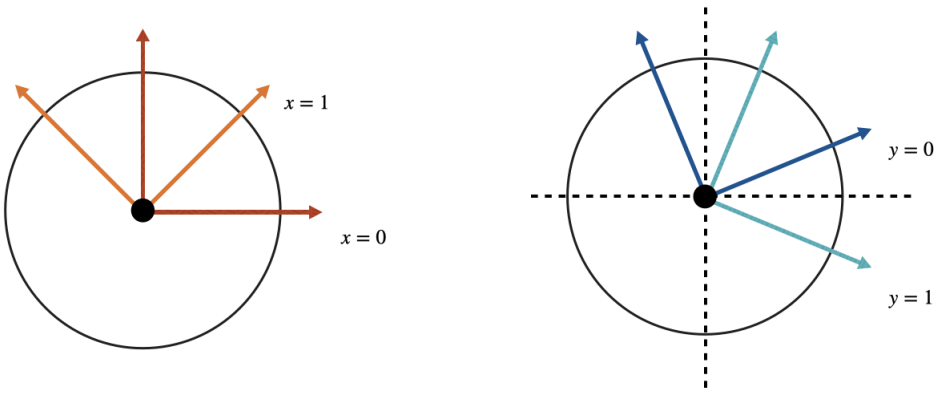
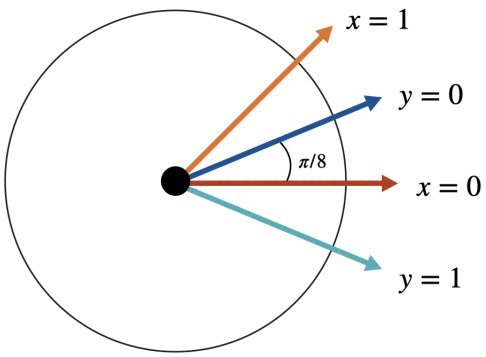


Figure 1.17: Analysis of the strategies



Turns out there is a quantum algorithm that wins with probability $\cos^2 \pi/8 \approx 85\%$. *Here is how. A and B follow the following strategy:*

- *A:*
  - *If receives* $x = 0$ *they measure in* $\theta = 0$ *basis*
  - *If receives* $x = 1$ *they measure in* $\theta = \pi/4$ *basis*
- *B*
  - *If receives* $y = 0$ *they measure in* $\theta = \pi/8$ *basis*
  - *If receives* $y = 1$ *they measure in* $\theta = -\pi/8$ *basis*

We now observe that if $x = y = 0$, $x = 0, y = 1$ *or* $x = 1, y = 0$ *then the angle between the basis of measurements for A and B is* $\pi/8$. *Therefore the probability that they measure the same bit is* $\cos^2(\pi/8)$. *If* $x = y = 1$, *on the other hand, the angle between the measurment bases is* $3\pi/8$ *therefore the probability that A and B measure different bits is* $\sin^2(3\pi/8) = \cos^2(\pi/8)$. *Therefore with probability that they win this game is* $\cos^2(\pi/8)$. *This is summarized below (see also fig. 1.16, fig. 1.17):*

- $\angle(x = 0, y = 0) = \pi/8$. *Output* $a = b$ *w.p.* $\cos^2 \pi/8$
- $\angle(x = 0, y = 1) = \pi/8$. *Output* $a = b$ *w.p.* $\cos^2 \pi/8$
- $\angle(x = 1, y = 0) = \pi/8$. *Output* $a = b$ *w.p.* $\cos^2 \pi/8$
- $\angle(x = 1, y = 1) = \pi/4 + \pi/8 = \pi/2 - \pi/8$. *Output* $a \neq b$ *wp* $\cos^2 \pi/8$

Exercise 24. *Can you win with higher probability by any quantum strategy?*

1.5.5 The GHZ state and its unusual properties

The GHZ state, named after "Daniel M. Greenberger, Michael A. Horne, and Anton Zeilinger", is the following 3-qubit entangled state defined as:

$$|GHZ\rangle_{ABC} = \frac{1}{\sqrt{2}}(|000\rangle_{ABC} - |111\rangle_{ABC})$$

It is also sometimes called the cat state (can you guess why?). Here we will show a nonclassical feature of this quantum state. Suppose three players A, B and C share a GHZ state (each holding a qubit). Each player either measures a Pauli X or a Pauli Y. We first note that if all three measure a Pauli X, then they collectively obtain the value -1. *This is because*

$$(X_A \otimes X_B \otimes X_C)\,|GHZ\rangle_{ABC} = -1\,|GHZ\rangle_{ABC}$$

Now suppose that two players measure Y and one player measures X. We can show that they will collectively measure $+1$. *This is because*

$$(X_A \otimes Y_B \otimes Y_C)\,|GHZ\rangle_{ABC} = +1\,|GHZ\rangle_{ABC}$$

$$(Y_A \otimes X_B \otimes Y_C)\,|GHZ\rangle_{ABC} = +1\,|GHZ\rangle_{ABC}$$

$$(Y_A \otimes Y_B \otimes X_C)\,|GHZ\rangle_{ABC} = +1\,|GHZ\rangle_{ABC}$$

This assignment of values to variables is not possible classically. To see this consider the classical variables $X_A, X_B, X_C, Y_A, Y_B, Y_C$ *each taking a value from* ± 1*. A classical analog of the GHZ measurements correspond to these classical variables satisfying*

$$X_A X_B X_C = -1$$

$$Y_A X_B X_C = +1$$

$$X_A Y_B X_C = +1$$

$$X_A X_B Y_C = +1$$

However, this is not possible classically. Do you see why? This indicates that quantum variables are not predetermined; they reveal their values only when a measurement is performed. This is another refutation of local hidden variables, based on quantum correlations that are fundamentally different from classical ones. The strength of the GHZ game over the CHSH game is that, unlike CHSH, which is a probabilistic win over a classical protocol, the GHZ game succeeds with probability 1 *in a task that cannot be accomplished classically.*

Chapter 2

Quantum Algorithms

2.1 Overview

So far we described the quantum formalism an saw a few important protocol in quantum computing such as Hadamard test and teleportation. We saw the role of entanglement in these protocol. We are now ready to get started with quantum algorithms. Quantum algorithms will be a large part of our focus during this semester. We first start with the quantum black box model, which is an idealized way of describing input to quantum computations. We will describe algorithms due to Deutsch-Josza, Bernstein-Vazirani and Simon. The problems these algorithms solve involve learning properties of Boolean functions. While these problems seem very abstract, they are the backbone of some of the algorithms we will describe later. Next, we will describe the celebrated Shor's algorithm for factoring large numbers. One of the main elements of this algorithm is the so-called quantum Fourier transform which we will describe in detail. Next we describe quantum phase estimation, Hamiltonian simulation and energy estimation. After that we will go over the Grover's search algorithm. We will then describe two special topics: quantum algorithms for linear systems which have applicaitons in machine learning and the hidden subgroup problem which generalizes the Shor's algorithm in several ways.

2.2 The quantum black-box model

How do we describe the input to a quantum computation? Suppose we have a Boolean function $f : \{0,1\}^n \to \{0,1\}$ *and would like to provide access to instances of this function to a quantum computer. For this purpose, we use the so-called black-box model. A black-box representation for a function is a box that takes a string of bits* $x \in \{0,1\}^n$ *as input and outputs a single bit equal to* $f(x)$*. The black-box model is also sometimes called the oracle model. We immediately face a problem. How do we query an oracle in a reversible way? the input to the back box is* n *bits and the output is a single bit. It turns out we can implement the oracle in two ways: the phase oracle and the index oracle. The phase query works according to* $O_f : |x\rangle \to (-1)^{f(x)}|x\rangle$*. How is this implementation reversible? If you query this oracle twice* $O_f(O_f(|x\rangle)) = (-1)^{f(x)}O_f(|x\rangle) = (-1)^{f(x)+f(x)}|x\rangle = |x\rangle$*. As a result the oracle is the inverse of itself and is hence reversible. The oracle is furthermore linear:* $O_f(|v\rangle + |w\rangle) = O_f(|v\rangle) + O_f(|w\rangle)$*. The second model we consider for quantum oracles is the index oracle. The oralce takes two set of registers as input. In the first set we encode the input and in the second set we input an arbitrary string that has the same size as the size of the output of the function. For the case of* f*, since* f *outputs one bit, the second register takes one bit. If the output of* f *was three bits the second set would take three bits. The way the oracle acts is similar to a controlled-NOT oparation.* O_f *takes* $|x, w\rangle$ *as*

input and produces $|x, w \oplus f(x)\rangle$. *Recall that for* $a, b \in \{0,1\}^m$, $c = a \oplus b$ *is the bit-wise XOR of the two bits, i.e.,* $c_i = a_i \oplus b_i$ *for* $1 \leq i \leq m$. *Why is this oracle reversible? If we query* O_f *twice we obtain* $O_f(O_f(|x\rangle\,|w\rangle)) = O_f(|x\rangle\,|w \oplus f(x)\rangle) = |x\rangle\,|w \oplus f(x) \oplus f(x)\rangle$. *Similar to the phase oracle, the index oracle is also linear.*

Exercise 25. *Prove that the above two notions are equivalent by allowing ancillas.*

2.2.1 The Deutsch-Josza Algorithm

A function is called constant if it outputs the same bit 0 *or* 1 *on every input. It is called balanced if the number of inputs that produce the* 0 *output is the same as the number of inputs that produce* 1. *For instance the NOT function is a balanced function. In the Deustch-Josza problem, we have black-box access to a function* $f : \{0,1\} \to \{0,1\}$, *and wish to see wether it is constant or balanced, ie is* $\alpha_f := f(0) \oplus f(1) = 0$ *or* 1. *Classically we need to make two queries. Why? There are four possible function* $: \{0,1\} \to \{0,1\}$*:* $f(x) = 0, f(x) = 1, f(x) = x$ *and* $f(x) = NOT(x)$. *Suppose we query the function on the* 1 *input and suppose we obtain the output* 0. *We know that the function cannot be the constant* $f(x) = 1$ *or the balanced* $f(x) = x$, *but we can't distinguish between* $f(x) = 0$ *and* $f(x) = NOT(x)$. *DJ showed that quantumly you can do this using* 1 *query: We first apply a Hadamard gate, then phase query the function, then apply the Hadamard gate again. We can show that if the function is balanced, then we will sample* 1 *from the output with probabibility* 1, *and otherwise* 0. *Here is the analysis.*

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{2.1}$$

$$\xrightarrow{O_f} \frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle) \tag{2.2}$$

$$\propto |(-1)^{\alpha_f}\rangle \tag{2.3}$$

$$\xrightarrow{H} |0\rangle \text{ if constant or } |1\rangle \text{ if balanced.} \tag{2.4}$$

• **Generalization to multi-qubit** *We can consider the generalization of the Deutsch-Josza (DJ) problem to functions taking many input bits. Similar to the* $\{0,1\} \to \{0,1\}$ *functions, we can define constant function to be functions that output the same value,* 0 *or* 1, *on every input. Similarly, we define the balanced function to be one for which, out of the* $N = 2^n$ *possible inputs,* $N/2$ *yield* 0 *and* $N/2$ *yield* 1 *(so they are called balanced). Consider the following problem.*

Problem 1 (Generalized Deutsch-Josza)**.** *Given a Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *and the promise that* f *is either constant of balanced decide which one is the case.*

Out of the 2^{2^n} *boolean functions taking* n*-bit string to one bit, there are* $\binom{2^n}{2^{n-1}} \sim 2^{2^n - n/2}$ *balanced functions. Why? Using counting argument (similar to what we did before), we can deduce that extereme majority of balanced functions require exponential-size circtuits. There are however only two constant functions* $f(x) = 0$ *and* $f(x) = 1$. *The generalized DJ problem is called a promise problem because we are "promised" that the black-box function is either constant or balanced. There are* $2^{2^n}(1 - \frac{1}{2^{n/2}})$ *functions that are neither constant of balanced and we are promised that those instances are given to us as input.*

Claim 1. *There is a quantum algorithm that decides balanced vs. constant using 1 single query.*

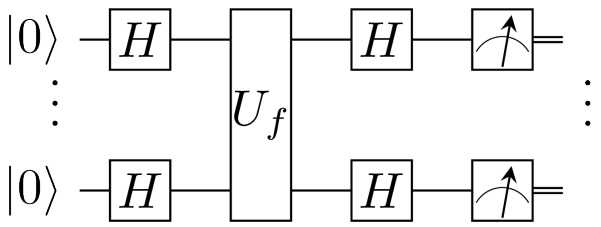


Figure 2.1: The circuit for multi-qubit Deutsch-Josza algorithm

Proof. We use the circuit in Figure 2.2.1.

$$|0^{\otimes n}\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \tag{2.5}$$

$$\xrightarrow{O_f} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle \tag{2.6}$$

$$\xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} H^{\otimes n} |x\rangle \tag{2.7}$$

To evaluate $H^{\otimes n}|x\rangle$ we use the following expression

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

Now examine the amplitude for $|0^{\otimes n}\rangle$ which appears with probability $\frac{1}{4^n}|\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|^2$. If f is constant this probability is 1, and if balanced it is 0. □

Remark 5. *Since there are doubly-exponentially different balanced functions one needs exponential queries to solve the generalized DJ problem classically. Another way to see this is by noticing that if we query f on any subset of inputs less than 2^{n-1} and we get the same value say 0, we cannot be still sure whether the function is constant or balanced. We hence get an exponential-to-one query improvement. However if we are allowed random queries, we can solve the problem classically also using constant (> 1) number of queries. See the exercise below.*

Exercise 26. *Show that there is a randomized classical query procedure which solves the DJ problem with high probability using constant number of queries.*

2.2.2 The Bernstein-Vazirani Algorithm

We saw that there is a quantum procedure that can solve the DJ problem using only a single query, while the best classical query procedure would require at last exponentially many queries in the worst case. We however saw in the last exercise that the same problem can be solved using a constant number of queries if we are allowed to use random queries. So in some ways we obtain a constant > 1 to 1 query improvement using quantum queries.

The Bernstein Vazirani problem is an instance where we get polynomial to constant improvement over a query problem. For $x, y \in \{0,1\}^n$, we use the notation $x \cdot y = x_1 \cdot y_1 \oplus \ldots \oplus x_n \cdot y_n$. The Bernstein-Vazirani Problem is the following problem

Problem 2 (Bernstein-Vazirani)**.** *Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$ with $f(x) = s\cdot x$ for a secret $s \in \{0,1\}^n$, find s.*

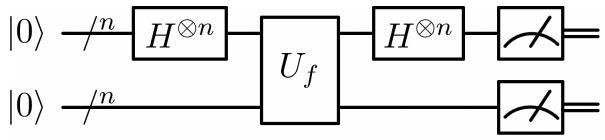


Figure 2.2: The circuit for Simon's algorithm (from Wikipedia)

We need $\Theta(n)$ classical queries. To see the $O(n)$ upper-bound note $s_i = f(0^{i-1}10^{n-i+1})$. For the lower-bound we use an information theoretic argument: if we make less than n queries there is always more than one candidates for s that are consistent with all the queries.

Claim 2. *There is a quantum query algorithm that achieves the goal with only 1 query.*

Proof. We use the circuit in Figure 2.2.1 again. Note the output of the circuit is

$$|\psi_{\text{out}}\rangle = \frac{1}{N} \sum_{x,y \in \{0,1\}^n} (-1)^{f(x)-x\cdot y} |y\rangle . \tag{2.8}$$

using $f(x) = s \cdot x$ we obtain

$$|\psi_{\text{out}}\rangle = \frac{1}{N} \sum_{x,y \in \{0,1\}^n} (-1)^{(s-y)\cdot x} |y\rangle = |s\rangle . \tag{2.9}$$

Here we used the observation that $\sum_{x \in \{0,1\}^n} e^{a\cdot x} = N$ if $a = 0$ and 0 otherwise. □

2.2.3 Simon's Algorithm

We wish a problem that we witness exponential speedup for a quantum algorithm. Simon's problem exactly achieves this:

Problem 3. *Given $f : \{0,1\}^n \to \{0,1\}^n$ and the promise that $f(x) = f(y)$ iff $x = y \oplus s$ for a secret key s. Find s.*

Example 5. $s = 110$

$$\begin{aligned} f(000) &= 1, \\ f(001) &= 2, \\ f(010) &= 3, \\ f(011) &= 4, \\ f(100) &= 3, \\ f(101) &= 4, \\ f(110) &= 1, \\ f(111) &= 2 \end{aligned}$$

Claim 3. *There is a quantum algorithm that achieves this using polynomially many queries.*

Proof. We use the query model to create $|0^n\rangle \otimes |0^n\rangle \to \frac{1}{\sqrt{N}} \sum_x |x\rangle |f(x)\rangle$ we then measure the f register to obtain $\frac{|x\rangle + |y\rangle}{\sqrt{2}}$ where $x \cdot s = y \cdot s$. We then apply Hadamard $H^{\otimes n}$ to the state to

obtain state $\propto (1 + (-1)^{(x-y)\cdot z}) |z\rangle \propto (1 + (-1)^{s\cdot z}) |z\rangle$ we measure to obtain z knowing that $z \cdot s = 0$. Taking many independent samples we end up with a system of equations:

$$\begin{aligned} z_1 \cdot s &= 0 \\ z_2 \cdot s &= 0 \\ &\vdots \\ z_n \cdot s &= 0 \end{aligned}$$

□

Exercise 27. *Show classically we need $\Theta(2^{n/2})$.*

2.3 Shor's problem

Simon's problem, while not directly addressing a real-world task, was an important milestone in quantum computing. It provided an oracle separation between classical and quantum computation, though it did not establish a true separation for naturally arising problems. The significance of Simon's work became clearer when Shor observed that, with additional effort, one could adapt the core ideas into a powerful algorithm for integer factoring on a quantum computer.

At the heart of this transformation lies the quantum Fourier transform (QFT). In fact, Fourier analysis underpins nearly all the quantum algorithms we have discussed. Simon's algorithm made use of the Fourier transform over the group $\mathbb{Z}_2^n$, while Shor extended this framework by applying Fourier transform techniques to the cyclic group of integers, $\mathbb{Z}_N$.

In Simon's problem, one is given a function f with the promise that there exists a secret string s such that $f(x + s) = f(x)$ for all x. The goal is to determine this hidden period s. Shor's algorithm, by contrast, tackles the period-finding problem over the integers: given a function $f : [N] \to [N]$ with the property that $f(x) = f(x + r) = f(x + 2r) = \cdots$ (with arguments taken modulo N), the task is to identify the hidden period r.

In what follows, we first provide an introduction to Fourier analysis and its quantum counterpart, the quantum Fourier transform. We then define the problem of integer factoring and explain how Shor's algorithm uses these tools to achieve an exponential speedup over classical methods.

2.3.1 Quantum Fourier Transform

Let $\omega = e^{2\pi i/N}$, be the N'th root of 1. Let $f : [N] \to \mathbb{C}$ be complex numbers. We define the Fourier transformation of f as

$$\hat{f}(k) = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f(j)\omega^{jk}$$

It is insightful to note ω satisfies the following identity

$$1 + \omega + \omega^2 + \ldots + \omega^{N-1} = 0.$$

To see this, let $S = 1 + \omega + \ldots + \omega^{N-1}$. Therefore $1 + \omega S = S + \omega^N$. Hence

$$S = \frac{1 - \omega^N}{1 - \omega} \tag{2.10}$$

since $\omega^N = 1$, therefore $S = 0$. Next consider the following sum for some integer l:

$$S_l := 1 + \omega^l + \omega^{2l} + \ldots + \omega^{(N-1)l}.$$

First we observe that if l is an integer multiple of N then $\omega^l = 1$, therefore $S_l = N$. Otherwise, using Equation 2.10 (replacing ω with ω^l):

$$S_l = \frac{\omega^{lN} - 1}{\omega - 1} = 0.$$

therefore

$$\frac{1}{N} S_l = \begin{cases} 1 & \text{if } l \text{ is an integer multiple of } N \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

One of the implications of the above Harmonic identity is that

$$f(j) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \hat{f}(k) \omega^{-jk}$$

To see this, we perform the following calculation

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \hat{f}(k) \omega^{-jk} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} (\frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} f(l) \omega^{lk}) \omega^{-jk} \tag{2.12}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} f(l) \omega^{(l-j)k} \tag{2.13}$$

$$= \sum_{l=0}^{N-1} f(l) \frac{1}{N} (\sum_{k=0}^{N-1} \omega^{(l-j)k}) \tag{2.14}$$

$$= \sum_{l=0}^{N-1} f(l) (\frac{1}{N} S_{l-j}) \tag{2.15}$$

$$= f(j) \tag{2.16}$$

To see the last line (Equation 2.16) we note that the only possibility for $l - j$ to be an integer multiple of N is that $l = j$. We now use Equation 2.11 to conclude that the only term that survives in the sum $\sum_{l=0}^{N-1} f(l)(\frac{1}{N} S_{l-j})$ is $f(j)$.

The Fourier transform maps the constant function to pulses and vice-versa. In particular, let δ be such that

$$\delta(j) = \begin{cases} 1 & j = 0 \\ 0 & j \neq 0 \end{cases}$$

and $c : [N] \to \mathbb{C}$ be such that $c(j) = 1/\sqrt{N}$ for all $j \in [N]$, then $\hat{c} = \delta$ and $\hat{\delta} = c$. In other words, if a function is very flat, then its Fourier tranform will be highly spiked (and vice versa).

The quantum Fourier transform: *Based on this background, we can now define the quantum Fourier transform. The Hilbert space is $\mathbb{C}^N$. QFT acts on this Hilbert space. In particular, it maps the basis according to*

$$QFT : |j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk} |k\rangle \tag{2.17}$$

$$QFT = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)^2} \end{pmatrix} \tag{2.18}$$

Exercise 28. *Show that this map is unitary.*

Let's solve some simple examples. Let's compute

$$QFT\,|0\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{0\times k}\,|k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle \tag{2.19}$$

We obtain a uniform superposition. Now imagine $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle$. *Then*

$$QFT\,|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} QFT\,|j\rangle \tag{2.20}$$

$$= \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \omega^{jk}\,|k\rangle \tag{2.21}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} (\sum_{j=0}^{N-1} \omega^{jk})\,|k\rangle \tag{2.22}$$

$$= |0\rangle\,. \tag{2.23}$$

Next, imagine we start with a superposition over even numbers. For simplicity let $N = 2L$

$$|\psi\rangle = \frac{1}{\sqrt{L}} \sum_{l=0}^{L-1} |2l\rangle \tag{2.24}$$

Now we apply the Quantum Fourier Transform

$$QFT\,|\psi\rangle = \frac{1}{\sqrt{L}} \sum_{l=0}^{L-1} QFT\,|2l\rangle \tag{2.25}$$

$$= \frac{1}{\sqrt{LN}} \sum_{l=0}^{L-1} \sum_{k=0}^{N-1} \omega^{2kl}\,|k\rangle \tag{2.26}$$

$$= \frac{1}{\sqrt{LN}} \sum_{k=0}^{N-1} (\sum_{l=0}^{L-1} \omega^{2kl})\,|k\rangle\,. \tag{2.27}$$

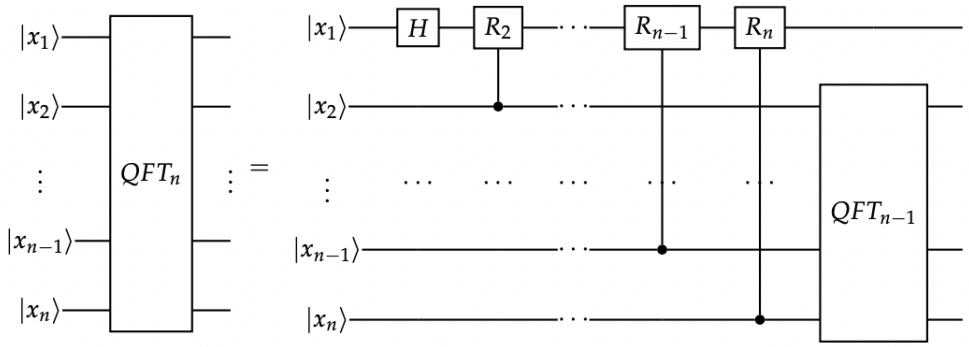
In the sum above only two terms survive $k = 0$ *and* $k = L = N/2$*; the reason is that at these two points* $\omega^k = 1$ *and the corresponding amplitude becomes* $\frac{1}{\sqrt{NL}} \sum_{l=0}^{L-1} 1 = \frac{1}{\sqrt{2}}$*; for other terms we get zero because* ω^{2k} *is a nontrivial* ($\neq 1$) *L'th root of identity. therefore the output becomes* $\frac{1}{\sqrt{2}}(|0\rangle + |N/2\rangle)$*. In general, using similar ideas, we can show for* $N = s \cdot L$*, if*

$$|\psi\rangle = \frac{1}{\sqrt{L}} \sum_{l=0}^{L-1} |s \cdot l\rangle$$

then

$$QFT\,|\psi\rangle = \frac{1}{\sqrt{s}} \sum_{j=0}^{s-1} |j \cdot N/s\rangle\,.$$

Figure 2.3: Recursive implementation of QFT



Implementation of the quantum Fourier transform: *Recall the binary notation for numbers; a number $j \in \mathbb{N}$ can be represented by $j = j_0 2^0 + j_1 2^1 + \ldots + j_{n-1} 2^{n-1}$, where $j_l \in \{0, 1\}$ and $j_{n-1} \ldots j_1 j_0$ is the binary representation. Similarly, $0.j_l \ldots j_m = \frac{j_l}{2} + \ldots + \frac{j_m}{2^{m-l+1}}$. We can show that the quantum Fourier transform maps:*

$$|j_1, \ldots, j_n\rangle \mapsto \frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i 0.j_n}|1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle) \ldots (|0\rangle + e^{2\pi i 0.j_1 \ldots j_{n-1}j_n}|1\rangle) \quad (2.28)$$

For proof, see Nielsen-Chuang chapter 5. We can implement this using a number of controlled rotations like

$$R_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi i \frac{2^k}{2^n}} \end{pmatrix} \quad (2.29)$$

Figure represents a recursive implementation of the quantum Fourier transform. QFT can be implemented using $O(n^2)$ elementary gates.

Exercise 29. *Show the above recursive implementation of the quantum Fourier transform works.*

2.3.2 The factoring problem

Here we formally define the factoring problem:

Problem 4 (Factoring)**.** *Given a composite number N output its prime factorization, i.e.,p, q s.t. $N = pq$, in time polynomial in $|N|$ number of digits in N in, say, binary representation.*

Mathematical Structure of Shor's Algorithm: *Shor's algorithm is fundamentally based on the number-theoretic structure that underlies the RSA cryptosystem. The key object of study is the multiplicative group of integers modulo N, denoted by*

$$\mathbb{Z}_N^\times := \{\, x \in \mathbb{Z} \mid \gcd(x, N) = 1 \,\}.$$

This set consists of all integers that are relatively prime to N, equipped with multiplication modulo N as the group operation. The size of this group is given by Euler's totient function $\phi(N)$. In particular,

$$|\mathbb{Z}_N^\times| = \begin{cases} N - 1, & \text{if } N \text{ is prime,} \\ (p-1)(q-1), & \text{if } N = pq \text{ is a product of two distinct primes.} \end{cases}$$

For any $x \in \mathbb{Z}_N^\times$, the order *of x, denoted by r, is defined as the smallest positive integer satisfying*

$$x^r \equiv 1 \pmod{N}.$$

This quantity plays a central role in Shor's algorithm. Defining the function

$$f_x(\ell) = x^\ell \bmod N,$$

we observe that f_x is periodic, and its period is precisely the order r of x in $\mathbb{Z}_N^\times$. In particular, the problem of determining this period is the following:

Problem 5 (Period finding)**.** *Find the order of x in $\mathbb{Z}_N$ in polynomial time $|N|$.*

The following result of Miller connects factoring with period finding

Theorem 2 (Miller '70)**.** *With constant probability a uniform random element x of $\mathbb{Z}_N$ has order $2r$ such that both $gcd(N, x^r + 1)$ and $gcd(N, x^r - 1)$ are non-trivial factors of N.*

Therefore, once the period r is known, it can be used to factor N efficiently on a classical computer. In the next section, we will show that we can solve it efficiently using the quantum Fourier transform.

Corollary 1. There is a reduction from factoring to Period finding.

We note the similarity between this problem and the Simon's problem: Here we have $f : \mathbb{Z}_N^+ \to \mathbb{Z}_N^+$, $f_x(l) = x^l \bmod N$, and the promise that $f_x(a) = f_x(b)$ iff $a - b$ is a period of f. The Simon's problem was about finding the period of a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$.

2.3.3 Shor's algorithm

Here we outline the steps of the Shor's algorithm. We only keep the high-level idea. We note that these steps are very similar to the steps in Simon's algorithm. We first query the function $f(l) = x^l \bmod N$ on all relevant inputs, then measure the funciton register to obtain a superposition over all pre-images of a specific value $f(r)$. We then use quantum Fourier transform (and extra postprocessing) to learn the period. The quantum Fourier transform is in place of the Hadamard transform in Simon's algorithm.

- *In the first step of the algorithm we find Q (a power of 2) that is essentially close to N^2. $\log Q$ is the number of qubits we use to store the input register. The reason for this choice is to have enough number of qubits to produce a superposition over all relevant input instances.*
- *Next we prepare*

$$\frac{1}{\sqrt{Q}} \sum_{r=1}^{Q-1} |r\rangle |f_x(r)\rangle = \frac{1}{\sqrt{Q}} \sum_{r=1}^{Q-1} |r\rangle |x^r \mod N\rangle \tag{2.30}$$

  Note $x^r \bmod N$ can be prepared using repeated squaring.

- *Next we measure the second register to obtain*

$$\frac{1}{\sqrt{l}} \sum_{i=1}^{l-1} |r_0 + is\rangle |f_x(r_0)\rangle \tag{2.31}$$

  where $l = \frac{Q - r_0 - 1}{s}$.

- *Apply QFT to the first register*

$$\frac{1}{\sqrt{lQ}} \sum_{i=0}^{l-1} \sum_{r=0}^{Q-1} \omega^{r(r_0+is)} |r\rangle |f_x(r_0)\rangle \tag{2.32}$$

QFT can be applied using a circuit of size $\log^2 N$ *composed of Hadamards and controlled phases.*

- *Now measure and obtain* $|r_1\rangle |f_x(r_0)\rangle$ *with probability*

$$\frac{1}{Ql} |\sum_{i=0}^{l-1} \omega^{r_1(r_0+is)}|^2 = \frac{1}{Ql} |\sum_{i=0}^{l-1} \omega^{ir_1 s}|^2 \tag{2.33}$$

If $r_1 s$ *is close to a multiple of* Q *then the above probability is close to* 1 *otherwise* 0*. Assuming we obtain one of these instances, once we measure the register we obtain* $r_1 s = mQ$ *for some integer* m*. We divide* r_1 *by* Q *and obtain* m/s *we can obtain* s *using a procedure known as the continued fraction procedure as we will outline below.*

Continued Fraction: *The continued fraction is a method which given a real number,* r*, computes an approximation of this number as a fraction of integers of the form*

$$r \approx a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ldots + \cfrac{1}{a_n}}}}} =: \frac{P_n(r)}{Q_n(r)}$$

I like this example in Vazirani's lecture notes. We try this approximation for the number π*. We can do it for the estimation* $\pi \approx 3.14$*.*

$$\pi \approx 3.14 \tag{2.34}$$
$$= 3 + \frac{1}{\frac{100}{14}} \tag{2.35}$$
$$= 3 + \frac{1}{7 + \frac{2}{14}} \tag{2.36}$$
$$\approx 22/7 \tag{2.37}$$

We can obtain other candidates by considering more decimals in π*.*

$$\pi \approx 3.1415 \tag{2.38}$$
$$= 3 + \frac{1}{\frac{10000}{1415}} \tag{2.39}$$
$$= 3 + \frac{1}{7 + \frac{95}{1415}} \tag{2.40}$$
$$= 3 + \frac{1}{7 + \frac{1}{14 + \frac{1}{\frac{85}{95}}}} \tag{2.41}$$
$$= 3 + \frac{1}{7 + \frac{1}{14}} \tag{2.42}$$
$$\approx 311/99 \tag{2.43}$$

Lemma 5. *If* r *is rational equal to* P/Q *then* $Q = Q_n(r)$ *for some* $n = O(\log Q)$*.*

Back to the Shor's algorithm. Recall that we have stored an estimation of m/s *in the output of our quantum algorithm. It turns out that if we obtain* m/s *with good enough accuracy then using the above lemma,* s *can be obtained using* $s' = Q_n(r)$ *for some* $n = O(\log N)$*. The point is that we can verify that* s' *is indeed the period; otherwise we try again.*

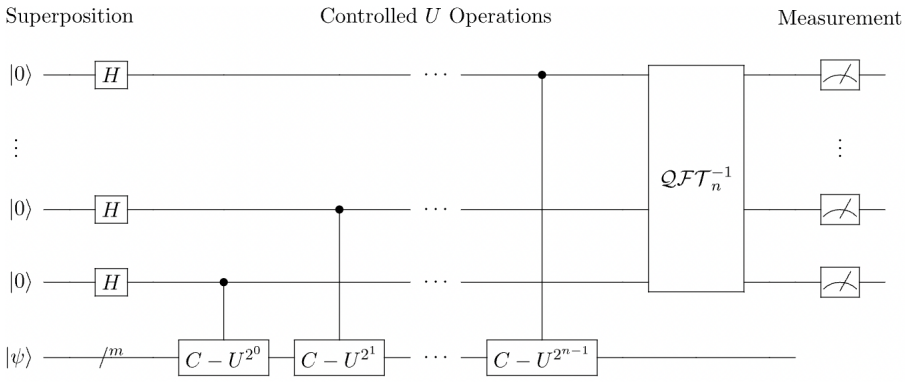


Figure 2.4: The circuit for phase estimation (from Wikipedia)

2.4 Quantum phase estimation

This section follows closely the narrative by Nielsen-Chuang. Another application of the quantum Fourier transform is phase estimation. The problem is as follows:

Problem 6 (Phase estimation)**.** *Given black-box access to controlled-U for a unitary U and one of the eigenvectors of U, output an estimation to the corresponding eigenvalue.*

Let $|u\rangle$ be the corresponding eigenvector. The algorithm is as follows: We start with $|0^t\rangle \otimes |u\rangle$ (t is large enough to store the final value for the eigenvalue with appropriate precision). We number the registers containing zeros by 1 to t. We apply Hadamards to the zeros (i.e., qubits 1 to t) to prepare a uniform superposition. We then apply controlled-U^{2^j}, controlled on the j'th qubit, to $|u\rangle$. After this operation, we will obtain

$$\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i 0.\phi_t}|1\rangle)(|0\rangle + e^{2\pi i 0.\phi_{t-1}\phi_t}|1\rangle)\dots(|0\rangle + e^{2\pi i 0.\phi_1\dots\phi_{t-1}\phi_t}|1\rangle)|u\rangle \tag{2.44}$$

We then apply the inverse Fourier transform to the first t qubits and read an estimation of ϕ.

Solving period finding via phase estimation: *We consider U to be the following unitary $U : |y\rangle \mapsto |xy \mod N\rangle$. Here in order for U to be unitary, x needs to be relatively prime with respect to N. Like before, let r be the period of x in N, i.e., $x^r = 1 \mod N$. We can show that*

$$|u_s\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp(\frac{-2\pi isk}{r})|x^k \mod N\rangle$$

is an eigenvector of U with eigenvalue

$$e^{\frac{2\pi is}{r}}$$

the main challenge is to implement $|u_s\rangle$. Well, if we don't know r, we can do that, but that is the same as solving the original period-finding question. The main observation is that

$$|1\rangle = \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle \tag{2.45}$$

Therefore, the output state is within close distance to $\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle \otimes |s/r\rangle$. So for each s we obtain an estimate of s/r with probability $\sim 1/r$ within $O(\log N)$ bits of accuracy. We can use the continued fraction algorithm to find the closest rational number to s/r and obtain s' and r' such that $s/r = s'/r'$. Assuming s and r are relatively prime, we obtain $r = r'$; we can test if r' is the real r and if it was not we can try again.

2.5 Energy estimation

We saw that using quantum phase estimation we can estimate eigenvalues of a unitary matrix U, given access to powers of controlled-U operations and the specific eigenvectors. Can we use this algorithm to estimate the eigenvalues of physical observables? Recall that a physical observable O is a Hermitian operator whose eigenvalues model the attainable physical measurements out of a physical observable. In particular if O has eigenvalues and eigenvectors $\lambda_i, |i\rangle$, respectively, then if we measure O in the eigenstate $|\psi_j\rangle$ we obtain $\langle j|O|j\rangle = \lambda_j$. One of the most important observables in physics is the Hamiltonian or energy observable. A Hamiltonian is an observable whose eigenvalues are a system's energy levels. It is important to note that the Hamiltonian also describes the time evolution of a closed physical system. It turns out that if a system is described according to a Hamiltonian H then after t steps it has evolved according to the unitary matrix

$$e^{-iHt}$$

Recall that if A is a Hermitian matrix, then e^{iA} is unitary. To get a better sense of why that happens, we note the well-known Schrödinger equation describing the time evolution of a system is according to the differential equation

$$i\partial_t |\psi(t)\rangle = H |\psi(t)\rangle$$

If the system at time zero starts with $|\psi(0)\rangle$ then the solution to the Schrödigner equation at time t is according to $|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle$.

Suppose we are given one of the eigenstates of a system with Hamiltonian H. How do we measure the energy of the system at that level? If we can prepare the unitary e^{-iH} and controlled operations based on it, then we can input the eigenstate to the phase estimation circuit to estimate and measure the energy. The reason this works is that if E_j is an eigenvalue of H with eigenvector $|j\rangle$ then e^{-iE_j} is an eigenvalue of e^{-iHt}. The reason is due to the spectral decomposition. Recall the spectral decomposition of $H = \sum_j E_j |j\rangle \langle j|$. We discussed that for any function f of H we get $f(H) = \sum_j f(E_j) |j\rangle \langle j|$.

It remains to explain how we can implement $U = e^{-iHt}$ using basic quantum gates. This is the topic of the quantum simulation algorithm, which we will discuss next. Let me remark that assuming we can implement U using basic two-qubit gates, we can also implement controlled-U^j for $j \geq 1$. To get from controlled-U to controlled-U^j we just repeat the former j times. Suppose $U = g_T \dots g_1$ where each g_i is a two-qubit gate. We notice that $c-U = \prod_i c-g_i$. To see this, we observe that for any operation A, $c-A = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes A$. Each gates $c-g_i$ is a quantum gate acting on constant (3) qubits and can be implemented using constant many elementary gates; the reason is that any quantum circuit acting on l qubits can be implemented using $2^{O(l)}$ elementary gates.

I would like to remark that if, instead of one of the eigenstates, we feed the phase estimation algorithm with an arbitrary quantum state $|\psi\rangle$, we obtain the following. We find the decomposition of $\psi = \sum_j \psi_j |j\rangle$ according to the orthonormal eigenbasis of H; recall that because H is Hermitian, its eigenstates can be made orthonormal. The quantum phase estimation algorithm maps $|0\rangle |\psi\rangle \mapsto\approx \sum_j \lambda_j |0\rangle |\tilde{E}_j\rangle$, where $\tilde{E}_j$ is an estimation of the energy E_j. If we sample from this distribution we will obtain $\tilde{E}_j$ with probability $|\lambda_j|^2$. By taking many samples and obtaining an average over them we get an estimation of the value $\sum_j |\psi_j|^2 \tilde{E}_j \approx \langle\psi|H|\psi\rangle$. In order to make the error less that ϵ we need to make around $\Omega(\frac{\|H\|_\infty^2}{\epsilon^2})$ measurements.

2.6 Quantum simulation algorithm

In the previous section, we discussed the energy observable known as Hamiltonians, which capture the energy levels of the system and also describe the time evolution of a closed system. Let H be the Hamiltonian of a system. We explained that after time t the evolution of a system is described according to the unitary matrix $U = e^{-iHt}$. We say a Hamiltonian is k-local if it can be written as $H = \sum_i H_i$, where each H_i acts upon k qubits. Quantum simulation, which is Feynman's original idea of simulating quantum physics using quantum computers, can be captured according to:

Problem 7 (Quantum simulation)**.** *Given access to terms H_j of a k-local Hamiltonian $H = \sum_{j=1}^{m} H_j$, with $\max_j \|H_j\|_\infty = h$ prepare an efficient unitary circuit that estimates the unitary e^{-iH}. Suppose further more for any terms H_j there are at most K terms that don't commute with it.*

The main idea is based on a formula due to Trotter, also known as the Lie product formula. In particular, for two matrices A and B, the Trotter formula is

$$e^{A+B} = \lim_{N\to\infty} (e^{A/N} e^{B/N})^N \tag{2.46}$$

In what follows, we prove this formula, analyze its rate of convergence, and explain how we can use it to simulate quantum Hamiltonians. If A and B are two matrices that commute, then $e^{A+B} = e^A e^B$. So if the terms of the Hamiltonian H commute with each other, then $e^{-iH} = \prod_j e^{-iH_j}$. Since each e^{-iH_j} acts at most on k qubits, then we can prepare it using a quantum circuit of size at most $2^{O(k)}$. So we can implement this unitary using $m \cdot 2^{O(k)}$ steps.

What happens when the terms of the Hamiltonian do not commute with each other? For matrices A and B, if they don't commute, then $e^{A+B} \neq e^A e^B$. In particular

$$e^{A+B} - e^A e^B = \sum_{k=0}^{\infty} \frac{(A+B)^k}{k!} - \sum_{k=0}^{\infty} \frac{A^k}{k!} \sum_{k=0}^{\infty} \frac{B^k}{k!} \tag{2.47}$$

$$= \frac{(A+B)^2}{2} - \frac{A^2}{2} - \frac{B^2}{2} - AB + h.o. \tag{2.48}$$

$$= -\frac{1}{2}[A,B] + h.o. \tag{2.49}$$

In general, we can show

$$e^{\sum_i A_i} - \prod_i e^{A_i} = -\frac{1}{2} \sum_{i<j} [A_i, A_j] + h.o. \tag{2.50}$$

The main idea is to simulate approximately $e^{-iH/N}$ and repeat N times. This way the commutation between two terms in the Hamiltonian decays as $\|[\frac{H_i}{N}, \frac{H_j}{N}]\|_\infty \sim \frac{h^2}{N^2}$. For an appropriately chosen N we show that this simulation gives a good estimation. In particular,

$$\|e^{-iH/N} - \prod_j e^{-iH_j/N}\|_\infty = O(\frac{Kmh^2}{N^2}) \tag{2.51}$$

Next, we have to repeat the above procedure for N times. There is a well-known observation in quantum information that the error in quantum circuits grows linearly.

Lemma 6. *Let $A = \prod_{j=1}^{N} A_j$ and $B = \prod_{j=1}^{N} B_j$ be unitary matrices such that $\|A_i - B_i\|_\infty \leq \epsilon$ for all $1 \leq i \leq N$ then $\|A - B\|_\infty \leq N\epsilon$.*

Exercise 30. *Prove this lemma.*

Applying this lemma to the Equation 2.52 we obtain

$$\|e^{-iH} - (\prod_j e^{-iH_j/N})^N\|_\infty = O(\frac{Kmh^2}{N}) \tag{2.52}$$

Error analysis: *If we choose $N = \Omega(\frac{Kmh^2}{\epsilon})$ we can estimate e^{-iH} within error ϵ. We can implement $(\prod_j e^{-iH_j/N})^N$ in time $T = mN2^{O(k)}$. As a result, we can simulate e^{-iH} within error ϵ in time $T = \frac{Km^2h^22^k}{\epsilon}$.*

Remark 6. *In order to simulate e^{-iHt} we need to replace H_j with H_jt and hence in the error analysis we need to replace h with ht. Assuming $k, h = O(1)$ we conclude that the Hamiltonian system after t time steps can be simulated in time $T = O(\frac{Km^2t^2}{\epsilon})$. Our intuition suggests that we should be able to simulate the system in a time linear in t. Using more careful analysis of recent work has been able to achieve the optimal bound $O(ts + \log\frac{1}{\epsilon})$, where s is the sparsity, ie, the maximum number of nonzero terms in each row of the Hamiltonian in the computational basis. See e.g. Berry-Childs-Kothari 2015 and Low-Chuang 2016.*

Remark 7. *In order to use the quantum phase estimation algorithm for the energy estimation problem we need to implement $e^{-i2^k|1\rangle\langle 1|\otimes H}$ for different values of k. (Remember that quantum phase estimation involved implementation of the 2^k'th power of controlled unitary for k up to the number of bits of precision we demand). However if we want polynomial bits of precision in energy (i.e. estimate energy within inverse exponential error), then we need to perform Hamiltonian simulation for $|1\rangle\langle 1| \otimes H$ for an exponentially long time. However the quantum simulation algorithms we provided will run in exponential time. This is not a coincidence because estimating the ground state energy of local Hamitlonian systems within exponential precision is hard for complexity class* PSPACE*. In practice, we can hope to estimate energy only up to polylogarithmic bits of precision (i.e. within inverse polynomial error). It happens that this problem is complete for the class* QMA *which is the quantum analog of complexity class* NP*.*

2.7 Grover's Search Algorithm

One of the most celebrated results in quantum computing is Grover's algorithm, which provides a quadratic speedup for the unstructured search problem. While the speedup is smaller than the exponential gain offered by Shor's algorithm for factoring, it is optimal for problems where no structure is assumed in the search space.

2.7.1 The Search Problem

The search problem can be phrased as follows:

Problem 8 (Search)**.** *Given oracle access to a Boolean function*

$$f : [N] \to \{0, 1\},$$

such that there exists at least one marked element x^* satisfying $f(x^*) = 1$, the goal is to find such x^*.

Remark 8. *If the function f is completely unstructured, any classical algorithm must evaluate f on at least $N-1$ inputs in the worst case (and $N/2$ on average). Thus, no classical approach can achieve sublinear query complexity without additional structure. The quantum search (Grover) algorithm which we will present shortly will accomplish this task using only $O(\sqrt{N})$ queries.*

2.7.2 Quantum Oracle Model

In the quantum setting, the function f is given as a phase oracle acting on computational basis states:

$$O_f : |x\rangle \mapsto (-1)^{f(x)} |x\rangle .$$

Equivalently, for a superposition $\sum_x \alpha_x |x\rangle$, the oracle acts as

$$O_f \left(\sum_x \alpha_x |x\rangle \right) = \sum_x \alpha_x (-1)^{f(x)} |x\rangle .$$

Hence, the oracle flips the phase of the marked basis states, leaving all others unchanged.

2.7.3 High-Level Intuition

The main idea behind Grover's algorithm is to amplify *the amplitude of the marked element(s) while reducing the amplitude of the unmarked ones. Starting from an equal superposition over all basis states, we repeatedly apply two reflections:*

- ***Oracle Query:*** *Apply the oracle O_f to flip the phase of the marked states:*

$$O_f |s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{f(x)} |x\rangle .$$

  Oracle query can be viewed as a reflection about the hyperplane orthogonal to the marked state $|x^*\rangle$ (implemented by O_f).

- ***Diffusion (Reflection) Operator:*** *A reflection about the mean amplitude (implemented by the* diffusion operator *D). Define*

$$D = 2 |s\rangle \langle s| - I = H^{\otimes n} (2 |0\rangle \langle 0| - I) H^{\otimes n}.$$

  This operator reflects the amplitude vector about the mean. For a general state $|\alpha\rangle = \sum_x \alpha_x |x\rangle$, let $S = \frac{1}{N} \sum_x \alpha_x$ be the average amplitude. Then D maps $\alpha_x \mapsto \alpha'_x = 2S - \alpha_x$.

The composition of these two reflections results in a rotation within a two-dimensional subspace spanned by:

$$|w\rangle = |x^*\rangle , \qquad |r\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq x^*} |x\rangle .$$

After a sufficient number of iterations, the state vector becomes close to $|w\rangle$, so measuring in the computational basis reveals the marked element with high probability.

Grover's Algorithm. *Let $N = 2^n$. The algorithm proceeds as follows:*

1. ***Initialization:*** *Prepare the uniform superposition*

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

   This can be obtained by applying $H^{\otimes n}$ to $|0\rangle^{\otimes n}$.

Figure 2.5: The circuit for Grover's algorithm

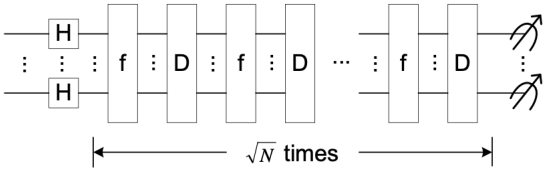
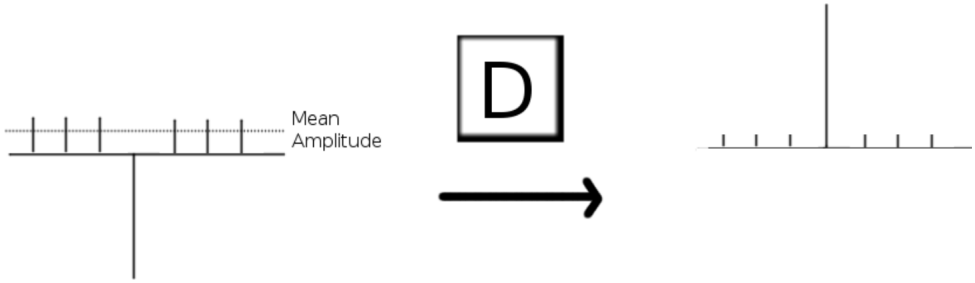


Figure 2.6: The effect of the reflection operator Grover's algorithm



2. ***Grover Iteration:*** *Apply the composite operator*

$$G := DO_f$$

We will argue that this operation will perform a rotation in the two-dimensional subspace $\text{span}\{|w\rangle, |r\rangle\}$ *by an angle* 2θ*, where* $\sin\theta = \frac{1}{\sqrt{N}}$.

3. ***Repetition:*** *Repeat the Grover iteration* G *approximately* $\frac{\pi}{4}\sqrt{N}$ *times. At this point, the amplitude of* $|w\rangle$ *is maximized.*

4. ***Measurement:*** *Measure the final state in the computational basis. With probability close to* 1*, the outcome will be the marked element* x^*.

The circuit for Grover's search is summarized in Fig. 2.7.3

2.7.4 Interpretation based on diffusion of amplitudes

We can show that D *maps the quantum state* $\sum_x \alpha_x |x\rangle$ *to* $\sum_x \alpha'_x |x\rangle$ *where* $\alpha'_x = 2S - \alpha_x$*, where* $S := \frac{\sum_i \alpha_i}{N}$.

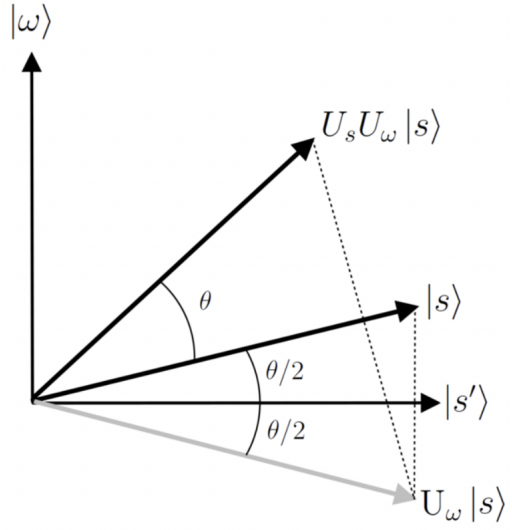
Exercise 31. *Prove this.*

The effect of this map is damping the points that are above S *and amplify the points that are below* S*, which in effect mimics a diffusion process. Another way to interpret this is that the map reflects points around* S*. See Fig. 2.7.4. As a result the algorithm at every step first flips the amplitude of the marked item (which will then be lower than any other amplitude on the diagram) then it applies the diffusion operator which amplifies the amplitude of the marked item.*

2.7.5 Geometric Interpretation

Initially, the state $|s\rangle$ *makes an angle* θ *with the marked state* $|w\rangle$*, where* $\sin\theta = 1/\sqrt{N}$*. Each Grover iteration applies a rotation by* 2θ *toward* $|w\rangle$*. After about* $\frac{\pi}{4\theta} \approx \frac{\pi}{4}\sqrt{N}$ *iterations, the*

Figure 2.7: Analysis of Grover (from Wikipedia)



state is almost parallel to $|w\rangle$, leading to successful measurement with high probability. Using this insight we give a rigorous proof of the performance of Grover's algorithm below:

Theorem 3 (Analysis of Grover)**.** *Grover's algorithm succeeds after time $O(\sqrt{N})$.*

Proof. Let x_0 be the solution to our search. We look at the space spanned by $|\omega\rangle := |x_0\rangle$ and $|s'\rangle := \frac{1}{\sqrt{N-1}} \sum_{x \neq x_0} |x\rangle$. Let $|s\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$. See Figure 2.7. The effect of the oracle query is given by $U_\omega := (I - 2 |\omega\rangle \langle\omega|)$; geometrically the effect of this operator is tantamount to reflecting the sign for the component of the state along the direction $|\omega\rangle$. Similarly the effect of the diffusion (or reflection) operator is given by $U_s = (2 |s\rangle \langle s| - I)$, geometrically equivalent to reflecting along $|s\rangle$. Initially, after applying the Hadamard operation, the state of the quantum computer is in $|A_0\rangle := |s\rangle = \frac{1}{\sqrt{N}} |\omega\rangle + \sqrt{\frac{N-1}{N}} |s'\rangle$. In each round we first query then function then the diffusion. Let $G = U_s U_\omega$. The state of our quantum computer after t iterations of G is $|A_t\rangle := G^t |A_0\rangle$.

Initially, the overlap between $|A_0\rangle$ and $|\omega\rangle$ is $\langle A_0 | \omega \rangle = \frac{1}{\sqrt{N}}$. Geometrically, this is the same as saying that the angle between this state with the $|s'\rangle$ is $\sin \Delta = \sqrt{\frac{1}{N}}$. In each iteration the angle will increase by 2Δ. Hence after $\sim \sqrt{N}$ iterations we obtain an angle around $\pi/2$ and hence we are at state $|\omega\rangle$. It is important to note that if we keep repeating this algorithm we will deviate from the correct answer.

□

2.7.6 Optimality and Generalization

Grover's algorithm achieves a query complexity of $O(\sqrt{N})$, which has been proven optimal in the black-box model by the Bennett–Bernstein–Brassard–Vazirani (BBBV) bound. Any quantum algorithm for unstructured search must make at least $\Omega(\sqrt{N})$ oracle calls. Moreover, Grover's algorithm can be generalized to multiple marked items.

Exercise 32. *Show that by generalizing the reflection (to reflect the state from the hyperplane of the span of marked items), the above Algorithm can find one out of k marked items in time $O(\sqrt{N/k})$.*

2.8 Solving systems of linear equations (Optional)

One of the most important computational problems throughout sciences and engineering is solving linear systems. The problem is, given a square matrix and a target vector $\vec{b}$, find vector $\vec{x}$ such that $A\vec{x} = \vec{b}$. We consider the following versions of this problem for which we can gain exponential quantum speedup.

Problem 9 (Linear systems)**.** *Given a $N \times N$ Hermitian matrix A, and a unit vector $|b\rangle$, and another Hermitian matrix M, estimate $\langle x| M |x\rangle$, where $|x\rangle$ is such that $A|x\rangle = |b\rangle$.*

- *In 2008 Harrow Hassidim Lloyd proposed a quantum algorithm that runs in $\log N \cdot O(\kappa^2)$.*
- *Best classical algorithm runs in time $O(N\kappa)$ (or $O(N\sqrt{\kappa})$ for PSD matrices).*

The Algorithm: *Let $\lambda_i, |u_i\rangle$ be the eigenvalue and eigenvectors of A. Consider the decomposition of $|b\rangle = \sum_{i=1}^{N} \beta_i |u_i\rangle$ in the eigenbasis of A.*

1. *Use quantum simulation to prepare e^{iAt}.*
2. *Use quantum phase estimation mapping : $|b\rangle \otimes |0\rangle \mapsto\approx \sum_{j=1}^{N} \beta_j |u_j\rangle \otimes |\lambda_j\rangle$.*
3. *Add an additional ancilla and rotate it conditioned on the value of the $|\lambda\rangle$ register and obtain*
$$\approx \sum_{j=1}^{N} \beta_j |u_j\rangle \otimes |\lambda_j\rangle \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle + \frac{C}{\lambda_j} |1\rangle\right)$$
4. *Uncompute the $|\lambda\rangle$ register and obtain*
$$\approx \sum_{j=1}^{N} \beta_j |u_j\rangle \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle + \frac{C}{\lambda_j} |1\rangle\right)$$
5. *Measure the ancilla register and condition on obtaining a 1 to obtain the state*
$$\approx \sqrt{\frac{1}{\sum_{j=1}^{N} C^2 |\beta_j|^2 / |\lambda_j|^2}} \sum_{j=1}^{N} \beta_j \frac{C}{\lambda_j} |u_j\rangle$$
6. *Up to normalization we obtain some state $\approx \sum_{j=1}^{N} \beta_j/\lambda_j |u_j\rangle = A^{-1}|b\rangle = |x\rangle$. The normalization is the probability of obtaining 1.*
7. *Measure POVM $\{M, I - M\}$ to obtain $\langle x|M|x\rangle$*

Analysis:

- *Assuming A is $s-$sparse, we can perform quantum simulation for e^{iAt} in time $O(\log N)s^2t$.*
- *If A is not Hermitian, then define*
$$\Lambda = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}$$
and solve $\Lambda |y\rangle = \begin{pmatrix} |b\rangle \\ 0 \end{pmatrix}$, where $|y\rangle = \begin{pmatrix} 0 \\ |x\rangle \end{pmatrix}$.

- *If* $f(i,j) = \sum_{k=i}^{j} |\langle k|b\rangle|^2$ *is efficiently computable, then we can prepare* $|b\rangle$ *efficiently; otherwise we can assume* $|b\rangle$ *is given to us as a subroutine in some other algorithm.*
- *in order to succeed we need to choose* $C = O(1/\kappa)$ *and succeed with probability* $\Omega(1/\kappa^2)$.

2.9 The hidden subgroup problem (Optional)

The most immediate generalization of Shor's problem which involves the (Abelian) cyclic group is to extend it to families of problems known as the "Hidden-Subgroup Problems" over arbitrary groups. Shor and Kitaev showed that we get quantum polynomial-time algorithms for Abelian groups. We don't know if the same is possible for arbitrary groups.

Problem 10 (The Hidden Subgroup Problem)**.** *Let* $(G,\cdot)$ *be a group and* $H \leq G$ *be a subgroup. Suppose* $f : G \to [N]$ *is such that for* $x, y \in G$, $f(x) = f(y)$ *iff* $x = hy$ *for some* $h \in H$. *(In other words of* x, y *belong to a left coset of* H.*) Find the generators of* H.

- *The Simon's problem is a special case:* $G = (\mathbb{Z}_2^N, +)$ *and* $H = \{0, s\}$.
- *Shor's problem is also a special case:* $G = (\mathbb{Z}_N, +)$ *and* $H = \{\ldots, -2s, -s, 0, s, 2s, \ldots\}$ *and* $f_x(l) = x^l \mod N$.

Theorem 4 (Shor-Kitaev)**.** *The Hidden subgroup problem over finite Abelian groups can be solved within* BQP.

- *We know if SAT is reducible to HSP then the polynomial Hierarchy collapses.*
- *HSP is within* NP ∩ coAM

Theorem 5. *Graph Isomorphism* $\leq$ *HSP*

Proof. For a graph C let Automorphism group of C be the set of permutations $Aut(C) := \{\pi \in S_n : \pi(C) \cong C\}$. Given two graphs C_1, C_2 let $C = C_1 \cup C_2$. Now if $C_1 \ncong C_2$ then $Aut(C) = Aut(C_1) \times Aut(C_2)$. Here is the reduction $G = S_n$, $H = Aut(C)$, $f_C(\pi) = \pi(C)$. □

2.9.1 Query complexity of HSP (optional)

Theorem 6 (Ettinger-Höyer-Knill)**.** *HSP can be solved using polynomial many queries to* $f : G \to \mathbb{N}$, *where* f *satisfies* $\forall x, y \in G, f(x) = f(y)$ *iff* $x = hy$ *for some* $h \in H \trianglelefteq G$.

Proof sketch. Suppose we have

1. Prepare the superposition: $\frac{1}{\sqrt{|G|}} \sum_{x\in G} |x\rangle$
2. Query f to prepare: $\frac{1}{\sqrt{|G|}} \sum_{x\in G} |x\rangle |f(x)\rangle$
3. Measure the second register to get a superposition: $|C\rangle = \frac{1}{\sqrt{|H|}} \sum_{h\in H} |hy\rangle$
4. Repeat this for $K \approx \log^2 |G|$ times to get: $|C_1\rangle, \ldots, |C_K\rangle$.

We claim is that there is a measurement to give us H. The measurement is not necessarily efficient. We observe there are at most $|G|^{\log|G|}$ different subgroups of G. The reason is that there are at most $\log|G|$ generators for G, and there are, therefore at most $\binom{|G|}{\log|G|}$ many distinct subgroups.

1. Prepare $|\psi_H\rangle = |C_1\rangle \otimes \ldots \otimes |C_K\rangle$.

We observe that if $H \neq H' \trianglelefteq G$ then $|H \cap H'| \leq |H|/2$. To see this, we observe that since $H \neq H'$ there exists a nontrivial $x \in H/H'$. Therefore for any $y \in H \cap H', xy \in H/H'$. Therefore $|H \cap H'| \leq |H/H'|$ and since $|H| = |H \cap H'| + |H/H'|$, $|H \cap H'| \leq |H|/2$. Therefore $|\langle H, H'\rangle| \leq \frac{1}{2}$. This because if $|H\rangle = \frac{1}{\sqrt{|H|}} \sum_{h\in H} |hy\rangle$ and $|H'\rangle = \frac{1}{\sqrt{|H'|}} \sum_{h'\in H'} |h'y'\rangle$ then $\langle H, H'\rangle = \frac{|H\cap H'|}{\sqrt{|H||H'|}}$. That means if two subgroups are all the same, they are exactly the same. Therefore $|\langle \psi_H, \psi_{H'}\rangle| \leq \frac{1}{2^K}$. There are at most $|G|^{\log|G|}$ different such ψ_H's, therefore, it is enough to take $K \approx \log^2|G|$ to reduce the error below the required amount.

□

Exercise 33. *How can we perform the measurement above?*

Chapter 3

Quantum error correcting codes

These notes are closely aligned with the topics presented in Nielsen Chuang and the quantum computer science course by David Mermin. For more details please read relevant chapters from those references.

3.1 Classical repetition code

Suppose we want to encode a bit 0 *or* 1 *assuming a model of noise where each bit has a probability* p *of being flipped. If we encode information using naive encoding, then with probability* p*, information gets corrupted; this error channel is known as the binary symmetric channel. Such a way of encoding information is not reliable. The repetition code instead repeats each bit at the encoding. For instance, suppose we encode* 0 *with* 000 *and* 1 *with* 111*. We furthermore decode information by taking a majority vote. In this way, we can correct one bit getting flipped. However, if two or more bits are flipped, our decoding procedure does not give the correct answer. The probability of two or more bits getting flipped is* $3p^2(1-p) + p^3 = O(p^2)$*. As a result, we suppress the probability of error by a quadratic factor. More generally, if we use the encoding* $0 \to 0^k$ *and* $1 \to 1^k$ *and decode using majority vote, then assuming* $p < 1/2$*, the probability of error decays exponentially fast in* k*.*

Exercise 34. *Prove this.*

3.2 Correcting quantum bit flips

Initially, it may appear that error correction is impossible when comparing classical and quantum information processing due to significant differences between the two frameworks. For instance, due to the no-cloning theorem, we may not copy information. Also, quantum measurements destroy quantum information. Furthermore, the space of errors is continuous and corresponds to a much larger space than the classical domain. For instance, a quantum state may experience Z *error,* X *error, or an error as a linear combination of the two. It is an outstanding discovery that besides all these limitations, quantum error correction is still possible.*

Suppose, for now, our noise model is that on each qubit, we have equal probability p *of getting an unwanted bit flip* X*. Let's choose an encoding:*

$$0 \to |\bar{0}\rangle := |000\rangle\,, \quad 1 \to |\bar{1}\rangle := |111\rangle\,.$$

we can perform this encoding using a CNOT between first and second and a CNOT between first and third qubits. The circuit implementing this operation is givne in Figure 3.1. A nice observa-

tion is that if we input an arbitrary quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$*, we obtain* $|\bar{\psi}\rangle = \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$*. The main observation is that any quantum state like* $|\bar{\psi}\rangle$ *belongs to a two-dimensional subspace spanned by* $|000\rangle$ *and* $|111\rangle$*. Let's call this subspace,* V_0*, the error-free subspace. Suppose we apply a bit flip to the first qubit of* $|\bar{\psi}\rangle$ *and obtain* $\alpha|100\rangle + \beta|011\rangle$*. Any such quantum state belongs to the subspace of quantum states spanned by* $|100\rangle$ *and* $|011\rangle$*. Let's call this subspace* V_1*. Similarly, if we apply a bit flip operation to the second or third qubit, we obtain a quantum state in* $V_2 = span\{|010\rangle, |101\rangle\}$ *and* $V_3 = span\{|001\rangle, |110\rangle\}$*. It is crucial to observe that subspaces* V_0, V_1, V_2, V_3 *are mutually orthogonal. To correct the incident error, we need to measure the subspace* V_i *(*$i = 0, 1, 2, 3$*) where the quantum state belongs to. To do this, we use the following POVM*

$$\Pi_0 = |000\rangle\langle 000| + |111\rangle\langle 111| \tag{3.1}$$
$$\Pi_1 = |100\rangle\langle 100| + |011\rangle\langle 011| \tag{3.2}$$
$$\Pi_2 = |010\rangle\langle 010| + |101\rangle\langle 101| \tag{3.3}$$
$$\Pi_2 = |001\rangle\langle 001| + |110\rangle\langle 110|. \tag{3.4}$$

To see why this set of operators corresponds to a POVM, we note that $\Pi_i \geq 0$*, and* $\Pi_0 + \Pi_1 + \Pi_2 + \Pi_3 = I$*. Once we perform this POVM measurement and obtain a label* i*, we can correct the error. If* $i = 0$*, we don't need to do anything. But if* $i \in \{1, 2, 3\}$*, then we can correct the quantum state by applying* X_i *(*X *on qubit* i*).*

Another way to understand the error measurement is through the framework of observable measurements. Consider two observables $O_1 = Z_1Z_2$ *and* $O_2 = Z_2Z_3$*. If* $|w\rangle \in V_0$ *then* $O_1|w\rangle = |w\rangle$ *and* $O_2|w\rangle = |w\rangle$*, if* $|w\rangle \in V_1$ *then* $O_1|w\rangle = -|w\rangle$ *and* $O_2|w\rangle = |w\rangle$*, if* $|w\rangle \in V_2$ *then* $O_1|w\rangle = -|w\rangle$ *and* $O_2|w\rangle = -|w\rangle$ *and if* $|w\rangle \in V_3$ *then* $O_1|w\rangle = |w\rangle$ *and* $O_2|w\rangle = -|w\rangle$*. In other words, if we measure* O_1, O_2 *and obtain* $x, y \in \pm$*, then* $+,+$ *corresponds to no error,* $-,+$ *corresponds to bit flip on the first qubit,* $-,-$ *corresponds to an error on the second qubit and* $+,-$ *corresponds to an error on the third qubit. We can correct each error correspondingly. This step is known as syndrome measurement.*

To implement this measurement using the circuit model, we can use two extra ancillary qubits, both initialized at 0*. We perform CNOT between the first qubit and the first ancilla qubit and another CNOT from the second qubit to the first ancillary qubits. This way, we store the parity between the first two qubits in the first ancillary qubit. Similarly, we apply CNOT from the second qubit onto the second ancillary qubit and another CNOT from the third qubit onto the second ancillary qubit. As a result, we obtain the parity between the second and third qubits in the second ancillary qubits. We then measure the ancillary qubits. If we obtain* 00 *we apply nothing* I*. If we obtain* 10 *we apply* X_1*. If we obtain* 11 *we apply* X_2 *and if we obtain* 01*, we apply* X_3*. We can implement this step using the SELECT operation we discussed before (using Toffoli gates). See Figure 3.2 for the implementation.*

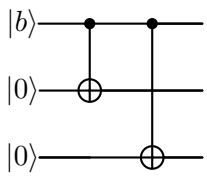


Figure 3.1: The encoding map which maps $|b\rangle$ to $|\bar{b}\rangle = |bbb\rangle$ for $b \in \{0, 1\}$. We note that the same map can be used for decoding, i.e., it transforms the quantum state $\alpha|000\rangle + \beta|111\rangle$ to $(\alpha|0\rangle + \beta|1\rangle)|00\rangle$.

Exercise 35. *Analyze an explain what happens to this error correcting code if two errors occur.*

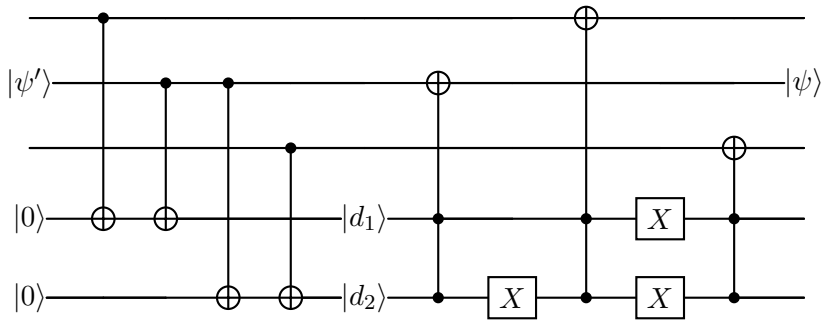


Figure 3.2: The error correcting map. The input is the quantum state $|\psi'\rangle$ which is equal to $|\psi\rangle$ after going through the noise channel. Assuming at most one bit flip has been applied we can correct this error. If $d_1 = d_2 = 0$ then no correction is needed. If $d_1 = 1, d_2 = 0$ we correct by applying an X operator on the first qubit of $|\psi\rangle$. If $d_1 = 1, d_2 = 1$ we correct by applying an X operator on the second qubit of $|\psi\rangle$. If $d_1 = 0, d_2 = 0$ we correct by applying an X operator on the third qubit of $|\psi\rangle$.

3.3 Correcting quantum phase flips

In the previous section we showed how we can correct one phase error. What happens if we consider phase flip errors i.e. receiving an unwanted Z error on one of the qubits. As we have seen before the Z operator can be obtained from X by the Hadamard change of basis: $Z = HXH$. Using this observation we consider the following encoding

$$0 \to |\bar{0}\rangle := |+++\rangle\,, \quad 1 \to |\bar{1}\rangle := |---\rangle\,.$$

we can implement this map using the circuit in Figure 3.5

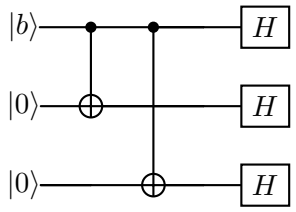


Figure 3.3: The encoding map which maps $|0\rangle$ to $|\bar{0}\rangle = |+++\rangle$ and $|1\rangle$ to $|\bar{1}\rangle = |---\rangle$.

The decoding map is according to the inverse of this map (and tossing out the two right-most ancillary qubits). The following figure captures this map:

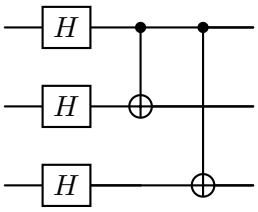


Figure 3.4: The dencoding map which maps $|0\rangle$ to $|\bar{0}\rangle = |+++\rangle$ and $|1\rangle$ to $|\bar{1}\rangle = |---\rangle$.

We show that we can correct up to one Z error using this encoding. To see this we observe that the dencoding from previous seciton works exactly the same way if we replace 0 with $+$ and 1 with $-$. For instance, if we start with a quantum state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$,

after the encoding step we get $|\bar{\psi}\rangle = \alpha|+++\rangle + \beta|---\rangle$. *If we apply a* Z *gate to the first qubit we get* $Z_1|\bar{\psi}\rangle = \alpha|-++\rangle + \beta|+--\rangle$. *So, the subspace corresponding to no error is* $V_0^Z = Span\{|+++\rangle, |---\rangle\}$, *the subspace corresponding to* Z_1 *error is* $V_1^Z = Span\{|-++\rangle, |+--\rangle\}$, *the subspace corresponding to* Z_2 *error is* $V_2^Z = Span\{|+-+\rangle, |-+-\rangle\}$, *the subspace corresponding to* Z_3 *error is* $V_3^Z = Span\{|++-\rangle, |--+\rangle\}$. *Similar to before, we can detect this error using the following POVM:*

$$\Pi_0^Z = |++\rangle\langle++| + |--\rangle\langle--| \tag{3.5}$$

$$\Pi_1^Z = |-+\rangle\langle-+| + |+-\rangle\langle+-| \tag{3.6}$$

$$\Pi_2^Z = |+-\rangle\langle+-| + |-+\rangle\langle-+| \tag{3.7}$$

$$\Pi_2^Z = |++\rangle\langle++| + |--\rangle\langle--|. \tag{3.8}$$

Also similar to before, to detect this error we can measure the operators $O_1^Z = X_1X_2$ *and* $O_2^Z = X_2X_3$. *We can see that the subspaces* V_i^Z *correspond to specific eigenspaces of* O_1^Z *and* O_2^Z. *For* V_0^Z *we obtain* $+,+$ *for the eigenvalues of* O_1^Z *and* O_2^Z, *respectively. For* V_1^Z *we obtain* $-,+$, *for* V_2^Z *we obtain* $+,+$, *and for* V_3^Z *we obtain* $+,-$. *The error correcting map is similar to Figure 3.2 except that we have to apply Hadamard gates in the begining and in the end to all qubits of* $|\psi'\rangle$.

3.4 Shor's 9-qubit code

We already saw how to correct errors in two (incompatible) basis. How can we design a quantum code that corrects both X *and* Z *errors at the same time? The Shor's 9-qubit code achieves this objective. As expected, the code is a combination of blocks involving phase flip correcting code and bit flip correcting code. The code is obtained by first mapping* $|0\rangle \to |+++\rangle$ *and* $|1\rangle \to |---\rangle$ *and then mapping each of the three qubits according to the bit flip encoding map and obtain*

$$|0_L\rangle = (\frac{|000\rangle + |111\rangle}{\sqrt{2}})(\frac{|000\rangle + |111\rangle}{\sqrt{2}})(\frac{|000\rangle + |111\rangle}{\sqrt{2}})$$

and

$$|1_L\rangle = (\frac{|000\rangle - |111\rangle}{\sqrt{2}})(\frac{|000\rangle - |111\rangle}{\sqrt{2}})(\frac{|000\rangle - |111\rangle}{\sqrt{2}})$$

The main observation is that if we apply one X *or* Z *gate we move to mutually orthogonal subspaces. This allows us to detect and correct error.*

To describe the error detection and correction step we use the syndrome measurement language. Now suppose a bit flip occurs on the first qubit. We can detect this error by measuring the syndrome observables Z_1Z_2 *and* Z_2Z_3 *and verifying that we get* -1 *on the first observable and* $+1$ *on the second. Similarly to detect bit flip error on the second block by measuring* Z_4Z_5, Z_5Z_6, *and on the third block by measuring* Z_7Z_8, Z_8Z_9. *We can correct this error by applying* X *operator to the faulty bit.*

Now we analyze what happens if a phase flip error Z *gets applied to one of the qubits. We claim by measuring the syndromes* $X_1X_2X_3X_4X_5X_6$ *and* $X_4X_5X_6X_7X_8X_9$ *we can detect and correct phase flip error.*

Exercise 36. *Explain how and why this syndrome measurement works. Give a procedure to correct this indecent error.*

Exercise 37. *Suppose a* Y *operation is applied to the first qubit. Analyze the Shor's code and explain how we can detect and correct this error. (Hint:* $Y = iXZ$.*)*

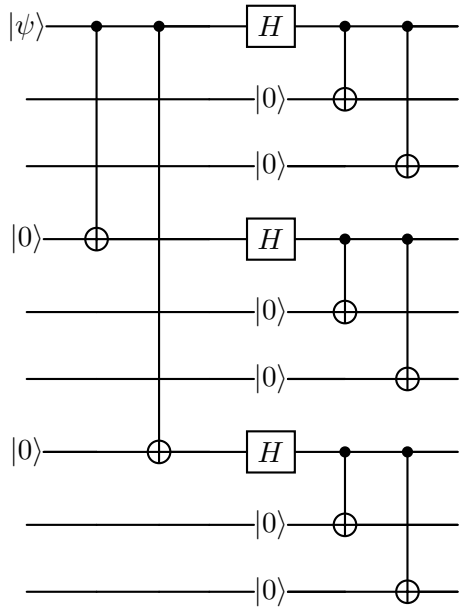


Figure 3.5: The encoding map for the Shor's code.

Exercise 38. *Suppose we apply a* $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ *to the first qubit. Show that Shor's code is capable of correcting this error.*

It turns out that any general quantum operation that involves affecting only one qubit can be corrected using Shor's code. For instance, if we apply a unitary operation $U = \sum_i \alpha_i A_i$ *where* A_i *is a single qubit operation, we can correct this error using Shor's code. The code can correct more general forms of (non-unitary) errors, which is beyond the scope of this course.*

3.5 Stabilizer codes

In this section, we introduce "stabilizer codes" as an important framework for error correction that formally generalizes the three codes discussed so far. Recall the insights we obtained so far. The code space corresponds to a subspace in the Hilbert space. Once an error occurs, the quantum state in the error-free subspace gets mapped to an error subspace, which is orthogonal to the error-free subspace, and hence, they can be perfectly distinguished from each other. Furthermore, we would like each Z *or* X *error on each qubit to map the subspace to subspaces that are mutually orthogonal to each other. Let's do a simple evaluation of how many qubits one needs to correct arbitrary single qubit gates. We saw that this is possible using the 9-qubit Shor's code. Can we do better? There are* $3n$ *single qubit operators (*X, Y, Z *on each qubit), so we need* $3n + 1$ *two-dimensional subspaces. Hence,* $2^n \geq 2(3n + 1)$*. We can see that to satisfy this criterion, we need* $n \geq 5$*. We will indeed give a five-qubit error-correcting code.*

Before we get there, let's make a few simple observations about the bit flip code. Recall that the syndrome observables for this code are Z_1Z_2 *and* Z_2Z_3*. We observe the following features: (1) these syndromes are tensor products of Pauli operators; hence they have eigenvalues* ± 1*, (2) they commute with each other (hence allow mutual eigenbasis), (3) they stabilize the code space (i.e., any quantum state of the form* $\alpha |000\rangle + \beta |111\rangle$*, (4) at least one of the syndrome operators anti-commutes with each of the bit flip errors which they can correct (they, however, commute with the* Z *errors and they are not able to correct these errors).*

Let's study the syndromes of Shor's 9-qubit code $Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9$, for bit flip errors and $X_1X_2X_3X_4X_5X_6, X_4X_5X_6X_7X_8X_9$ for phase flip errors. We note that these syndrome operators satisfy all the mentioned criteria. This inspires us to define a more general family of codes, namely the stabilizer codes that generalize all the codes discussed so far.

3.5.1 The stabilizer formalism

We define the Pauli group $\mathcal{P}_n$ as the group of Pauli strings of size n with phases $\pm, \pm i$. For instance, $iX \otimes Y \in \mathcal{P}_2$ and $-X \otimes X \otimes Z \in \mathcal{P}_3$[1]*. All elements in a Pauli group either commute or anti-commute with each other. A quantum state is a stabilizer state if a subgroup of Pauli exists that stabilizes it. In other words,*

Definition 3. *A quantum state $|\psi\rangle$ is a stabilizer state if there exists a subgroup $G \trianglelefteq \mathcal{P}_n$ s.t. $\forall g \in G, g|\psi\rangle = |\psi\rangle$. More generally, a subspace of the Hilbert space is a stabilizer subspace if a subgroup of Pauli stabilizes it.*

To see why the set of vectors $V \subseteq \mathcal{H}$ stabilized by a subgroup of Pauli constitute a linear subspace, note that if $|\psi\rangle, |\phi\rangle \in \mathcal{H}$, for any $\alpha, \beta \in \mathbb{C}$, $\alpha|\psi\rangle + \beta|\phi\rangle \in V$. Note that if the subgroup contains $-I, -iI, iI$, elements that have eigenvalues other than ± 1 or elements that anti-commute with each other, then the stabilizer state is trivially the 0 state. Why? We saw before that the elements of the Pauli group either commute or anti-commute with each other. Therefore, nontrivial stabilizer subgroups are commutative. Conversely, we can define a stabilizer group for a linear subspace $V \subseteq \mathcal{H}$.

Definition 4. *The stabilizer group G_V corresponding to the linear subspace of the Hilbert space $V \subseteq \mathbb{C}^n$, is defined as the largest subgroup $G \trianglelefteq \mathcal{P}_n$ that stabilizes V, i.e., for all $|v\rangle \in V$, $g \in G_V$, $g|v\rangle = |v\rangle$.*

To see why the set of elements that stabilize a subspace correspond to a group, we note that if two elements g and h stabilize a subspace, so does their multiplication. Furthermore, if g stabilizes V, so does g^{-1}, and clearly, the identity element stabilizes any element. Any subspace of the Hilbert space has a stabilizer subgroup because I by itself is a subgroup of Pauli.

How large is the stabilizer subspace for a given subgroup of Pauli? Suppose a G_V subgroup of $\mathcal{P}_n$ has k generators[2] *$g_1, \ldots, g_k$ and stabilizes the subspace $V \subseteq (\mathbb{C})^{\otimes n}$. To get a nontrivial subspace, furthermore assume that all elements of G_V commute with each other and furthermore $g^2 = I$, for all $g \in G_V$; furthermore, except for the identity element, all elements of G_V have zero traces. Our first observation is that the projector onto V is given by*

$$\Pi_V = \frac{1}{2^k} \sum_{g \in G_V} g$$

To see this, for each $g \in G_V$, since $g^2 = I$ then $\frac{I+g}{2}$ is the projector onto the $+1$ eigenspace of g. (Similarly, $\frac{I-g}{2}$ corresponds to the -1 subspace.) Since all elements in G_V commute, $\Pi_V = \prod_{g \in G}(\frac{I+g}{2})$. We know that $\dim(V) = Tr(\Pi_V)$. Therefore,

Lemma 7. *If the stabilizer subgroup of V has k generators, then $\dim(V) = 2^{n-k}$.*

[1] Recall that a group is a collection of objects with a multiplication rule, which is (1) closed under multiplication and is associative, (2) has an identity element, and (3) has an inverse element

[2] By $\langle g_1, \ldots, g_k \rangle$, we mean the set of elements generated by compositions of $g_1, \ldots, g_k$; a generator is the smallest set of group elements that generates that group

Intuitively, what this lemma is saying is that each generator of G_V divides the 2^n-dimensional Hilbert space $\mathbb{C}^{\otimes n}$ into two halves, hence the $+1$ subspace of k generators has dimension 2^{n-k}.

Let's work out a few examples. For the Hilbert space of one qubits, $\langle Z\rangle$ is a stabilizer subgroup with $k = 1$ generator. The dimension of the subspace stabilized by this group is $2^{1-1} = 1$ dimensional. We can see that this subspace is exactly the set of vectors spanned by $|0\rangle$. Similarly $\langle -Z\rangle$ stabilizes the subspaces $|1\rangle$. Now let us look at the quantum state $|0\dots 0\rangle$ (n zeros). What is the stabilizer group corresponding to this state? Clearly, the subspace is one-dimensional, so $2^{n-k} = 1$ only when $k = n$. Hence, we need to find n generators. It is easy to see that $\langle Z_1, \dots, Z_n\rangle$ is the stabilizer group. Let us consider the subgroup $\langle XX\rangle$. The stabilizer subspace corresponding to this group has $2^{n-k} = 2$ elements. One of the elements will be $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ and the other $\frac{|01\rangle+|10\rangle}{\sqrt{2}}$. Let us consider the subgroup $\langle XZ\rangle$. Like the previous example, its stabilizer subspace is a 2-dimensional subspace. This subgroup stabilizes $\frac{|00\rangle+|10\rangle}{\sqrt{2}}$ and $\frac{|01\rangle-|11\rangle}{\sqrt{2}}$. Next, consider the stabilizer subgroup $\langle XX, YY\rangle$. XX and YY commute with each other, and we have a 1-dimensional stabilizer subspace which is spanned by $\frac{|01\rangle+|10\rangle}{\sqrt{2}}$. Finally, consider the example $\langle Z_1Z_2, Z_2Z_3\rangle$, which corresponds to the syndrome operators for the bit flip error. We expect a $2^{3-2} = 2$-dimensional subspace, which not surprisingly happens to be the code subspace for the bit flip error, i.e., the set of vectors spanned by $|000\rangle, |111\rangle$.

Lastly, we present a useful lemma in the stabilizer formalism:

Lemma 8. *Let G_V be the stabilizer subgroup corresponding to a linear subspace V, and let U be any unitary operator, then $G_{UV} = UG_VU^{-1}$.*

Here $UV = \{U|v\rangle : |v\rangle \in V\}$, and $UGU^{-1} = \{UgU^{-1} : g \in G\}$. We leave the proof as an exercise. For instance, $\langle Z_1, Z_2\rangle$ is the stabilizer subspace of $|00\rangle$. Now let $U = CNOT_{12}H_1$. We know that $U|00\rangle = \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$. We also know that the stabilizer subgroup for $\frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$ is $\langle XX, ZZ\rangle$.

Exercise 39. *Verify that $UZ_1U^2 = X_1X_2$, $UZ_2U^{-1} = Z_1Z_2$.*

3.5.2 Stabilizer formalism for error correction

Let us get back to the bit flip code. As portrayed in the previous section, the code (no errors) subspace corresponds to the stabilizer subspace for the group $G_0 = \langle Z_1Z_2, Z_2Z_3\rangle$. Now consider the error X_1 being applied to a quantum state $|\psi\rangle$ being initially stabilized by G_0. We will obtain $|\psi'\rangle = X_1|\psi\rangle$. Using Lemma 8 is now stabilized by $X_1Z_1Z_2X_1 = -Z_1Z_2$ and $X_1Z_2Z_3X_1 = Z_2Z_3$. That is why the syndrome Z_1Z_2 detects a -1 and Z_2Z_3 keeps detecting $+1$. More generally, the syndrome g detects $+1$ if the incident error commutes with g and -1 if it anti-commutes. We can deduce the pattern of $+1, -1$ in syndrome measurements for other bit-flip errors using this window of reasoning. Furthermore, we can understand why these syndromes cannot detect Z errors. That is because Z errors commute with the syndromes. Moreover, we can understand why these syndromes fail to detect X_1X_2 errors correctly. That is because, for instance, this error term commutes with Z_1Z_2 and anti-commutes with Z_2Z_3, so it incorrectly detects X_1 error. Let's look at the syndromes of Shor's code. Recall the syndromes are $Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9$, for bit-flip errors and $X_1X_2X_3X_4X_5X_6, X_4X_5X_6X_7X_8X_9$. As an exercise, show that if a single qubit X or Z error occurs, we measure -1 when the error anti-commutes with a syndrome and $+1$ when it commutes.

Suppose we measured the syndrome, and we are sure that not more than one error has been applied. How should we decide what correction circuit we should apply? We need to find the

set of Pauli operators that anti-commute with the syndromes that are measured to be -1 and commute with those operators we measured to be $+1$. In the case of the example above, X_1 is specifically the operator that anti-commutes with Z_1Z_2 and commutes with Z_2Z_3.

How do we find the correction circuit systematically? We will do this next. But before that, let's define some notation. Let $a \in \mathbb{F}^n$ and $X^a = X_1^{a_1} \dots X_n^{a_n}$ (similarly for Z). Let's use the notation $P_{a,b} = i^{-a\cdot b} X^a Z^b$ to capture arbitrary Pauli strings, where $a \cdot b = a_1b_1 + \dots + a_nb_n$ is the usual inner product. We chose $c = i^{-a\cdot b}$ as the overall phase so that $P_{a,b}^2 = I$[3]. How do we capture Y using this notation? We leave it as an exercise. We can show that

$$P_{a,b}P_{a',b'} = (-1)^{a\cdot b' + a'\cdot b} P_{a',b'}P_{a,b}$$

Let

$$\Lambda := \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}.$$

then $a \cdot b' + a' \cdot b = (a,b)\Lambda \begin{pmatrix} a' \\ b' \end{pmatrix}$. This is called the symplectic inner product. As a result $P_{(a,b)}$ commutes with $P_{(a',b')}$ iff (a,b) is orthogonal to (a',b') according to the Symplectic inner product. The question of finding a Pauli string that commutes with a given set of syndromes and anti-commutes with others can be, therefore, captured according to a linear algebra problem over $\mathbb{F}_2^n$, $Aa = s$ where s is the vector of syndromes (0 for $+1$ and 1 for -1).

3.5.3 The five qubit code

As promised, in this section, we describe an error-correcting code encoding one logical qubit and correcting single-qubit errors with five qubits. To encode a two-dimensional error-free subspace, we need to provide $k = 4$ syndrome measurements; to see this, recall 2^{n-k} is the dimension of the stabilized subspace, so $k = n - 1 = 4$ gives us a two-dimensional subspace). Consider the syndrome measurements:

$$G = \langle XZZXI, IXZZX, XIXZZ, ZXIXZ \rangle$$

Fortunately, you have all the tools to analyze this code. We will now analyze the properties of this code.

Exercise 40. *Address the following items for the 5-qubit code*

1. *Show that all syndrome elements commute.*
2. *Write an expression for the projector Π_G onto the code subspace.*
3. *Suppose there is a bit-flip error on one of the qubits; find the pattern of $\pm$ in syndrome measurements. Repeat the same with phase-flip errors.*
4. *Define the logical qubits as $|\bar{0}\rangle := 4\Pi_G |00000\rangle$ and $|\bar{1}\rangle = 4\Pi_G |11111\rangle$ show that $\bar{X} := XXXXX$ performs logical X and $\bar{Z} := ZZZZZ$ performs logical Z on this basis.*

We now give a solution to this exercise: (1) We first show that all syndrome elements commute. To see this we note that all Pauli generators consist of only X and Z operations and X and Z operations overlap in even number of positions, hence they commute. To see this better, we use the symplectic inner product. Let M_j be the j'th Pauli generator and let $M_j = X^{a_j}Z^{b_j}$ for $a_j, b_j \in \mathbb{F}^5$. We can show that for any $1 \le j,k \le 4$, $a_j \cdot b_k + a_k \cdot b_j = 0 \mod 2$. For

[3] We equate $c^2 X^a Z^b X^a Z^b = c^2(-1)^{a\cdot b} = I$, So $c = i^{-a\cdot b}$ works

instance in $M_1 = XZZXI = X^{10010}Z^{01100}$ *corresponding to* $a_1 = 10010, b_1 = 01100$ *and* $M_2 = IXZZX = X^{01001}Z^{00110}$*, corresponding to* $a_2 = 01001, b_2 = 00110$*. We can verify* $a_1 \cdot b_2 + a_2 \cdot b_1 = 0 \mod 2$*.*

(2) The projector onto the code subspace is overage over the stabilizer group elements. In particular:

$$\Pi_G = \frac{I + XZZXI}{2}\frac{I + IXZZX}{2}\frac{I + XIXZZ}{2}\frac{I + ZXIXZ}{2}$$

(3) We first do this for the X_1 *error. We obtain a* $+$ *sign when the error commutes with a given syndrome and a* $-$ *sign if it anti-commutes. Hence we obtain* $+$ *for* $XZZXI, IXZZX, XIXZZ$*, and* $-$ *for* $ZXIXZ$*. Similarly for* Z_1 *we obtain* $-$ *for* $XZZXI, XIXZZ$*, and* $+$ *for* $IXZZX, ZXIXZ$*. The patterns can be obtained similarly for the other errors.*

(4) It is enough to observe that $\bar{X}\Pi_G = \Pi_G\bar{X}$ *and* $\bar{X}\Pi_G = \Pi_G\bar{X}$*. To see this, we can verify that* $\bar{X}$ *and* $\bar{Z}$ *commute with all elements in* G *(why?). This implies that:*

$$\bar{X}\left|\bar{0}\right\rangle = 4\bar{X}\Pi_G\left|00000\right\rangle = 4\Pi_G\bar{X}\left|00000\right\rangle = 4\Pi_G\left|11111\right\rangle$$

3.6 The Gottesman-Knill theorem

An important result in the theory of quantum computing is the Gottesman-Knill theorem, which introduces an important family of quantum computations, namely Clifford circuits, that generate large entanglement but can be simulated on a classical computer. We included this result in this handbook because it heavily builds on the stabilizer formalism we introduced in the previous sections.

Recall the Clifford gateset from previous lectures consisting of

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \quad \textit{(Hadamard)}$$

$$S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix} \quad \textit{(Phase)}$$

$$CNOT = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix} \quad \textit{(CNOT)}$$

While Clifford is not known to be universal If we add another gate $T = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}$*, also known as the* $\pi/4$ *phase shift, to the Clifford gates, we obtain a universal gate set. What about the Clifford gateset? Can we perform universal quantum computation based on the Clifford gateset alone? The Gottesman-Knill theorem gives strong evidence that the answer should be "no", by giving a classical algorithm that simulates this gateset.*

Theorem 7 (The Gottesman-Knill theorem)**.** *There is a classical algorithm that takes the description of a quantum circuit from the Clifford gateset* C *and samples from the output of* $C|0\ldots0\rangle$ *in polynomial time.*

To set up the ground, let's study some basic features of the Clifford gateset. Among the operations this gateset generates include the Pauli strings. To see this note $S^2 = Z$*,* $HZH = X$ *and*

$HXH = Z$. As a matter of fact, if we conjugate any Pauli string with any of the Clifford gates, we obtain other Pauli strings:

$$(CNOT)X_1(CNOT) = X_1X_2, \quad (CNOT)X_2(CNOT) = X_2, \tag{3.9}$$

$$(CNOT)Z_1(CNOT) = Z_1, \quad (CNOT)Z_2(CNOT) = Z_1Z_2, \tag{3.10}$$

$$SXS^{-1} = Y, \quad SZS^{-1} = Z \tag{3.11}$$

By induction, if we conjugate any Pauli string with any Clifford circuit C we obtain another Pauli string. Now recall the stabilizer formalism. Suppose we start with a subspace of quantum states V that are stabilized by a given set of Pauli stabilizer generations $\langle g_1, \ldots, g_k\rangle$. Let C be a Clifford circuit and let $CV = \{C|\psi\rangle : |\psi\rangle \in V\}$. Then CV is also a stabilizer subspace and is stabilized by $\langle Cg_1C^\dagger, \ldots, Cg_kC^\dagger\rangle$. This is the main idea behind the Gottesman-Knill algorithm. In particular, consider the quantum state $|0\ldots0\rangle$. We discussed that this is a 1-dimensional subspace stabilized by $\langle Z_1, \ldots, Z_n\rangle$. In order to store the description of the quantum state $C|0\ldots0\rangle$ we store $\langle CZ_1C^\dagger, \ldots, CZ_nC^\dagger\rangle$. Let C be a polynomial-size circuit in $CNOT, S$, and H. We can use Equations 3.9, 3.10, and 3.11 to update the set of generators at each step. Using the representation $P_{a,b} = i^{-a\cdot b}X^aZ^b$, for $a, b \in \mathbb{F}^n$, we can capture the Clifford operations using basic linear operations on over $\mathbb{F}^n$. For instance $(CNOT)X_1^{a_1}X_2^{a_2}(CNOT) = X_1^{a_1}X_2^{a_1\oplus a_2}$.

Exercise 41. *Show that if C is a Clifford operation then $CP_{(a,b)}C^\dagger = (-1)^{g(a,b)}P_{f(a,b)}$ for suitable functions $f : \mathbb{F}^{2n} \to \mathbb{F}^{2n}$, $g : \mathbb{F}^{2n} \to \mathbb{F}$. Describe f and g for basic Clifford gates $X, Y, Z, CNOT, H, S$.*

It remains to describe the measurement process. Suppose the set of generators before the measurement is $G = \langle g_1, \ldots, g_n\rangle$, and we want to measure a specific Pauli element g. Let $|\psi\rangle$ be the state of the quantum computer. There are two cases.

(1) If g commutes with all elements g_i. Therefore $g_ig|\psi\rangle = gg_i|\psi\rangle = g|\psi\rangle$. As a result, $g|\psi\rangle$ is itself in V_G. Therefore either $g|\psi\rangle = |\psi\rangle$ or $g|\psi\rangle = -|\psi\rangle$. So either g or $-g$ belongs to G. In the former case, we obtain $+1$, and in the latter case, we obtain -1. In either case, we don't have to update the stabilizer set. Deciding between $+1$ and -1 can be reduced to a linear algebra problem over $\mathbb{F}_2$. In particular, as described before, to any generator $g_i \in G$, we can assign a pair of strings $z_i := (x_i, y_i) \in \mathbb{F}_2^{2n}$ (omitting the phase information for now). Suppose g corresponds to $z := (x, y) \in \mathbb{F}_2^{2n}$. Since we know g or $-g$ is generated by $\langle g_1, \ldots, g_n\rangle$, then there should exist $(\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_2^n$ such that $g = g_1^{\alpha_1}\ldots g_n^{\alpha_n}$. Hence $\sum_i \alpha_i z_i = z_j$.

(2) If g anti-commutes with one or more generators. Without loss of generality, we can assume g anti-commutes with g_1 and commutes with the rest. The reason is that if it anticommutes with g_2, then we can replace g_2 with g_1g_2 which commutes with g. We claim that in this case, we obtain $+1$ half of the time and -1 half of the time. To see this note $Pr(+1) = \langle\psi|\frac{I+g}{2}|\psi\rangle = \langle\psi|\frac{I+g}{2}g_1|\psi\rangle = \langle\psi|g_1\frac{I-g}{2}|\psi\rangle = Pr(-1)$. We flip a coin. If it was heads, we sample $+1$ and replace g_1 with Z_1 and if it was tails, we sample -1 and replace g_1 with $-Z_1$.

Appendix A

Mathematical background

Here, we provide an overview of the mathematical concepts that are fundamental for understanding and working with quantum systems. The document covers complex numbers and linear algebra, which are crucial tools for representing quantum states and operators in a mathematical framework.

The first part of the document introduces complex numbers and their properties, such as the complex conjugate and modulus. The second part of the document covers linear algebra basics, including vector spaces, basis vectors, and linear transformations. It explains how these concepts are used to represent quantum states and operators in a mathematical framework. We then turn to vector spaces with inner products, which are used to compute probabilities and measure the similarity between quantum states.

Overall, this set of background mathematical notes provides a solid foundation for understanding the mathematical concepts and tools that are necessary for quantum computing.

We thank student Elene Ivaniashvili for scribing this chapter.

A.1 Complex Numbers

A.1.1 Complex Numbers

A complex number is a number of the form $z = a + bi$, where a and b are real numbers, and i is the imaginary unit, which has the property that $i^2 = -1$. The set of all complex numbers is denoted by $\mathbb{C}$.

Real and Imaginary Parts

For a complex number $z = a + bi$, the real part is denoted by $\mathrm{Re}(z) = a$, and the imaginary part is denoted by $\mathrm{Im}(z) = b$.

Complex Conjugate

The complex conjugate of a complex number $z = a + bi$ is the complex number $z^* = a - bi$, which is also written sometimes as $\bar{z}$. The complex conjugate has the following properties:

- $(z^*)^* = z$
- $zz^* = |z|^2$

Magnitude and Argument

The magnitude (or modulus) of a complex number $z = a + bi$ is denoted by $|z|$ and is defined as $|z| = \sqrt{a^2 + b^2}$. The argument (or phase) of a complex number $z = a + bi$ is denoted by $\arg(z)$ and is defined as the angle $\theta \in [0, 2\pi)$ such that $z = |z|(\cos(\theta) + i\sin(\theta))$. The argument is not a unique real-number, as it is defined modulo 2π.

Polar Form

A complex number $z = a + bi$ can be expressed in polar form as $z = r(\cos(\theta) + i\sin(\theta))$, where $r = |z|$ and $\theta = \arg(z)$.

Euler's Formula

Euler's formula states that for any real number θ,

$$e^{i\theta} = \cos(\theta) + i\sin(\theta) \tag{A.1}$$

Using Euler's formula, we can write the polar form of a complex number as $z = re^{i\theta}$.

A.1.2 Complex Number Arithmetic

Addition

The sum of two complex numbers is obtained by adding their real and imaginary parts separately:

$$(a + bi) + (c + di) = (a + c) + (b + d)i \tag{A.2}$$

Subtraction

The difference of two complex numbers is obtained by subtracting their real and imaginary parts separately:

$$(a + bi) - (c + di) = (a - c) + (b - d)i \tag{A.3}$$

Multiplication

The product of two complex numbers is obtained by expanding and simplifying the terms:

$$(a + bi)(c + di) = (ac - bd) + (ad + bc)i \tag{A.4}$$

Division

The division of two complex numbers is obtained by multiplying both the numerator and the denominator by the complex conjugate of the denominator and simplifying:

$$\frac{a + bi}{c + di} = \frac{(a + bi)(c - di)}{(c + di)(c - di)} = \frac{(ac + bd) + (bc - ad)i}{c^2 + d^2} \tag{A.5}$$

Complex Exponentiation

Using Euler's formula, we can define the exponentiation of a complex number:

$$z^n = (re^{i\theta})^n = r^n e^{in\theta} = r^n(\cos(n\theta) + i\sin(n\theta)) \tag{A.6}$$

A.1.3 Complex Functions

Certain complex functions are important in quantum computing, such as the complex exponential and trigonometric functions.

Complex Exponential Function

The complex exponential function is defined as:

$$f(z) = e^z = e^{a+bi} = e^a e^{bi} = e^a(\cos(b) + i\sin(b)) \tag{A.7}$$

Complex Trigonometric Functions

The complex sine and cosine functions are defined in terms of complex exponentials as:

$$\sin(z) = \frac{e^{iz} - e^{-iz}}{2i} = \sin(a)\cosh(b) + i\cos(a)\sinh(b) \tag{A.8}$$

$$\cos(z) = \frac{e^{iz} + e^{-iz}}{2} = \cos(a)\cosh(b) - i\sin(a)\sinh(b) \tag{A.9}$$

A.2 Bra-Ket Notation

A.2.1 Vectors in braket notation

Recall that vectors are elements of a vector space like $\mathbb{R}^n$ or $\mathbb{C}^n$. In the Dirac braket notation, a vector $\mathbf{v}$ is written as:

$$|v\rangle \tag{A.10}$$

where the vertical bars are called kets, and the vector is enclosed inside the ket. The ket vector is a notation for the column vector. The ket represents the vector as an abstract object, without specifying any particular coordinates or basis. For example, if $\mathbf{v}$ is a vector in $\mathbb{R}^3$, we can represent it as a ket vector:

$$|\mathbf{v}\rangle = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \tag{A.11}$$

where v_1, v_2, v_3 are the components of the vector in some chosen basis.

We can also represent vectors in bra notation, which is the complex-conjugated transpose of the corresponding ket, and is written as a row vector. The bra corresponding to the ket $|\mathbf{v}\rangle$ is denoted by $\langle\mathbf{v}|$ and is defined as:

$$\langle\mathbf{v}| = |\mathbf{v}\rangle^\dagger = \begin{pmatrix} v_1^* & v_2^* & v_3^* \end{pmatrix} \tag{A.12}$$

where † denotes the Hermitian conjugate, which is the transpose of the matrix with complex conjugate entries.

The inner product of two vectors $|\mathbf{v}\rangle$ and $|\mathbf{w}\rangle$ is denoted by $\langle\mathbf{v}|\mathbf{w}\rangle$ and is defined as:

$$\langle\mathbf{v}|\mathbf{w}\rangle = \mathbf{v}^\dagger\mathbf{w} = \sum_{i=1}^{n} v_i^* w_i \tag{A.13}$$

where n is the dimension of the vectors. The vectors $|\mathbf{v}\rangle$ and $|\mathbf{w}\rangle$ are both in C^n. i.e., they have same dimension. Note that the inner product of two vectors is a complex number.

The norm of a vector $|\mathbf{v}\rangle$ is denoted by $\||\mathbf{v}\rangle\|$ and is defined as:

$$\||\mathbf{v}\rangle\| = \sqrt{\langle\mathbf{v}|\mathbf{v}\rangle} = \sqrt{\sum_{i=1}^{n} |v_i|^2} \tag{A.14}$$

where $|v_i|$ denotes the absolute value of the vector entry v_i.

A.2.2 Operators in braket notation

In braket notation, a matrix operator $\hat{A}$ is represented as:

$$\hat{P} = |A\rangle\langle A| \tag{A.15}$$

where $|A\rangle$ is a ket and $\langle A|$ is a bra. The operator $\hat{A}$ maps a ket $|\mathbf{v}\rangle$ to another ket $\hat{A}|\mathbf{v}\rangle$, which is defined as:

$$\hat{A}|\mathbf{v}\rangle = |A\rangle\langle A|\mathbf{v}\rangle \tag{A.16}$$

*Note that the result of applying an operator to a ket is also a ket. The complex-conjugated transpose, or **adjoint** of an operator $\hat{A}$ is denoted by $\hat{A}^\dagger$ and is defined as:*

$$\hat{A}^\dagger = |A\rangle^\dagger\langle A| = \langle A|\hat{A}|A\rangle \tag{A.17}$$

The adjoint of an operator is also an operator, and it satisfies the following properties:

$$(\hat{A}^\dagger)^\dagger = \hat{A} \tag{A.18}$$

$$(\hat{A}\hat{B})^\dagger = \hat{B}^\dagger\hat{A}^\dagger \tag{A.19}$$

$$(\hat{A} + \hat{B})^\dagger = \hat{A}^\dagger + \hat{B}^\dagger \tag{A.20}$$

An operator is said to be Hermitian if it is equal to its own adjoint, i.e., $\hat{A}^\dagger = \hat{A}$. In braket notation, a Hermitian operator $\hat{A}$ is represented as:

$$\hat{A} = \hat{A}^\dagger = |A\rangle\langle A| \tag{A.21}$$

where $|A\rangle$ is a ket. Note that a Hermitian operator is always diagonalizable and has real eigenvalues (see appendix A.8.3 for a recap of these concepts).

A unitary operator is an operator that preserves the inner product of vectors, i.e., it satisfies the condition:

$$\langle \mathbf{u}|\mathbf{v}\rangle = \langle \hat{U}\mathbf{u}|\hat{U}\mathbf{v}\rangle \tag{A.22}$$

for all vectors $|\mathbf{u}\rangle$ and $|\mathbf{v}\rangle$. In braket notation, a unitary operator $\hat{U}$ is represented as:

$$\hat{U}\hat{U}^\dagger = \hat{U}^\dagger\hat{U} = \hat{I} \tag{A.23}$$

where $\hat{I}$ is the identity operator, and $\hat{U}^\dagger$ is the adjoint of $\hat{U}$.

A.2.3 Examples

Here are some examples of how braket notation is used in linear algebra:

- *The projection operator onto a subspace V of a vector space W is given by:*

$$\hat{P_V} = \sum_{i=1}^{n} |v_i\rangle\langle v_i| \tag{A.24}$$

  where $|v_i\rangle$ are basis vectors of V.

- *The identity operator in a vector space is given by:*

$$\hat{I} = \sum_{i=1}^{n} |e_i\rangle\langle e_i| \tag{A.25}$$

  where $|e_i\rangle$ represent the standard basis vectors of the vector space, i.e.,

$$|e_1\rangle = (1, 0, 0, \cdots, 0) \tag{A.26}$$
$$|e_2\rangle = (0, 1, 0, \cdots, 0) \tag{A.27}$$

  and so on.

- *The Pauli matrices in quantum mechanics are given by:*

$$\sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0| \tag{A.28}$$
$$\sigma_y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = -i|0\rangle\langle 1| + i|1\rangle\langle 0| \tag{A.29}$$
$$\sigma_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1| \tag{A.30}$$

  See appendix A.3.4 for an explanation of the notation.

A.3 Scalars, Vectors, and Matrices

A.3.1 Scalars

Scalars in quantum computing are complex numbers, denoted as $c \in \mathbb{C}$. Scalars are used to represent the probability amplitudes of quantum states and the elements of matrices that represent quantum operators. Complex numbers can be written in the form $c = a + bi$, where $a, b \in \mathbb{R}$ and i is the imaginary unit, satisfying $i^2 = -1$.

A.3.2 Vectors

Vectors in quantum computing are elements of a complex vector space. Quantum states are represented as column vectors called state vectors. For a quantum system with n basis states (e.g., qubits), the quantum state vector is an element of $\mathbb{C}^n$. The quantum states can be expressed as linear combinations of the orthonormal basis vectors:

$$|\psi\rangle = \sum_{i=1}^{n} c_i |\mathbf{e}_i\rangle \tag{A.31}$$

where $c_i \in \mathbb{C}$ are the complex coefficients, and $|\mathbf{e}_i\rangle$ are the basis vectors.

A.3.3 Matrices

Matrices in quantum computing are used to represent linear operators that act on quantum states. A quantum operator is represented by a square matrix $A \in \mathbb{C}^{n \times n}$, which acts on a quantum state vector $|\psi\rangle \in \mathbb{C}^n$ to produce a new quantum state vector $|\phi\rangle \in \mathbb{C}^n$:

$$|\phi\rangle = A|\psi\rangle \tag{A.32}$$

Quantum observables are represented by Hermitian matrices, which are matrices that are equal to their conjugate transpose. Unitary matrices are used to represent quantum gates and time evolution operators. A unitary matrix U satisfies $UU^\dagger = U^\dagger U = I$, where I is the identity matrix and $U^\dagger$ is the conjugate transpose of U.

Basic Matrix Operations

In this section, we briefly review some basic matrix operations that are important in quantum computing.

Matrix Addition and Subtraction

Two matrices of the same size can be added or subtracted element-wise:

$$(A \pm B)_{ij} = A_{ij} \pm B_{ij} \tag{A.33}$$

Matrix Multiplication

Matrix multiplication is a binary operation that takes a pair of matrices and produces another matrix. If $A \in \mathbb{C}^{n \times m}$ and $B \in \mathbb{C}^{m \times p}$, then their product $AB \in \mathbb{C}^{n \times p}$ is defined as:

$$(AB)ij = \sum k = 1^m A_{ik} B_{kj} \tag{A.34}$$

Matrix multiplication is associative but not, in general, commutative, meaning that $(AB)C = A(BC)$*, but* $AB \neq BA$ *in general.*

Conjugate Transpose

The conjugate transpose of a complex matrix $A \in \mathbb{C}^{n \times m}$*, denoted as* $A^\dagger$*, is obtained by taking the transpose of the matrix and then taking the complex conjugate of each element:*

$$A^\dagger_{ij} = \overline{A}_{ji} \tag{A.35}$$

where $\overline{A}ji$ *is the complex conjugate of* Aji*.*

A.3.4 Matrix representation of quantum computations

Using the matrix representation of computations, we see that a classical state is a vector of zeros and ones such that one entry is 1 *and the rest of zeros. We could view this as a vector of zeros and ones such that the sum of (squares) of entries is* 1*. We saw that a probability vector is a vector of non-negative numbers that sum to* 1*. Classical states were special cases of probability vectors. If we ask a state to have complex number square summing to* 1 *we get quantum states. Physically, we can encode a quantum bit within the degrees of freedom of a physical system: Electron spin up or down, photon polarization being clockwise or counter clockwise. Mathematically we have.*

- *Vector notation* $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$*. E.g.* $|0\rangle$ *could mean spin up and* $|1\rangle$ *spin down.*
- *Superposition:* $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ *satisfying* $|\alpha|^2 + |\beta|^2 = 1$*.*
- ***Example:*** $|+\rangle := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ *and* $|-\rangle := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$
- *Normalization* $|\psi\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \cdots \\ \alpha_{N-1} \end{pmatrix}$ *then* $\sum_i |\alpha_i|^2 = 1$

A.4 Vector Space

A vector space is a set of objects (vectors) that can be added together and multiplied by scalars (complex numbers in quantum computing), and it follows a set of rules called axioms. In quantum computing, vector spaces are used to represent the state space of quantum systems.

A.4.1 Hilbert Space

In quantum computing, a complex Hilbert space is used as the vector space to describe quantum states. A Hilbert space is a complex vector space equipped with an inner product, which allows

us to define the distance and angle between vectors. It also has the property that it is complete, meaning that any Cauchy sequence of vectors in the space converges to a limit in the space.

A.4.2 Basis Vectors and Linear Combinations

A basis of a vector space is a set of linearly independent vectors that span the entire space. In other words, every vector in the space can be expressed as a unique linear combination of the basis vectors. For a quantum system with n basis states (e.g., qubits), the quantum state vector is an element of $\mathbb{C}^n$. The quantum states can be expressed as linear combinations of the orthonormal basis vectors:

$$|\psi\rangle = \sum_{i=1}^{n} c_i |\mathbf{e}_i\rangle \tag{A.36}$$

where $c_i \in \mathbb{C}$ are the complex coefficients, and $|\mathbf{e}_i\rangle$ are the basis vectors.

A.4.3 Superposition

Superposition is a fundamental concept in quantum mechanics, which is a direct consequence of the vector space structure of quantum states. Superposition states that a quantum system can exist in multiple states simultaneously. Mathematically, this means that a quantum state vector can be a linear combination of basis vectors:

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \tag{A.37}$$

where $c_0, c_1 \in \mathbb{C}$ are probability amplitudes, and $|0\rangle$ and $|1\rangle$ are basis vectors.

A.4.4 Linear Combinations and Span

A linear combination of a set of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ is an expression of the form:

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n \tag{A.38}$$

where $c_1, c_2, \ldots, c_n$ are scalars. The span of a set of vectors is the set of all possible linear combinations of those vectors:

$$span\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n : c_1, c_2, \ldots, c_n \in \mathbb{F} \tag{A.39}$$

where $\mathbb{F}$ is the field of scalars (usually the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$). The span of a set of vectors is always a subspace of the vector space.

A.4.5 Linear Independence and Dependence

A set of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ is linearly independent if the only linear combination that equals the zero vector is the trivial linear combination (i.e., all coefficients are zero):

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n = \mathbf{0} \implies c_1 = c_2 = \cdots = c_n = 0 \tag{A.40}$$

If there exists a non-trivial linear combination that equals the zero vector, the set of vectors is linearly dependent.

A.4.6 Basis and Dimension

A basis of a vector space is a set of linearly independent vectors that spans the vector space. In other words, every vector in the vector space can be uniquely expressed as a linear combination of the basis vectors.

The dimension of a vector space is the number of vectors in any basis of the vector space. The dimension is denoted as $\dim(V)$.

A.4.7 Orthogonality and Orthonormality

Two vectors are orthogonal if their dot product (inner product) is zero:

$$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} u_i v_i = 0 \tag{A.41}$$

A set of vectors is orthogonal if every pair of distinct vectors in the set is orthogonal.

A set of vectors is orthonormal if it is orthogonal and all the vectors in the set have a norm (magnitude) of 1.

An orthogonal basis is a basis in which all the basis vectors are orthogonal. An orthonormal basis is a basis in which all the basis vectors are orthonormal.

A.4.8 Gram-Schmidt Process

The Gram-Schmidt process is a method for orthogonalizing a set of vectors in an inner product space. It is commonly used in linear algebra and is particularly useful for constructing orthonormal bases.

Description

Given a set of n linearly independent vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ *in an inner product space, the Gram-Schmidt process produces a set of n orthogonal vectors* $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$, *where* $\mathbf{u}_1 = \mathbf{v}_1$ *and* $\mathbf{u}_i$ *is obtained by subtracting from* $\mathbf{v}_i$ *its projection onto the subspace spanned by* $\mathbf{u}_1, \mathbf{u}2, \ldots, \mathbf{u}i - 1$ *and then normalizing the result:*

$$\mathbf{u}_1 = \mathbf{v}_1 \ \mathbf{u}_i \qquad = \mathbf{v}i - \sum j = 1^{i-1} \frac{\langle \mathbf{u}_j, \mathbf{v}_i \rangle}{\langle \mathbf{u}_j, \mathbf{u}_j \rangle} \mathbf{u}_j, \quad i = 2, 3, \ldots, n.$$

Here, $\langle \cdot, \cdot \rangle$ *denotes the inner product, which is a bilinear form that satisfies certain properties, such as linearity in the first argument and conjugate symmetry.*

After applying the Gram-Schmidt process, the resulting set of vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ *is an orthonormal basis for the subspace spanned by the original set of vectors* $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$.

Instructions

To apply the Gram-Schmidt process to a set of vectors, follow these steps:

1. *Start with the first vector* $\mathbf{v}_1$ *and set* $\mathbf{u}_1 = \mathbf{v}_1$.

2. *For $i = 2, 3, \ldots, n$, compute $\mathbf{u}_i$ using the formula above, where $\langle \cdot, \cdot \rangle$ denotes the inner product.*

3. *Normalize each $\mathbf{u}_i$ by dividing it by its norm: $\mathbf{u}_i = \frac{\mathbf{u}_i}{|\mathbf{u}_i|}$, where $|\cdot|$ denotes the norm induced by the inner product.*

4. *The resulting set of vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ is an orthonormal basis for the subspace spanned by the original set of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$.*

A.5 Inner Product, Norm, and Outer Product

A.5.1 Inner Product

The inner product is a map that takes two vectors as input and returns a scalar. In quantum computing, we deal with complex vector spaces of the form $V = \mathbb{C}^d$ where d is the dimension of the space, so the inner product is defined as follows:

$$\langle \cdot | \cdot \rangle : V \times V \to \mathbb{C} \tag{A.42}$$

$$\langle \psi | \phi \rangle = \sum_{i=1}^{n} \overline{c_i} d_i \tag{A.43}$$

where $|\psi\rangle = \sum_{i=1}^{n} c_i |\mathbf{e}_i\rangle$ and $|\phi\rangle = \sum_{i=1}^{n} d_i |\mathbf{e}_i\rangle$ are quantum states, and $\overline{c_i}$ is the complex conjugate of c_i.

The inner product satisfies the following properties for all $|\psi\rangle, |\phi\rangle \in V$ and all complex scalars $a, b \in \mathbb{C}$:

- *Conjugate symmetry: $\langle \psi | \phi \rangle = \overline{\langle \phi | \psi \rangle}$*
- *Linearity: $\langle \psi | (a\phi_1 + b\phi_2) \rangle = a\langle \psi | \phi_1 \rangle + b\langle \psi | \phi_2 \rangle$*
- *Positivity: $\langle \psi | \psi \rangle > 0$*
- *Definiteness: $\langle \psi | \psi \rangle = 0$ if and only if $|\psi\rangle = 0$*

Note that conjugate symmetry implies that for all $|x\rangle \in V$, the inner-product with itself $\langle x | x \rangle$ is a real number.

A.5.2 Norm

The norm of a vector is a measure of its magnitude or length. In quantum computing, the norm of a quantum state vector $|\psi\rangle$ is given by the square root of the inner product of the vector with itself:

$$\| |\psi\rangle \| = \sqrt{\langle \psi | \psi \rangle} \tag{A.44}$$

A quantum state is said to be normalized if its norm is equal to 1. Normalized quantum states are important because their coefficients (probability amplitudes) can be used to compute probabilities of measurement outcomes.

A.5.3 Cauchy-Schwarz Inequality

The Cauchy-Schwarz inequality is an important result relating the inner product and the norm. It states that for any two vectors $\mathbf{u}$ *and* $\mathbf{v}$ *in an inner product space:*

$$|\langle \mathbf{u}|\mathbf{v}\rangle|^2 \leq |\mathbf{u}|^2|\mathbf{v}|^2 \tag{A.45}$$

A.5.4 Orthogonality

Two vectors are orthogonal if their inner product is zero:

$$\langle \mathbf{u}|\mathbf{v}\rangle = 0 \tag{A.46}$$

A.5.5 Projection

The projection of a vector $\mathbf{u}$ *onto another vector* $\mathbf{v}$ *is defined as:*

$$proj_{\mathbf{v}}\mathbf{u} = \frac{\langle \mathbf{u}|\mathbf{v}\rangle}{|\mathbf{v}|^2}\mathbf{v} \tag{A.47}$$

A.5.6 Outer Product

The outer product is a function that takes two vectors as input and returns a matrix. In quantum computing, the outer product of two quantum state vectors $|\psi\rangle$ *and* $|\phi\rangle$ *is defined as:*

$$|\psi\rangle\langle\psi| = \begin{pmatrix} \psi_1\overline{\psi_1} & \psi_1\overline{\psi_2} & \cdots & \psi_1\overline{\psi_n} \\ \psi_2\overline{\psi_1} & \psi_2\overline{\psi_2} & \cdots & \psi_2\overline{\psi_n} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_n\overline{\psi_1} & \psi_n\overline{\psi_2} & \cdots & \psi_n\overline{\psi_n} \end{pmatrix} \tag{A.48}$$

where $|\psi\rangle = \sum_{i=1}^{n} c_i|\mathbf{e}i\rangle$ *and* $|\phi\rangle = \sum i = 1^n d_i|\mathbf{e}_i\rangle$ *are quantum states.*
The outer product has the following properties:

- *Linearity:* $(a|\psi_1\rangle + b|\psi_2\rangle)\langle\phi| = a|\psi_1\rangle\langle\phi| + b|\psi_2\rangle\langle\phi|$
- *Linearity:* $|\psi\rangle\langle(a\phi_1 + b\phi_2)| = a|\psi\rangle\langle\phi_1| + b|\psi\rangle\langle\phi_2|$

A.6 Tensor Product

The tensor product, also known as the Kronecker product or the outer product, is an essential mathematical tool in quantum computing. It is used to describe the combined state of multiple qubits and to construct multi-qubit gates. In this document, we present the definition, properties, and applications of the tensor product in quantum computing.

A.6.1 Definition

Given two matrices A of size $m \times n$ and B of size $p \times q$, the tensor matrix product of A and B, denoted by $A \otimes B$, is a matrix of size $(mp \times nq)$ defined as:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}. \tag{A.49}$$

A.6.2 Properties

The tensor product has several important properties, including:

1. ***Bilinearity:*** *The tensor product is bilinear, meaning it is linear in both factors:*

$$(A + A') \otimes B = A \otimes B + A' \otimes B \tag{A.50}$$
$$A \otimes (B + B') = A \otimes B + A \otimes B' \tag{A.51}$$

   for any matrices A, A', B, and B' of compatible dimensions.

2. ***Associativity:*** *The tensor product is associative when applied to finite-dimensional matrices, meaning:*

$$(A \otimes B) \otimes C = A \otimes (B \otimes C) \tag{A.52}$$

   for any matrices A, B, and C of compatible dimensions. So we may omit brackets when taking tensor products.

3. ***Distributivity over Matrix Multiplication:*** *The tensor product distributes over matrix multiplication:*

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{A.53}$$

   for any matrices A, B, C, and D of compatible dimensions.

4. ***Identity:*** *The identity element for the tensor product is the 1×1 identity matrix I_1:*

$$A \otimes I_1 = I_1 \otimes A = A \tag{A.54}$$

   for any matrix A.

5. ***Transpose:*** *The transpose of a tensor product is given by:*

$$(A \otimes B)^T = A^T \otimes B^T \tag{A.55}$$

   for any matrices A and B.

6. ***Conjugate:*** *The conjugate of a tensor product is given by:*

$$(A \otimes B)^* = A^* \otimes B^* \tag{A.56}$$

   for any matrices A and B.

7. ***Adjoint:*** *The adjoint of a tensor product is given by:*

$$(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger \tag{A.57}$$

   for any matrices A and B.

8. ***Determinant:*** *The determinant of a tensor product of square matrices is given by:*

$$\det(A \otimes B) = (\det(A))^m (\det(B))^n \tag{A.58}$$

where A is an $n \times n$ matrix, B is an $m \times m$ matrix, and both A and B have compatible dimensions.

9. ***Trace:*** *The trace of a tensor product is given by:*

$$Tr(A \otimes B) = Tr(A)\, Tr(B) \tag{A.59}$$

for any square matrices A and B of compatible dimensions.

A.6.3 Multi-Qubit States

In quantum computing, the tensor product is used to represent the combined state of multiple qubits. Given two qubits in states $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ and $|\phi\rangle = \begin{pmatrix} \gamma \\ \delta \end{pmatrix}$, their combined state is described by the tensor product $|\psi\rangle \otimes |\phi\rangle$, which is a 4×1 column vector:

$$|\psi\rangle \otimes |\phi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{pmatrix} \tag{A.60}$$

For n qubits, the combined state is an $2^n \times 1$ column vector, which can be written as the tensor product of the individual qubit states. For example, for three qubits in states $|\psi\rangle$, $|\phi\rangle$, and $|\chi\rangle$, the combined state is:

$$|\psi\rangle \otimes |\phi\rangle \otimes |\chi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \delta \end{pmatrix} \otimes \begin{pmatrix} \epsilon \\ \zeta \end{pmatrix} = \begin{pmatrix} \alpha\gamma\epsilon \\ \alpha\gamma\zeta \\ \alpha\delta\epsilon \\ \alpha\delta\zeta \\ \beta\gamma\epsilon \\ \beta\gamma\zeta \\ \beta\delta\epsilon \\ \beta\delta\zeta \end{pmatrix} \tag{A.61}$$

A.6.4 Multi-Qubit Gates

The tensor product is also used to construct multi-qubit gates by combining single-qubit gates or other multi-qubit gates. For example, given two single-qubit gates U and V, their combined action on a two-qubit state can be represented as:

$$(U \otimes V)|\psi\rangle \otimes |\phi\rangle = U|\psi\rangle \otimes V|\phi\rangle. \tag{A.62}$$

As an example, the combined action of two Hadamard gates H on a two-qubit state is given by:

$$(H \otimes H)|\psi\rangle \otimes |\phi\rangle = H|\psi\rangle \otimes H|\phi\rangle. \tag{A.63}$$

For controlled gates, such as the CNOT gate, the tensor product is used to express the gate as a matrix that acts on the combined state of the control and target qubits:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{A.64}$$

See appendix A.10.4 for more details.

A.7 Matrix Operations

A.7.1 Matrix Addition and Subtraction

Matrix addition and subtraction are performed element-wise. If $A \in \mathbb{C}^{n\times m}$ and $B \in \mathbb{C}^{n\times m}$, then their sum $A + B$ and difference $A - B$ are given by:

$$(A \pm B)_{ij} = A_{ij} \pm B_{ij} \tag{A.65}$$

A.7.2 Matrix Multiplication

Matrix multiplication is the primary operation in quantum computing, as it is used to describe the action of quantum gates and operators. If $A \in \mathbb{C}^{n\times m}$ and $B \in \mathbb{C}^{m\times p}$, then their product $AB \in \mathbb{C}^{n\times p}$ is defined as:

$$(AB)_{ij} = \sum_{k=1}^{m} A_{ik}B_{kj} \tag{A.66}$$

Matrix multiplication is associative but not commutative, meaning that $(AB)C = A(BC)$, but $AB \neq BA$ in general.

A.7.3 Transpose

The transpose of a matrix A of size $m \times n$ is a matrix A^T of size $n \times m$, and its elements are defined as:

$$A^T_{ij} = A_{ji} \tag{A.67}$$

for all $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. The transpose operation has the following properties:

$$(A^T)^T = A \quad (A+B)^T = A^T + B^T \quad (cA)^T = c(A^T) \quad (AB)^T = B^T A^T \tag{A.68}$$

A.7.4 Matrix Inversion

The inverse of a square matrix A of size $n \times n$ is a matrix A^{-1} of the same size, such that their product is the identity matrix I_n:

$$AA^{-1} = A^{-1}A = I_n \tag{A.69}$$

Not all matrices have an inverse; a matrix is called invertible or nonsingular if it has an inverse, and non-invertible or singular if it does not. If a matrix is invertible, its inverse is unique. The matrix inversion operation has the following properties:

$$(A^{-1})^{-1} = A \quad (AB)^{-1} = B^{-1}A^{-1} \quad (A^T)^{-1} = (A^{-1})^T \quad (cA)^{-1} = \frac{1}{c}A^{-1} \quad \text{for nonzero } c \tag{A.70}$$

A.7.5 Determinant

The determinant is a scalar function that takes a square matrix and returns a scalar value. The determinant of a 2×2 matrix A is defined as:

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \tag{A.71}$$

For an $n \times n$ matrix A, the determinant can be calculated using the Laplace expansion, which is a recursive formula:

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(A_{ij}) \tag{A.72}$$

where A_{ij} is the $(n-1) \times (n-1)$ matrix obtained by deleting the i-th row and j-th column of A. The determinant has the following properties:

$$\det(A^T) = \det(A) \quad \det(AB) = \det(A)\det(B) \quad \det(A^{-1}) = \frac{1}{\det(A)} \quad \text{if } A \text{ is invertible} \tag{A.73}$$

A.7.6 Trace

The trace of a matrix is an important concept with various applications. The trace of a square matrix A of size $n \times n$ is defined as the sum of its diagonal elements:

$$Tr(A) = \sum_{i=1}^{n} A_{ii}. \tag{A.74}$$

In the context of quantum mechanics, the trace often appears in calculations involving density matrices, which describe the state of a quantum system. For instance, the trace of a density matrix ρ is always equal to 1, representing the total probability of the system:

$$Tr(\rho) = 1. \tag{A.75}$$

Moreover, the trace operation is used to compute expectation values of observables, which are represented by Hermitian matrices. Given an observable O and a quantum state represented by a density matrix ρ, the expectation value of the observable is given by:

$$\langle O \rangle = Tr(O\rho). \tag{A.76}$$

A.7.7 Identity Matrix

The identity matrix $I \in \mathbb{C}^{n\times n}$ is a square matrix with ones on the diagonal and zeros elsewhere:

$$I_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{A.77}$$

The identity matrix has the property that $AI = IA = A$ for any matrix $A \in \mathbb{C}^{n\times n}$.

A.7.8 Conjugate Transpose

The conjugate transpose of a complex matrix $A \in \mathbb{C}^{n\times m}$, denoted as $A^\dagger$, is obtained by taking the transpose of the matrix and then taking the complex conjugate of each element:

$$A^\dagger_{ij} = \overline{A}_{ji} \tag{A.78}$$

where $\overline{A}_{ji}$ is the complex conjugate of A_{ji}.

A.7.9 Hermitian Matrices

A Hermitian matrix $H \in \mathbb{C}^{n\times n}$ is a square matrix that is equal to its conjugate transpose:

$$H = H^\dagger \tag{A.79}$$

Hermitian matrices have real eigenvalues and play an important role in quantum mechanics, as they represent observable quantities in quantum systems.

A.8 Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors are essential concepts in linear algebra that provide insight into the behavior of linear transformations. Given a square matrix A of size $n \times n$, a scalar λ is an eigenvalue of A if there exists a non-zero vector $\mathbf{v}$ such that:

$$A\mathbf{v} = \lambda\mathbf{v} \tag{A.80}$$

The vector $\mathbf{v}$ is called an eigenvector corresponding to the eigenvalue λ.

A.8.1 Characteristic Equation

To find the eigenvalues of a matrix A, we can rewrite the eigenvalue equation as follows:

$$(A - \lambda I_n)\mathbf{v} = \mathbf{0} \tag{A.81}$$

where I_n is the identity matrix of size $n \times n$. For a non-trivial solution $\mathbf{v}$, the matrix $(A - \lambda I_n)$ must be singular, which means that its determinant is zero:

$$\det(A - \lambda I_n) = 0 \tag{A.82}$$

This equation is called the characteristic equation of the matrix A. Solving it yields the eigenvalues of A.

A.8.2 Finding Eigenvectors

Once the eigenvalues have been found, the corresponding eigenvectors can be obtained by solving the following system of linear equations:

$$(A - \lambda I_n)\mathbf{v} = \mathbf{0} \tag{A.83}$$

for each eigenvalue λ.

A.8.3 Diagonalizing a matrix

*A square matrix A is called **diagonalizable** if there exists an invertible matrix V and a diagonal matrix D such that $A = VDV^{-1}$. Note that in general, V and D are non-unique.*

Here is how you diagonalize a matrix: For a complex matrix $A \in \mathbb{C}^{d\times d}$ the eigenvalues of A are numbers λ such that $A - \lambda I$ is singular. That means $det(A - \lambda I) = 0$; that means you have to solve this equation for λ. For a 2×2 matrix, the determinant is the product of entries on the diagonal minus the product of off-diagonal entries. For instance in order to find the eigenvalues of X you should solve $det(X - \lambda I) = 0$ which gives you $(-\lambda)(-\lambda) - (1)(1) = 0$. Which gives you $\lambda = \pm 1$ as its solution.

Once you have found the eigenvalues, it is time to find eigenvectors. The eigenvector $|v\rangle$ corresponding to eigenvalue λ satisfies $(A - \lambda I)|v\rangle = 0$. You should write this as a system of equations. If λ is a unique eigenvalue, you will find the entries of $|v\rangle$ up to a free parameter. You can set that free parameter to make $|v\rangle$ have a unit norm. For instance, in order to find the eigenvector corresponding to eigenvalue $+1$ for the Pauli X matrix from appendix A.2.3, you have to solve

$$\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = 0$$

Solving this you will find that $a = b$, and in order to normalize the vector you can choose $a = b = 1/\sqrt{2}$.

The next step is to find a unitary matrix O that diagonalizes A, ie, $OAO^-1 = D$ where D is the diagonal matrix with the eigenvalues of A. If you think about it, O^{-1} needs to be a matrix that maps the basis A is defined to its eigenbasis. So to construct O, we place each eigenvector as its columns. For instance, in the case of X, the eigenvectors are $|+\rangle$ and $|-\rangle$. So

$$O = (|+\rangle|-\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Which is the Hadamard matrix, as we expected.

A.8.4 Eigenvalues and Matrix Powers

If a matrix A is diagonalizable, its powers can be computed easily using the diagonal form:

$$A^k = PD^kP^{-1} \tag{A.84}$$

for any positive integer k. This property is useful for computing the exponential of a matrix, which has applications in solving systems of linear differential equations.

A.9 Unitary Matrices

A unitary matrix $U \in \mathbb{C}^{n\times n}$ is a square matrix that satisfies the following condition:

$$UU^\dagger = U^\dagger U = I \tag{A.85}$$

where $U^\dagger$ denotes the conjugate transpose of U, and I is the identity matrix. Unitary matrices preserve the inner product and norms of vectors, making them essential for describing the evolution of quantum states in quantum computing.

Unitary matrices are the complex analogs of orthogonal matrices, which are matrices with real entries satisfying $A^TA = AA^T = I_n$. Unitary matrices preserve the inner product between vectors, which means that for any vectors $\mathbf{v}, \mathbf{w} \in \mathbb{C}^n$, the following holds:

$$\langle U\mathbf{v}, U\mathbf{w}\rangle = \langle \mathbf{v}, \mathbf{w}\rangle \tag{A.86}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product in the complex vector space $\mathbb{C}^n$.

A.9.1 Properties of Unitary Matrices

Unitary matrices have several important properties:

- *The product of two unitary matrices is unitary:*

$$(UV)^\dagger UV = V^\dagger U^\dagger UV = V^\dagger VU^\dagger U = I_n \tag{A.87}$$

- *The inverse of a unitary matrix is unitary:*

$$(U^{-1})^\dagger U^{-1} = U^\dagger U = I_n \tag{A.88}$$

- *The determinant of a unitary matrix is a complex number with absolute value 1:*

$$|\det(U)| = 1 \tag{A.89}$$

- *The eigenvalues of a unitary matrix are complex numbers with absolute value 1.*

A.9.2 Hermitian Matrices

A square matrix A of size $n\times n$ with complex entries is called Hermitian if its conjugate transpose $A^\dagger$ (also denoted as $A^\dagger$) is equal to itself:

$$A^\dagger = A \tag{A.90}$$

Hermitian matrices are the complex analogs of symmetric matrices, which are matrices with real entries satisfying $A^T = A$. The eigenvalues of a Hermitian matrix are always real.

A.9.3 Unitary Diagonalization

A Hermitian matrix A of size $n \times n$ can be diagonalized by a unitary matrix U:

$$A = UDU^\dagger \tag{A.91}$$

where D is a diagonal matrix with the eigenvalues of A on its diagonal, and the columns of U are the eigenvectors of A corresponding to the eigenvalues in D. This process is called unitary diagonalization.

The diagonalization process can be summarized as follows:

1. *Find the eigenvalues λ_i of the Hermitian matrix A.*
2. *For each eigenvalue λ_i, find a corresponding eigenvector $\mathbf{v}_i$ by solving the equation $(A - \lambda_i I_n)\mathbf{v}_i = 0$.*
3. *Normalize the eigenvectors and form the unitary matrix U with the normalized eigenvectors as its columns.*
4. *Form the diagonal matrix D with the eigenvalues λ_i on its diagonal.*
5. *Verify that $A = UDU^\dagger$.*

A.9.4 Unitary Transformations

Unitary matrices represent unitary transformations, which are linear transformations that preserve the inner product of vectors. Given two vectors $|\psi\rangle$ and $|\phi\rangle$ in a complex vector space, a unitary transformation U satisfies the following property:

$$\langle U\psi | U\phi \rangle = \langle \psi | \phi \rangle \tag{A.92}$$

Unitary transformations preserve the orthogonality and norms of vectors, ensuring that quantum states remain normalized after the application of quantum gates.

A.9.5 Unitary Matrices and Quantum Gates

In quantum computing, unitary matrices are used to represent quantum gates, which are the basic building blocks of quantum circuits. Quantum gates operate on quantum states, which are represented as unit vectors in a complex Hilbert space. Since unitary matrices preserve inner products and norms, they ensure that quantum gates maintain the normalization of quantum states.

A quantum gate U is a unitary matrix acting on a quantum state $|\psi\rangle$:

$$|\psi'\rangle = U|\psi\rangle \tag{A.93}$$

where $|\psi'\rangle$ is the resulting quantum state after applying the gate.

Examples of Quantum Gates

Some common quantum gates represented by unitary matrices include:

- *Identity gate (I):*

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{A.94}$$

- *Pauli-X gate (X), also known as the NOT gate:*

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{A.95}$$

- *Pauli-Y gate (Y):*

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{A.96}$$

- *Pauli-Z gate (Z):*

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{A.97}$$

- *Hadamard gate (H):*

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{A.98}$$

A.10 Special Matrices in Quantum Computation

A.10.1 Pauli Matrices

The Pauli matrices are a set of three 2×2 matrices that are widely used in quantum computing. They are defined as follows:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{A.99}$$

These matrices are also commonly written as the X, Y and Z matrices. The Pauli matrices have the following properties:

- *They are Hermitian: $\sigma_x^\dagger = \sigma_x$, $\sigma_y^\dagger = \sigma_y$, $\sigma_z^\dagger = \sigma_z$.*
- *They are unitary: $\sigma_x^\dagger \sigma_x = \sigma_y^\dagger \sigma_y = \sigma_z^\dagger \sigma_z = I$.*
- *Their square is the identity matrix: $\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I$.*
- *They anti-commute: $\sigma_x \sigma_y \sigma_z = -\sigma_y \sigma_x \sigma_z = -\sigma_z \sigma_y \sigma_x = -\sigma_y \sigma_z \sigma_x = -\sigma_x \sigma_z \sigma_y = -\sigma_z \sigma_x \sigma_y$.*

A.10.2 Hadamard Gate

The Hadamard gate is a 2×2 matrix that is used to create superpositions in quantum computing. It is defined as:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}. \tag{A.100}$$

The Hadamard gate has the following properties:

- *It is Hermitian: $H^\dagger = H$.*
- *It is unitary: $H^\dagger H = I$.*
- *Its square is the identity matrix: $H^2 = I$.*

A.10.3 Phase Gates

Phase gates are a family of 2×2 matrices that introduce a relative phase between the basis states. The most common phase gates are the S and T gates, defined as:

$$S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, \quad T = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix} \tag{A.101}$$

In general, a phase gate can be represented as:

$$R_\theta = \begin{pmatrix}1 & 0\\ 0 & e^{i\theta}\end{pmatrix}$$

where θ is the phase angle.

Phase gates have the following properties:

- *They are Hermitian: $S^\dagger = S$, $T^\dagger = T$, and $P(\theta)^\dagger = P(\theta)$.*
- *They are unitary: $S^\dagger S = T^\dagger T = P(\theta)^\dagger P(\theta) = I$.*
- *The S and T gates satisfy: $S^2 = \sigma_z$, $T^4 = \sigma_z$, and $T^8 = I$.*

A.10.4 Controlled Gates

Controlled gates act on two qubits and perform an operation on the target qubit if the control qubit is in the $|1\rangle$ state. The most common controlled gate is the Controlled-NOT (CNOT) gate, which is defined as:

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

The controlled version of a unitary matrix U is given by:

$$CU = \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix}$$

where C is the controlled gate.

Controlled gates have the following properties:

- *They are unitary:* $\mathrm{CNOT}^\dagger\mathrm{CNOT} = I$ *and* $\mathrm{C}(U)^\dagger\mathrm{C}(U) = I$.
- *The CNOT gate can be expressed in terms of the Pauli matrices as* $\mathrm{CNOT} = I \otimes \frac{1}{2}(\sigma_z + I) + \sigma_x \otimes \frac{1}{2}(\sigma_z - I)$.

A.10.5 SWAP Gate

The Swap gate exchanges the states of two qubits. It is represented by the following 4×4 matrix:

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

The Swap gate has the following properties:

- *It is Hermitian:* $\mathrm{SWAP}^\dagger = \mathrm{SWAP}$.
- *It is unitary:* $\mathrm{SWAP}^\dagger\mathrm{SWAP} = I$.
- *Its square is the identity matrix:* $\mathrm{SWAP}^2 = I$.

A.11 Measurements in Quantum Computing

Measurements play a crucial role in quantum computing as they extract information from quantum states, collapsing them into classical outcomes. This section presents a comprehensive and detailed overview of measurements in quantum computing, including the postulates of quantum mechanics, types of measurements, and the measurement process.

A.11.1 Postulates of Quantum Mechanics

Quantum mechanics is governed by a set of postulates that describe the behavior of quantum systems. The following postulates are relevant to measurements in quantum computing:

1. *Quantum states are represented by vectors in a complex vector space called the Hilbert space. For a qubit, the Hilbert space is a two-dimensional complex vector space, and its state can be represented as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$.*
2. *Observables are represented by Hermitian operators acting on the Hilbert space. An observable $\hat{A}$ has a set of eigenvectors $\{|a_i\rangle\}$ and eigenvalues $\{a_i\}$, satisfying $\hat{A}|a_i\rangle = a_i|a_i\rangle$.*

3. *The outcome of a measurement is one of the eigenvalues of the observable being measured. The probability of obtaining a particular eigenvalue a_i when measuring the state $|\psi\rangle$ is given by $p(a_i) = |\langle a_i|\psi\rangle|^2$.*

4. *After a measurement yielding the outcome a_i, the quantum state collapses to the corresponding eigenvector $|a_i\rangle$.*

A.11.2 Types of Measurements

In quantum computing, the most common type of measurement is the projective measurement, also known as the von Neumann measurement. This type of measurement is based on the eigenvalues and eigenvectors of the observable being measured. For qubits, the most common observables are the Pauli matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{A.102}$$

Other types of measurements include generalized measurements, such as positive operator-valued measures (POVMs), which allow for a more general description of the measurement process, including the effects of noise and decoherence.

A.11.3 Projective Measurements

Projective measurements are a specific type of measurement in quantum computing that project the measured quantum state onto one of the eigenvectors of the measurement operator. The outcome of a projective measurement is a classical bit of information, and the post-measurement state is one of the eigenvectors corresponding to the obtained classical outcome.

Mathematical Representation

A projective measurement is represented by a set of projectors $\{P_i\}$, where each projector P_i corresponds to a possible outcome i of the measurement. A projector is an idempotent, Hermitian operator, satisfying the following conditions:

$$P_i^\dagger = P_i \tag{A.103}$$
$$P_i^2 = P_i \tag{A.104}$$

The projectors must also be orthogonal and sum up to the identity operator:

$$P_i P_j = \delta_{ij} P_i \tag{A.105}$$
$$\sum_i P_i = I \tag{A.106}$$

where δ_{ij} is the Kronecker delta, and I is the identity operator.

Measurement Outcomes and Probabilities

When a quantum state $|\psi\rangle$ is measured using a set of projectors $\{P_i\}$, the probability of obtaining the outcome i is given by:

$$p(i) = \langle\psi|P_i|\psi\rangle \tag{A.107}$$

After the measurement, the quantum state collapses to the eigenvector corresponding to the outcome i. The post-measurement state $|\psi_i\rangle$ is given by:

$$|\psi_i\rangle = \frac{P_i|\psi\rangle}{\sqrt{\langle\psi|P_i|\psi\rangle}} \tag{A.108}$$

Example: Measurement in the Computational Basis

A common projective measurement in quantum computing is the measurement in the computational basis, which uses the standard basis vectors $|0\rangle$ and $|1\rangle$. The projectors for this measurement are:

$$P_0 = |0\rangle\langle 0| \tag{A.109}$$
$$P_1 = |1\rangle\langle 1| \tag{A.110}$$

Given a quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the probabilities of obtaining the outcomes 0 and 1 are:

$$p(0) = |\alpha|^2 \tag{A.111}$$
$$p(1) = |\beta|^2 \tag{A.112}$$

The post-measurement states for the outcomes 0 and 1 are:

$$|\psi_0\rangle = \frac{|0\rangle\langle 0|(\alpha|0\rangle + \beta|1\rangle)}{\sqrt{|\alpha|^2}} = \frac{\alpha|0\rangle}{\sqrt{|\alpha|^2}} = |0\rangle \;\; |\psi_1\rangle \quad = \frac{|1\rangle\langle 1|(\alpha|0\rangle + \beta|1\rangle)}{\sqrt{|\beta|^2}} = \frac{\beta|1\rangle}{\sqrt{|\beta|^2}} = |1\rangle \tag{A.113}$$

Properties of Projective Measurements

Projective measurements have several important properties that are essential for quantum computing:

- ***Irreversibility:*** *Projective measurements are inherently irreversible, meaning that once a quantum state has been measured, it is not possible to recover the original state. This property is a consequence of the projection postulate and the probabilistic nature of quantum mechanics.*
- ***Non-destructive Measurements:*** *Projective measurements can be non-destructive if the measured quantum state is already an eigenvector of the measurement operator. In this case, the state remains unchanged after the measurement. However, if the state is not an eigenvector, the measurement will collapse the state to one of the eigenvectors, destroying the original state in the process.*

- ***Measurement-induced Entanglement:*** *Projective measurements can induce entanglement between two or more quantum systems. This property is useful for preparing entangled states and implementing quantum algorithms that rely on entanglement.*
- ***No-cloning Theorem:*** *Due to the irreversibility of projective measurements, it is not possible to create a perfect copy of an unknown quantum state. This property, known as the no-cloning theorem, is a fundamental constraint in quantum information theory and has important implications for quantum cryptography and quantum error correction.*

A.11.4 POVM Measurements

This section provides a detailed description of POVM (Positive Operator-Valued Measure) measurements in quantum computing. Unlike projective measurements, which are based on the eigenstates of a Hermitian operator, POVMs are a more general way to describe measurements in quantum mechanics. They can be applied to scenarios where the measurement process is not ideal or where the measurement outcomes are not orthogonal. This description covers the fundamentals of POVM measurements, their mathematical representation, and their properties. Unlike projective measurements, POVM measurements can describe non-orthogonal measurement outcomes and non-unitary measurement processes.

Mathematical Representation

A POVM measurement is represented by a set of positive semi-definite (defined below) operators $\{E_i\}$, called POVM elements, which act on the quantum state space. These elements must satisfy the following conditions:

$$E_i \geq 0 \tag{A.114}$$

$$\sum_i E_i = I \tag{A.115}$$

where the first condition (positive semi-definiteness) can be restated in the form $\langle x|E_i|x\rangle \geq 0$ for any vector $|x\rangle$, and I is the identity operator.

Measurement Outcomes and Probabilities

When a quantum state $|\psi\rangle$ is measured using a POVM $\{E_i\}$, the probability of obtaining the outcome i is given by:

$$p(i) = \langle\psi|E_i|\psi\rangle \tag{A.116}$$

The post-measurement state $|\psi_i\rangle$ can be obtained by applying an appropriate quantum operation, which may be different for each outcome. However, unlike projective measurements, the post-measurement state is not uniquely determined by the POVM elements alone.

Properties of POVM Measurements

POVM measurements have several important properties that make them useful in quantum computing:

- ***Generality:*** *POVM measurements are more general than projective measurements, as they can describe non-orthogonal measurement outcomes and non-unitary measurement processes. This makes them suitable for a wide range of scenarios, including open quantum systems, quantum error correction, and quantum cryptography.*
- ***Optimality:*** *In some situations, POVM measurements can provide optimal discrimination between non-orthogonal quantum states. This property is important for various quantum information processing tasks, such as quantum state discrimination, quantum cloning, and quantum communication.*
- ***Physical Realizability:*** *POVM measurements can be realized using a combination of unitary operations, ancillary quantum systems, and projective measurements. This makes them physically realizable in practice, which is essential for implementing quantum algorithms and protocols that rely on generalized measurements.*
- ***Connection to Projective Measurements:*** *Every projective measurement can be represented as a POVM measurement, making POVMs a natural generalization of projective measurements. In particular, a projective measurement can be described by a POVM with elements $E_i = P_i$, where P_i are the projectors corresponding to the measurement operator's eigenvectors.*

A.11.5 Entangled States and Measurements

When measuring entangled states, the outcomes of the measurements on the individual qubits are correlated. For example, consider the Bell state:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{A.117}$$

where we have used the tensor-product notation for,

$$|00\rangle := |0\rangle \otimes |0\rangle \tag{A.118}$$

$$|11\rangle := |1\rangle \otimes |1\rangle \tag{A.119}$$

When measuring the σ_z observable on both qubits, the possible outcomes are:

- *Both qubits yield $\lambda_0 = 1$ with probability $|\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$, resulting in the state $|00\rangle$.*
- *Both qubits yield $\lambda_1 = -1$ with probability $|\frac{1}{\sqrt{2}}|^2 = \frac{1}{2}$, resulting in the state $|11\rangle$.*

Notice that the outcomes are perfectly correlated, i.e., if one qubit yields λ_0, the other qubit will also yield λ_0, and if one qubit yields λ_1, the other qubit will also yield λ_1. This correlation is a result of the entanglement between the qubits.