Introduction to Quantum Computation

Sevag Gharibian

July 2021

Department of Computer Science
Paderborn University
Germany

Contents

1 Introduction and Linear Algebra Review

I recall that during one walk Einstein suddenly stopped, turned to me and asked whether I really believed that the moon exists only when I look at it.
— Abraham Pais.

1.1 Introduction

Welcome to Introduction to Quantum Computation! In this course, we shall explore the subject of quantum computation from a theoretical computer science perspective. As the quote by Abraham Pais above foreshadows, our story will involve surprising twists and turns, which will seem completely at odds with your perception of the world around you. Indeed, in a quantum world, a single particle can be in two places simultaneously; two particles can be so strongly "bound" that they can *appear* to communicate instantaneously even if they are light-years apart; and the very act of "looking" at a quantum system can irreversibly alter the system itself! It is precisely these quirks of quantum mechanics which we shall aim to exploit in our study of computation.

The basic premise of quantum computing is "simple": To build a computer whose bits are not represented by transistors, but by subatomic particles such as electrons or photons. In this subatomic world, the pertinent laws of physics are no longer Newton's classical mechanics, but rather the laws of *quantum mechanics*. Hence, the name "quantum computing". Why would we ever want to build such a computer? There are a number of reasons. From an engineering standpoint, microchip components have become so small that they encounter quantum effects which impede their functionality. To a physicist, the study of quantum computing is a natural approach for simulating and studying quantum systems in nature. And to a computer scientist, quantum computers are remarkable in that they can solve problems which are believed to be intractable on a classical computer!

The field of quantum computing, which arguably started with famed physicist Richard Feynman's ideas (1982) for efficiently simulating physical systems (although it should be noted that ideas for crytography based on quantum mechanics date back to Stephen Wiesner around 1970), is nowadays far too large to be captured in a single course. Here, we shall focus on a broad introduction which aims to cover topics such as: What is quantum mechanics, and how can it be harnessed to build a computer? What kind of computational problems can such a computer solve? Are there problems which are hard even for a quantum computer? And finally, what does the study of quantum computing tell us about nature itself? Even if this course is the last time you encounter the topic of quantum computing, the experience should hopefully leave you with an appreciation for the fine interplay between the limits of physics and computing, as well as strengthen your background in Linear Algebra, which is useful in many other areas of computer science.

The entire course will take place in the mathematical framework of Linear Algebra, which we now review. It is crucial that you familiarize yourself with these concepts before proceeding with the course. These notes contain many exercises intended to help the reader; it is strongly recommended for you to work on these as you read along.

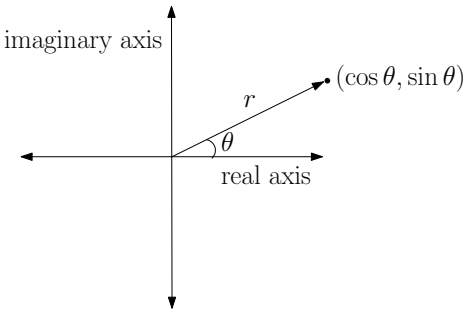
1.2 Linear Algebra

This course assumes a basic background in Linear Algebra. Thus, much of what is covered in this section is intended to be a refresher (although some of the later concepts here may be new to you); we thus cover this section briskly. Throughout this course, the symbols $\mathbb{C}$, $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$ denote the sets of complex, real, integer, and natural numbers, respectively. For m a positive integer, the notation $[m]$ indicates the set $\{1, \ldots, m\}$.

The basic objects we shall work with are complex vectors $|\psi\rangle \in \mathbb{C}^d$, i.e.

$$|\psi\rangle = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_d \end{pmatrix}, \tag{1.1}$$

for $\psi_i \in \mathbb{C}$. Recall here that a complex number $c \in \mathbb{C}$ can be written in two equivalent ways: Either as $c = a + bi$ for $a, b \in \mathbb{R}$ and $i^2 = -1$, or in its *polar form* as $c = re^{i\theta}$ for $r, \theta \in \mathbb{R}$. One of the advantages of the polar form is that it can directly be visualized on the 2D complex plane:



Here, the x and y axes correspond to *real* and *imaginary* axes, and r denotes the *length* of the vector $(\cos\theta, \sin\theta)$. For example, observe that 1 can be written in polar form with $r = 1$ and $\theta = 0$, i.e. 1 is represented in the 2D plane as vector $(1, 0)$. In this course, the polar form will be used repeatedly. Recall also that the complex conjugate of c, denoted c^*, is defined as $a - bi$ or $re^{-i\theta}$, respectively. Finally, the notation $|\cdot\rangle$ is called *Dirac* notation, named after physicist Paul Dirac, and is simply a useful convention for referring to column vectors. The term $|\psi\rangle$ is read "ket ψ".

Exercise 1.1. The *magnitude* or "length" of $c \in \mathbb{C}$ is given by $|c| = \sqrt{cc^*}$. What is the magnitude of $e^{i\theta}$ for any $\theta \in \mathbb{R}$? How about the magnitude of $re^{i\theta}$?

Complex vectors shall be crucial to us for a single reason: They represent quantum states (more details in subsequent lectures). It is thus important to establish some further basic properties of vectors. First, the *conjugate transpose* of $|\psi\rangle$ is given by

$$\langle\psi| = (\psi_1^*, \psi_2^*, \ldots, \psi_d^*), \tag{1.2}$$

where $\langle\psi|$ is a *row* vector. The term $\langle\psi|$ is pronounced "bra ψ". This allows us to define how much two vectors "overlap" via the *inner product* function, defined as

$$\langle\psi|\phi\rangle = \sum_{i=1}^{d} \psi_i^* \phi_i. \tag{1.3}$$

The inner product satisfies $(\langle\psi|\phi\rangle)^* = \langle\phi|\psi\rangle$. The "length" of a vector $|\psi\rangle$ can now be quantified by measuring the overlap of $|\psi\rangle$ with itself, which yields the *Euclidean norm*, $\| |\psi\rangle \|_2 = \sqrt{\langle\psi|\psi\rangle}$.

Exercise 1.2. Let $|\psi\rangle = \frac{1}{\sqrt{2}}(1,i)^T \in \mathbb{C}^2$, where T denotes the transpose. What is $\langle\psi|$? How about $\| |\psi\rangle \|_2$?

With a norm in hand, we can define a notion of *distance* between vectors $|\psi\rangle, |\phi\rangle$, called the Euclidean distance: $\| |\psi\rangle - |\phi\rangle \|_2$. This distance will play the important role of quantifying how well two quantum states $|\psi\rangle$ and $|\phi\rangle$ can be "distinguished" via measurements. Two useful properties of the Euclidean norm are:

1. (Positive scalability) $\| a|\psi\rangle \|_2 = |a| \, \| |\psi\rangle \|_2$ for $a \in \mathbb{C}$.
2. (Triangle inequality) For any $|\psi\rangle, |\phi\rangle$, one has $\| |\psi\rangle + |\phi\rangle \|_2 \leq \| |\psi\rangle \|_2 + \| |\phi\rangle \|_2$.

These two properties can be used to show that for all $|\psi\rangle \in \mathbb{C}^d$, $\| |\psi\rangle \|_2 \geq 0$.

Exercise 1.3. Let $|\psi\rangle = \frac{1}{\sqrt{2}}(1,i)^T \in \mathbb{C}^2$ and $|\phi\rangle = (\frac{1}{2}, \frac{\sqrt{3}}{2})^T \in \mathbb{C}^2$, where T denotes the transpose. What is $\| |\psi\rangle - |\phi\rangle \|_2$?

Orthonormal bases. A much more natural way to represent vectors in this course shall be in terms of *orthonormal bases*. Recall that a set of vectors $\{|\psi\rangle_i\} \subseteq \mathbb{C}^d$ is *orthogonal* if for all $i \neq j$, $\langle\psi|_i|\psi\rangle_j = 0$, and *orthonormal* if $\langle\psi|_i|\psi\rangle_j = \delta_{ij}$. Here, δ_{ij} is the Kroenecker delta, whose value is 1 if $i = j$ and 0 otherwise. For the vector space $\mathbb{C}^d$, which has dimension d, it is necessary and sufficient to use d orthonormal vectors in order to form an orthonormal basis.

One of the most common bases we use is the *computational basis*, defined for $\mathbb{C}^2$ as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad\qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{1.4}$$

Since $\{|0\rangle, |1\rangle\}$ is an orthonormal basis, any vector $|\psi\rangle \in \mathbb{C}^2$ can be written as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ for some $\alpha, \beta \in \mathbb{C}$. We say $|\psi\rangle$ is *normalized* when it has length 1, i.e. $\| |\psi\rangle \|_2 = 1$; equivalently, this means $|\alpha|^2 + |\beta|^2 = 1$.

Exercise 1.4. Let $|\psi\rangle = \frac{1}{\sqrt{2}}(1,1)^T \in \mathbb{C}^2$. Write $|\psi\rangle$ in terms of the computational basis for $\mathbb{C}^2$. Is $|\psi\rangle$ normalized?

The computational basis is easily extended to d-dimensional vectors by defining $|i\rangle \in \mathbb{C}^d$ as having a 1 in position i and 0 elsewhere (here, $0 \leq i \leq d-1$). In this course, vectors labelled by integers (e.g. $|1\rangle$, $|3\rangle \in \mathbb{C}^d$) will be assumed to be d-dimensional computational basis vectors.

Linear maps. Given a vector $|\psi\rangle \in \mathbb{C}^d$, we are interested in how $|\psi\rangle$ can be "mapped" to other vectors. The maps we consider are *linear*, which by definition means that for map $\Phi : \mathbb{C}^d \mapsto \mathbb{C}^d$ and arbitrary $\sum_i \alpha_i|\psi_i\rangle \in \mathbb{C}^d$,

$$\Phi\left(\sum_i \alpha_i|\psi_i\rangle\right) = \sum_i \alpha_i \Phi(|\psi_i\rangle). \tag{1.5}$$

The set of linear maps from vector space $\mathcal{X}$ to $\mathcal{Y}$ is denoted $\mathcal{L}(\mathcal{X}, \mathcal{Y})$. For brevity, we use the shorthand $\mathcal{L}(\mathcal{X})$ to mean $\mathcal{L}(\mathcal{X}, \mathcal{X})$.

Exercise 1.5. Consider the linear map $\Phi : \mathbb{C}^2 \mapsto \mathbb{C}^2$ with action $\Phi(|0\rangle) = |1\rangle$ and $\Phi(|1\rangle) = |0\rangle$. If $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, what is $\Phi(|\psi\rangle)$?

The exercise above teaches us an important lesson — the action of a linear map $\Phi \in \mathcal{L}(\mathbb{C}^d)$ is fully characterized by understanding how Φ acts on a basis for $\mathbb{C}^d$. This leads to a natural representation for Φ in terms of a *matrix*.

Recall that a $d \times d$ *matrix* A is a two-dimensional array of complex numbers whose (i, j)th entry is denoted $A(i, j) \in \mathbb{C}$ for $i, j \in [d]$. To represent a linear map $\Phi : \mathbb{C}^d \mapsto \mathbb{C}^d$ as an $d \times d$ matrix A_Φ, we use its action on a basis for $\mathbb{C}^d$. Specifically, define the ith column of A_Φ as $\Phi(|i\rangle)$ for $\{|i\rangle\}$ the standard basis for $\mathbb{C}^d$, or

$$A_\Phi = \left[\; \Phi(|0\rangle), \Phi(|1\rangle), \ldots, \Phi(|d-1\rangle) \; \right]. \tag{1.6}$$

In this course, we use both the matrix and linear map views interchangeably, with the application notion clear from context.

Exercise 1.6. What is the 2×2 complex matrix representing the linear map Φ from the previous exercise? What is the linear map whose matrix (with respect to the computational basis) is the *identity matrix*

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}? \tag{1.7}$$

The product AB of two $d \times d$ matrices A and B is also a $d \times d$ matrix with entries

$$AB(i, j) = \sum_{k=1}^{d} A(i, k)B(k, j). \tag{1.8}$$

Note that unlike for scalars, for matrices it is *not* always true that $AB = BA$. In the special case where $AB = BA$, we say A and B *commute*.

Exercise 1.7. Define

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \text{and} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.9}$$

Do X and Z commute?

We would like to understand how "large" the output space of a linear map $A \in \mathcal{L}(\mathbb{C}^d)$ is. To this end, the *image* of A is the set of all possible output vectors under the action of A, i.e.

$$\mathrm{Im}(A) := \Big\{ |\psi\rangle \in \mathbb{C}^d \mid |\psi\rangle = A|\phi\rangle \text{ for some } |\phi\rangle \in \mathbb{C}^d \Big\}. \tag{1.10}$$

The *rank* of A is the dimension of its image.

Exercise 1.8. Suppose $\mathrm{rank}(A) = 0$. What is $\mathrm{Im}(A)$? How about the case of $\mathrm{rank}(A) = d$?

The set of all vectors sent to zero by A is called its *null space*, i.e. $\mathrm{Null}(A) := \big\{ |\psi\rangle \in \mathbb{C}^d \mid A|\psi\rangle = 0 \big\}$. It holds that $\dim(\mathrm{Null}(A)) + \dim(\mathrm{Im}(A)) = d$ (here dim denotes dimension).

Exercise 1.9. Is there a non-zero vector in the null space of matrix Z from Equation (1.9)? (Hint: Multiply an arbitrary vector $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ by Z and see if you can make the zero vector pop out.) What does the answer tell you about rank(Z)? What is the null space of matrix

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \tag{1.11}$$

and what is rank(B)?

Matrix operations. Matrices encode the operations which we are allowed to perform on our vectors. There are some simple operations on matrices *themselves* which will show up repeatedly in our discussions. The first three of these are the linear maps *complex conjugate*, *transpose* and *adjoint*, defined respectively as

$$A^*(i,j) = (A(i,j))^* \qquad A^T(i,j) = A(j,i) \qquad A^\dagger = (A^*)^T. \tag{1.12}$$

Note that $(AB)^\dagger = B^\dagger A^\dagger$, and similarly for the transpose. These operations apply to vectors as well so that $\langle\psi|$, defined in Equation (1.2), is simply $|\psi\rangle^\dagger$. The adjoint will especially play a crucial role in this course.

Exercise 1.10. Calculate $X^\dagger$ and $Z^\dagger$ for X and Z from Equation (1.9), as well as the adjoint of

$$A = \begin{pmatrix} 1 & 2 \\ 3 & e^{i\pi/2} \end{pmatrix}. \tag{1.13}$$

Another useful function on matrices is the *trace*, which is simply a linear map $\mathrm{Tr} : \mathcal{L}(\mathbb{C}^d) \mapsto \mathbb{C}$ summing the entries on the diagonal of A, i.e. $\mathrm{Tr}(A) = \sum_{i=1}^d A(i,i)$. A wonderful property of the trace is that it is *cyclic*, i.e. $\mathrm{Tr}(ABC) = \mathrm{Tr}(CAB)$. This implies that even if A and B do not commute, i.e. $AB \neq BA$, it nevertheless holds that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$!

Exercise 1.11. In a previous exercise, you showed that X and Z do not commute. Compute $\mathrm{Tr}(XZ)$ and $\mathrm{Tr}(ZX)$ and verify that they are indeed the same.

Outer products. The Dirac notation lends itself particularly well to an alternate description of matrices via *outer products*. For vectors $|\psi\rangle, |\phi\rangle \in \mathbb{C}^d$, the outer product is $|\psi\rangle\langle\phi| \in \mathcal{L}(\mathbb{C}^d)$; unlike the inner product, the outer product yields a $d \times d$ matrix. It can be computed straightforwardly via the rules of matrix multiplication. For example,

$$|0\rangle\langle 0| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad |1\rangle\langle 0| = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}. \tag{1.14}$$

More generally, the matrix $|i\rangle\langle j| \in \mathcal{L}(\mathbb{C}^d)$ has a 1 at position (i,j) and zeroes elsewhere. This yields a simple yet neat trick: A matrix $A \in \mathcal{L}(\mathbb{C}^d)$ written in the computational basis can hence be expressed as $\sum_{ij} A(i,j)|i\rangle\langle j|$. It is thus easy to evaluate expressions of the form

$$\langle i|A|j\rangle = \langle i| \left(\sum_{i'j'} A(i',j')|i'\rangle\langle j'| \right) |j\rangle = \sum_{i'j'} A(i',j')\langle i|i'\rangle\langle j|j'\rangle = \sum_{i'j'} A(i',j')\delta_{ii'}\delta_{jj'} = A(i,j), \tag{1.15}$$

where the third equality follows since $\{|i\rangle\}$ forms an orthonormal basis for $\mathbb{C}^d$. In other words, $\langle i|A|j\rangle$ simply rips out entry $A(i,j)$! These types of expressions will be ubiquitous in the setting of quantum measurements.

Exercise 1.12. Observe that X from Equation 1.9 can be written $X = |0\rangle\langle 1| + |1\rangle\langle 0|$. What is $\langle 0|X|0\rangle$? How about $\langle 0|X|1\rangle$? How can you rewrite $\mathrm{Tr}(X)$ in terms of expressions of this form?

Eigenvalues and eigenvectors. With outer products in hand, we can discuss one of the most fundamental tools in our Linear Algebraic toolkit — eigenvalues and eigenvectors. Given any matrix $A \in \mathcal{L}(\mathbb{C}^d)$, an *eigenvector* is a special non-zero vector satisfying the equation

$$A|\psi\rangle = \lambda|\psi\rangle, \tag{1.16}$$

for some $\lambda \in \mathbb{C}$ which is the corresponding *eigenvalue*.

Exercise 1.13. Show that $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ are both eigenvectors of X from Equation (1.9). What are their respective eigenvalues?

The outer product can now be used to state an expression which will be used repeatedly in this course. For any matrix A satisfying $AA^\dagger = A^\dagger A$ (such matrices are called *normal*; most matrices in this course will be normal), we can decompose A in terms of its eigenvalues and eigenvectors as

$$A = \sum_{i=1}^{d} \lambda_i |\lambda_i\rangle\langle\lambda_i|, \tag{1.17}$$

where λ_i and $|\lambda_i\rangle$ are the eigenvalues and corresponding eigenvectors of A. This is called the *spectral decomposition* of A. The spectral decomposition is useful for a few reasons. First, it tells us exactly how A acts on $\mathbb{C}^d$; this is because the eigenvectors $|\lambda_i\rangle \in \mathbb{C}^d$ can be chosen[1] to form an orthonormal basis for $\mathbb{C}^d$. Thus, any vector $|\psi\rangle \in \mathbb{C}^d$ can be written in terms of the eigenvectors of A, i.e. $|\psi\rangle = \sum_i \alpha_i |\lambda_i\rangle$ for some $\alpha_i \in \mathbb{C}$. The spectral decomposition also immediately reveals the rank of A; specifically, $\mathrm{rank}(A)$ is just the number of non-zero eigenvalues of A. Finally, $\mathrm{Tr}(A)$ has a very simple expression in terms of A's eigenvalues — $\mathrm{Tr}(A) = \sum_i \lambda_i$. Let us quickly prove this last claim:

$$\mathrm{Tr}(A) = \mathrm{Tr}\left(\sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i|\right) = \sum_i \lambda_i \mathrm{Tr}(|\lambda_i\rangle\langle\lambda_i|) = \sum_i \lambda_i \mathrm{Tr}(\langle\lambda_i|\lambda_i\rangle) = \sum_i \lambda_i. \tag{1.18}$$

Here, the second equality follows since the trace is linear, the third by the cyclic property of the trace, and the last since the eigenvectors $|\lambda_i\rangle$ are orthonormal.

Exercise 1.14. In the previous exercise, you computed the eigenvectors and eigenvalues of X. Use these to write down the spectral decomposition of X, and verify that it indeed evaluates to X. Next, recall that $X|0\rangle = |1\rangle$. Note that $|0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$. Use this and the spectral decomposition of X to verify that indeed $X|0\rangle = |1\rangle$.

Finally, recall that the eigenvalues of $A \in \mathcal{L}(\mathbb{C}^d)$ can be computed as the roots of the degree-d *characteristic polynomial* of A, p_A, defined

$$p_A(\lambda) = \det(\lambda I - A), \tag{1.19}$$

[1] This statement need not hold for non-normal matrices. In fact, one can prove that a matrix is normal if and only if it admits a spectral decomposition. Non-normal matrices do, however, admit the more general *singular value decomposition*.

where the determinant det can be defined recursively as

$$\det(A) = \sum_{j=1}^{d} (-1)^{i+j} A(i,j) \det(A_{ij}). \tag{1.20}$$

Here, A_{ij} is the matrix obtained from A by deleting row i and column j, and we define the base case of this recursion (i.e. a 1×1 matrix $[c]$) as $\det([c]) = c$. This equation holds for any $i \in [d]$. In the special case when $d = 2$, this reduces nicely to

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc. \tag{1.21}$$

Exercise 1.15. Use Equations (1.19) and (1.21) to compute the eigenvalues of Z from Equation (1.9). Then, plug these back into Equation (1.16) to solve for the eigenvectors of Z. What is the spectral decomposition of Z?

Let us close with a simple observation: For any diagonal matrix A (written in the computational basis), the eigenvalues of A are simply the entries on the diagonal of A, and the eigenvectors are just the computational basis vectors. In the exercise above, this immediately confirms that the eigenvalues of Z are 1 and -1 with eigenvectors $|0\rangle$ and $|1\rangle$, respectively.

2 Introduction to Quantum Mechanics

...the "paradox" is only a conflict between reality and your feeling of what reality "ought to be."
— Richard Feynman.

2.1 The four postulates of quantum mechanics

In this course, our aim is to study computing devices which operate according to the laws of *quantum mechanics*. Developed during the early 20th century by physicists Max Planck, Albert Einstein, Erwin Schrödinger and many others, quantum mechanics is a set of mathematical laws which describe the behaviour of subatomic particles such as protons, electrons, and photons. Although the theory has proven remarkably successful since its inception, it is nevertheless notoriously counterintuitive, an aspect which we shall explore in this lecture.

Quantum mechanics is based on four postulates, which describe the following four intuitive ideas: How to describe a single quantum system, how to perform quantum operations on a quantum system, how to describe multiple quantum systems, and how to measure or extract classical information from a quantum system. In this lecture, we explore the first three of these postulates. The fourth postulate is discussed in the following lecture.

2.1.1 Postulate 1: Individual quantum systems

Recall that in the classical world, a bit x can take on one of two values: 0 or 1. In the quantum world, we immediately see a radical departure from this statement — a quantum bit, or *qubit*, can take on not just 0 *or* 1, but rather *both* values 0 and 1 simultaneously. This is a very deep and counterintuitive statement, so it worth reflecting on — it is like saying you can be both asleep and awake at the same time, or here on Earth and simultaneously on Mars at the same time. Indeed, relative to life as we know it, *it makes no sense*!

Let us formalize this phenomenon. We begin by encoding bits 0 and 1 via the standard basis vectors $|0\rangle, |1\rangle \in \mathbb{C}^2$. Then, to denote that a qubit is in states $|0\rangle$ and $|1\rangle$ simultaneously, we write

$$|0\rangle + |1\rangle.$$

This is called a *superposition*. More generally, we can change the "extent" to which the qubit is in state $|0\rangle$ versus $|1\rangle$ via *amplitudes* $\alpha, \beta \in \mathbb{C}$, i.e.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle.$$

The only restriction is that $|\psi\rangle$ must be a unit vector, i.e. that $|\alpha|^2 + |\beta|^2 = 1$. To summarize, any unit vector in $\mathbb{C}^2$ describes the state of a single qubit.

Exercise 2.1. Among the most commonly used single qubit states are $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Verify that these are indeed unit vectors.

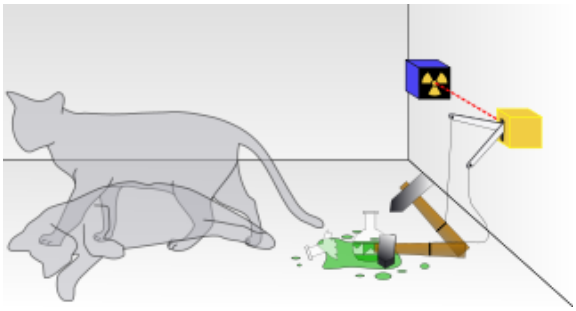


Figure 2.1: A depiction of Schrödinger's cat.

Aside: Schrödinger's cat. To demonstrate how strange the concept of quantum superposition is, in 1935 Austrian physicist Erwin Schrödinger devised a thought experiment, nowadays infamously referred to as *Schrödinger's cat*. The experiment, depicted in[1] Figure 2.1, goes as follows (we give a slight variant suitable to our exposition of quantum mechanics here): Suppose that we place a cat in a box and *close* the box (i.e. one cannot look inside the box). In the box, we place a flask of poison, along with a hammer. The hammer is connected to a mechanism outside the box, which is controlled by a computer. If the computer is fed input 0, then nothing happens, and the cat is happy doing whatever it is doing in the box. On the other hand, if the input is 1, then the hammer falls and breaks the flask, releases the poison, and kills the cat.

And now Schrödinger asked the key question: *What if we input a* superposition *of* 0 *and* 1 *to the computer, i.e. the state* $|0\rangle + |1\rangle$*?* If we interpret quantum mechanics literally, then we conclude that the cat is both alive and dead, *at the same time*! Of course, this makes absolutely no sense. Moreover, common sense tells us that if you simply *open* the box and look inside, we will find either a cat which is alive or dead, not both. How can this paradox be resolved? Read on to Postulate 4 to find out!

Finally, thus far we have described the state of a single (2-dimensional) qubit. More generally, the state of a d-dimensional quantum system, called a qu$d$it, is described by a unit vector $|\psi\rangle \in \mathbb{C}^d$, which can be described as

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \cdots + \alpha_{d-1}|d-1\rangle = \sum_{i=0}^{d-1} \alpha_i|i\rangle,$$

where recall $|i\rangle \in \mathbb{C}^d$ denotes the ith computational basis vector and $\alpha_i \in \mathbb{C}$. Since $|\psi\rangle$ is a unit vector, we have $\sum_{i=0}^{d-1} |\alpha_i|^2 = 1$.

2.1.2 Postulate 2: Quantum operations

We next ask: What types of operations or maps can we perform on a qubit? Since a qubit is a vector, the natural choice is a linear map, i.e. multiplication by a matrix. However, not all matrices are fair game — it turns out that nature only allows a special class of matrices known as *unitary* matrices. A unitary matrix $U \in \mathcal{L}(\mathbb{C}^d)$ is one which satisfies $UU^\dagger = U^\dagger U = I$.

[1] Figure due to user Dhatfield, obtained from `https://commons.wikimedia.org/wiki/File:Schrodingers_cat.svg`.

In other words, the *inverse* of U is simple to calculate — just take the dagger of U. This immediately yields a key insight — all quantum gates are *reversible*.

Among the most common single qubit gates are the following, known as the *Pauli* gates, after Austrian-Swiss physicist Wolfgang Pauli:

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

Exercise 2.2. Verify that Pauli X, Y, and Z are unitary.

The X gate acts as a "quantum" NOT gate, as we see below:

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \qquad \text{and} \qquad X|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle.$$

It follows that $|+\rangle$ and $|-\rangle$ are eigenvectors of X, i.e. $X|+\rangle = |+\rangle$ and $X|-\rangle = -|-\rangle$ (as we calculated in the last lecture). The spectral decomposition of X is hence $X = |+\rangle\langle+| - |-\rangle\langle-|$.

Exercise 2.3. Write $|+\rangle\langle+| - |-\rangle\langle-|$ out as a matrix to verify that it indeed equals X.

The Z gate, on the other hand, has no classical analogue. It acts as

$$Z|0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \qquad \text{and} \qquad Z|1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -\begin{pmatrix} 0 \\ 1 \end{pmatrix} = -|1\rangle.$$

In other words, Z leaves $|0\rangle$ invariant, but injects a "phase" of -1 in front of $|1\rangle$. This also immediately shows that $|0\rangle$ and $|1\rangle$ are eigenvectors of Z with eigenvalues 1 and -1, respectively.

Exercise 2.4. Write down the spectral decomposition of Z.

The Z gate is special in that it allows us to inject a *relative phase* into a quantum state. For example,

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

By relative phase, we mean that only the amplitude on $|1\rangle$ had its sign changed (or more generally, was multiplied by a phase $e^{i\pi} = -1$). If *all* the amplitudes in the state were instead multiplied by $e^{i\pi}$, then we could simply factor out the $e^{i\pi}$ from the entire state — in this case, we would call $e^{i\pi}$ a *global* phase. It turns out that a global phase is insignificant in that it cannot be experimentally detected. A relative phase may seemingly also look unimportant - yet, as we shall see in this course, it is one of the features of quantum mechanics which allows quantum computers to outperform classical ones!

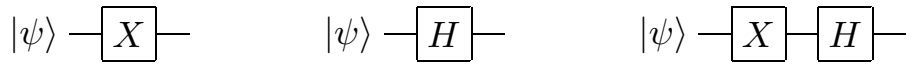
Finally, we come to a fourth important unitary gate, the *Hadamard* gate:

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

The Hadamard gate is special in that it creates superpositions for us. Namely, we have $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$. It can also "erase" superpositions, i.e. $H|+\rangle = |0\rangle$ and $H|-\rangle = |1\rangle$. In other words, H is self-inverse — we have that $H^2 = I$ for I the identity matrix. In fact, the Pauli matrices are also self-inverse.

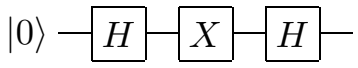
Exercise 2.5. Verify that $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$. Also verify that $H^2 = I$.

It is very useful to graphically depict sequences of quantum gates via *quantum circuits*. For example, here are three circuits:



They correspond to evolutions $X|\psi\rangle$, $H|\psi\rangle$, and $HX|\psi\rangle$, respectively. Each wire in such a diagram denotes a quantum system, and a box labelled by gate U depicts the action of unitary U. We think of time going from left to right; for the last circuit above, note that the X appears on the "left" in the circuit diagram but on the "right" in the expression $HX|\psi\rangle$; this is because X should be applied first to $|\psi\rangle$, then H.

Exercise 2.6. What single-qubit state does the following circuit output? (Hint: Rather than explicitly calculating this, try to use your knowledge of the action of H on states $|0\rangle$ and $|+\rangle$, and the eigenvectors of X.)



2.1.3 Postulate 3: Composite quantum systems

Thus far, we have considered only single quantum systems, i.e. states $|\psi\rangle \in \mathbb{C}^d$ for $d \geq 2$. But a computer with just a single qubit might be rather uninteresting! What we would instead like is to discuss *multiple* qubits simultaneously. How can we mathematically describe, for example, the joint state of two qubits?

The correct Linear Algebraic tool for this task is the *tensor product*, denoted $\otimes$. The tensor product allows us to "stitch together" two vectors, say $|\psi\rangle, |\phi\rangle \in \mathbb{C}^2$, to obtain a larger 4-dimensional vector $|\psi\rangle \otimes |\phi\rangle \in \mathbb{C}^4$. Formally, we have $\mathbb{C}^2 \otimes \mathbb{C}^2 = \mathbb{C}^{2\times 2}$. In other words, the entries of a vector $|\psi\rangle \otimes |\phi\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^2$ can be referenced via a pair of indices (i, j) for $i, j \in \{0, 1\}$, and the specific rule for doing so is

$$(|\psi\rangle \otimes |\phi\rangle)(i, j) := \psi_i \phi_j,$$

where recall ψ_i and ϕ_j are the entries of $|\psi\rangle$ and $|\phi\rangle$, respectively. Here, you should think of the pair (i, j) as representing the bits of a single index $x \in \{0, 1, 2, 3\}$. So for example, $(0, 0)$ is equivalent to index 0, $(0, 1)$ to index 1, and $(1, 1)$ to index 3. This implies that we can think of $|\psi\rangle \otimes |\phi\rangle$ as having four entries, i.e. $|\psi\rangle \otimes |\phi\rangle \in \mathbb{C}^4$. Let us demonstrate with some examples:

$$|0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

Exercise 2.7. Verify that

$$|1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Note that in the four equations above, the four-dimensional vectors obtained are just the computational basis vectors for $\mathbb{C}^4$! This hints at an important fact: If we take orthonormal bases $B_1 = \{|\psi_0\rangle, |\psi_1\rangle\}$ and $B_2 = \{|\phi_0\rangle, |\phi_1\rangle\}$ for $\mathbb{C}^2$, then we can obtain an orthonormal basis for $\mathbb{C}^4$ by tensoring together the elements of B_1 and B_2 in all four possible combinations, i.e. $\{|\psi_0\rangle \otimes |\phi_0\rangle, |\psi_0\rangle \otimes |\phi_1\rangle, |\psi_1\rangle \otimes |\phi_0\rangle, |\psi_1\rangle \otimes |\phi_1\rangle\}$ forms an orthonormal basis for $\mathbb{C}^4$. For brevity, we shall often drop the notation $\otimes$ and simply write $|\psi\rangle \otimes |\phi\rangle = |\psi\rangle|\phi\rangle$.

Exercise 2.8. Compute the 4-dimensional vectors corresponding to $|1\rangle \otimes |-\rangle$ and $|+\rangle \otimes |+\rangle$.

Our discussion thus far generalizes straightforwardly to the case of $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$. Specifically, for $|\psi\rangle \in \mathbb{C}^{d_1}$ and $|\phi\rangle \in \mathbb{C}^{d_2}$, we have that $|\psi\rangle \otimes |\phi\rangle \in \mathbb{C}^{d_1 d_2}$. Then, for $i \in \{0, \ldots, d_1 - 1\}$ and $j \in \{0, \ldots, d_2 - 1\}$, we have $(|\psi\rangle \otimes |\phi\rangle)(i, j) := \psi_i \phi_j$. Thus, for example, if we add a third qubit to our existing two qubit system, then we have a state which lives in $\mathbb{C}^4 \otimes \mathbb{C}^2 = \mathbb{C}^8$. In fact, for each qubit we add to our system, the dimension grows by a factor of 2, i.e. it grows exponentially — in general, an n-qubit state will correspond to a vector $|\psi\rangle \in (\mathbb{C})^{2^n}$! It is precisely this exponential growth in complexity which makes it difficult for classical computers to simulate the mechanics of an n-qubit quantum state — indeed, this was the reason why physicist Richard Feynman proposed the concept of a quantum computer in 1982 to begin with!

Finally, the tensor product has the following important properties for any $|a\rangle, |b\rangle \in \mathbb{C}^{d_1}$ and $|c\rangle, |d\rangle \in \mathbb{C}^{d_2}$, which we will use repeatedly:

$$\begin{aligned}
(|a\rangle + |b\rangle) \otimes |c\rangle &= |a\rangle \otimes |c\rangle + |b\rangle \otimes |c\rangle && (2.1)\\
|a\rangle \otimes (|c\rangle + |d\rangle) &= |a\rangle \otimes |c\rangle + |a\rangle \otimes |d\rangle && (2.2)\\
c(|a\rangle \otimes |c\rangle) &= (c|a\rangle) \otimes |c\rangle = |a\rangle \otimes (c|c\rangle) && (2.3)\\
(|a\rangle \otimes |c\rangle)^\dagger &= |a\rangle^\dagger \otimes |c\rangle^\dagger = \langle a| \otimes \langle c| && (2.4)\\
(\langle a| \otimes \langle c|)(|b\rangle \otimes |d\rangle) &= \langle a|b\rangle\langle c|d\rangle. && (2.5)
\end{aligned}$$

Exercise 2.9. What is the inner product of $|0\rangle|1\rangle$ and $|1\rangle|0\rangle$? How about the inner product of $|0\rangle|0\rangle$ and $|+\rangle|-\rangle$?

Quantum entanglement. Now that we know how to stitch together a pair of single qubit states, it turns out we have opened Pandora's box. For we can now talk about the two-qubit state which troubled Einstein to the end of his days — the innocuous-looking *Bell state*:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}.$$

This state demonstrates a quantum phenomenon known as *entanglement* — intuitively, if a pair q_0 and q_1 of qubits are entangled, then they are so "tightly bound" that one cannot accurately

describe the state of q_0 or q_1 alone — only the *joint* state of q_0 and q_1 can be described precisely. In the language of tensor products, this is captured by the following statement: There do not exist $|\psi_1\rangle, |\psi_2\rangle \in \mathbb{C}^2$ such that $|\Phi^+\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$. In 1935, Einstein, Podolsky and Rosen published a famous paper nowadays referred to as the "EPR" paper, in which they argue that quantum mechanics cannot be a complete physical theory because it allows the existence of states such as $|\Phi^+\rangle$. Fast forwarding a number of decades, we now not only believe entanglement is real, but we know that is is a *necessary resource* for quantum computers to outperform classical ones.

We shall later return to the topic of entanglement, but for now let us remark that there are three other such Bell states:

$$\begin{aligned}
|\Phi^-\rangle &= \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \\
|\Psi^+\rangle &= \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \\
|\Psi^-\rangle &= \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}.
\end{aligned}$$

Note that here we have further simplified notation by letting (e.g.) $|0\rangle|0\rangle = |00\rangle$. The four Bell states $\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$ form an orthonormal basis for $\mathbb{C}^4$ known as the *Bell* basis, after Northern Irish physicist John Bell.

Exercise 2.10. Verify that the Bell basis indeed forms an orthonormal basis, i.e. check that the Bell states are pairwise orthogonal unit vectors.

Two-qubit quantum gates. We have seen that two-qubit quantum states are described by unit vectors in $\mathbb{C}^4$. We can thus discuss two-qubit quantum gates, i.e. unitary operators $U \in \mathcal{L}(\mathbb{C}^4)$. There are two types of such gates: The first are simply tensor products of one-qubit gates, such as $X \otimes Z$ or $H \otimes H$. Here, the tensor product is defined analogously for matrices as it is for vectors. (The formal description is cumbersome, but we follow with a helpful illustration to clarify.) For any $A \in \mathcal{L}(\mathbb{C}^{d_1})$, $B \in \mathcal{L}(\mathbb{C}^{d_2})$, $A \otimes B$ is a $d_1 d_2 \times d_1 d_2$ complex matrix whose entries are indexed by $([d_1] \times [d_2], [d_1] \times [d_2])$ (where $[d] = \{0, \ldots, d-1\}$ here), such that

$$(A \otimes B)((i_1, j_1), (i_2, j_2)) := A(i_1, i_2)B(j_1, j_2).$$

To clarify this definition, suppose

$$A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \qquad \text{and} \qquad B = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix}.$$

Then, $A \otimes B$ is given by

$$A \otimes B = \begin{pmatrix} a_1 \cdot \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} & a_2 \cdot \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \\ a_3 \cdot \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} & a_4 \cdot \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \end{pmatrix}.$$

In other words, $A \otimes B$ is obtained by taking four copies of B, each time multiplying by a different scalar entry of A.

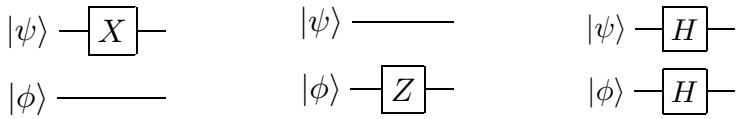
Exercise 2.11. What is $X \otimes I$? How about $Z \otimes H$?

The tensor product for matrices shares the properties of the tensor product for vectors, with the addition of two rules below:

$$(A \otimes B)(C \otimes D) = AC \otimes BD \qquad \text{and} \qquad \mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A)\mathrm{Tr}(B).$$

Exercise 2.12. What is $(Y \otimes Y)(Y \otimes Y)$? How about $\mathrm{Tr}(X \otimes X)$?

The circuit diagrams for tensor products of unitaries are depicted below: We consider the cases of $X \otimes I$, $I \otimes Z$, and $H \otimes H$, respectively.



Exercise 2.13. What is the circuit diagram for $Z \otimes Z$? What is $(X \otimes X)|0\rangle \otimes |1\rangle$? How about $(Z \otimes Z)|1\rangle \otimes |1\rangle$?

Finally, we can also consider genuinely two-qubit gates, i.e. gates which are not the tensor product of single qubit gates. One important such gate is the *controlled-NOT* gate, denoted CNOT. The CNOT treats one qubit as the *control* qubit, and the other as the target *qubit*. It then applies the Pauli X gate to the target qubit only if the control qubit is set to $|1\rangle$. More precisely, the action of the CNOT on a two-qubit basis is given as follows, where qubit 1 is the control and qubit 2 is the target:

$$\mathrm{CNOT}\,|00\rangle = |00\rangle \quad \mathrm{CNOT}\,|01\rangle = |01\rangle \quad \mathrm{CNOT}\,|10\rangle = |11\rangle \quad \mathrm{CNOT}\,|11\rangle = |10\rangle.$$

Exercise 2.14. What is $\mathrm{CNOT}\,|\Phi^+\rangle$ for $|\Phi^+\rangle$ the Bell state? Can you simplify the result to get answer $|+\rangle|0\rangle$?

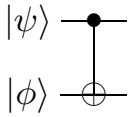
The CNOT gate is given by matrix:

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix},$$

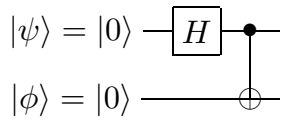
where the second expression is in block matrix form with I and X the identity and X matrices.

Exercise 2.15. Verify that multiplying $|11\rangle$ by the matrix for CNOT indeed yields $|10\rangle$.

The circuit diagram for the CNOT is given by



With this in hand, we can do our first interesting computation — we can prepare the Bell state $|\Phi^+\rangle$ starting from an initial state of $|0\rangle|0\rangle$! The preparation circuit is given as:



To see that this works, note that this diagram is equivalent to

$$\begin{aligned}
\mathrm{CNOT}(H \otimes I)|0\rangle|0\rangle &= \mathrm{CNOT}\,|+\rangle|0\rangle \\
&= \mathrm{CNOT}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)|0\rangle \\
&= \frac{1}{\sqrt{2}}\,\mathrm{CNOT}\,(|00\rangle + |10\rangle) \\
&= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\
&= |\Phi^+\rangle.
\end{aligned}$$

Exercise 2.16. Show that applying the preparation circuit above on initial states $|01\rangle$, $|10\rangle$, and $|11\rangle$ yields the remaining Bell basis states $|\Psi^+\rangle$, $|\Phi^-\rangle$, and $|\Psi^-\rangle$, respectively.

3 Measurement and Quantum Teleportation

"I think that a particle must have a separate reality independent of measurements. That is, an electron has spin, location and so forth even when it is not being measured. I like to think the moon is there even if I am not looking at it."
— Albert Einstein

"... experiments have now shown that what bothered Einstein is not a debatable point but the observed behaviour of the real world."
— N. David Mermin

3.1 Postulate 4: Measurement

We have arrived at the final postulate of quantum mechanics, which asks: How does one mathematically model the act of *measuring* or *observing* a quantum system? For example, suppose we run a quantum algorithm to compute the prime factors of a large composite integer — how can we *measure* the answer register to obtain a classical output string which encodes the prime factors? At this point it may come as no surprise to you that here again quantum mechanics plays an unexpected card — it turns out that the very act of looking at or *observing* a quantum system irreversibly alters the state of the system! This is precisely the phenomenon Einstein's quote above refers to — it's like saying the "state" of the moon is only well-defined the moment you *look* at it.

To model this phenomenon, we shall use the notion of a *projective* or *von Neumann* measurement (named after Hungarian child prodigy and mathematician, John von Neumann, who was apparently already familiar with calculus at the age of 8). To do so, we must define three classes of linear operators, each of which is increasingly more restricted. All three classes will prove vital throughout this course.

Hermitian, positive semi-definite, and orthogonal projection operators.

1. *Hermitian operators*: An operator $M \in \mathcal{L}(\mathbb{C}^d)$ is *Hermitian* if $M = M^\dagger$. Examples you are already familiar with are the Pauli X, Y, and Z gates, which are not only unitary, but also Hermitian. As you have shown in Assignment 1, a Hermitian operator has the important property that all of its eigenvalues are *real*. Thus, Hermitian operators can be thought of as a higher dimensional generalization of the real numbers.

   Exercise 3.1. Verify that Pauli Y is Hermitian. Is the Hadamard gate Hermitian?

2. *Positive semi-definite operators*: If a Hermitian operator has only *non-negative* eigenvalues, then it is called *positive-semidefinite*. Thus, positive semi-definite (or positive for short) matrices generalize the non-negative real numbers.

Exercise 3.2. Prove that Pauli X and Z are not positive semi-definite. (Hint: Recall the spectral decompositions of X and Z.)

3. *Orthogonal projection operators:* A Hermitian matrix $\Pi \in \mathcal{L}(\mathbb{C}^d)$ is an *orthogonal projection operator* (or projector for short) if $\Pi^2 = \Pi$. This is equivalent to saying Π has only eigenvalues 0 and 1. Let us prove this equivalence briefly: Since Π is Hermitian, we can take its spectral decomposition, $\Pi = \sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i|$. Hence,

$$\sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i| = \Pi = \Pi^2 = \left(\sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i|\right)\left(\sum_j \lambda_j |\lambda_j\rangle\langle\lambda_j|\right) = \sum_i \lambda_i^2 |\lambda_i\rangle\langle\lambda_i|,$$

where the last equality follows since $\{|\lambda_i\rangle\}$ is an orthonormal basis. Since the $|\lambda_i\rangle$ are orthogonal, we thus have that for all i, $\lambda_i = \lambda_i^2$. But this can only hold if $\lambda_i \in \{0, 1\}$, as claimed.

Exercise 3.3. Verify that I, $|0\rangle\langle 0|$, and $|1\rangle\langle 1|$ are all projectors. More generally, show that for arbitrary unit vector $|\psi\rangle \in \mathbb{C}^d$, $|\psi\rangle\langle\psi|$ is a projector.

It is important to note that since a projector Π's eigenvalues are all 0 or 1, its spectral decomposition must take the form $\Pi = \sum_i |\psi_i\rangle\langle\psi_i|$, where $\{|\psi_i\rangle\}$ are an orthonormal set. Conversely, summing any set of orthonormal $\{|\psi_i\rangle\}$ in this fashion yields a projector, as you will now show.

Exercise 3.4. Let $\{|\psi_i\rangle\}$ be an orthonormal set. Prove that $\Pi = \sum_i |\psi_i\rangle\langle\psi_i|$ is a projector.

Observe that a projector Π has rank 1 if and only if $\Pi = |\psi\rangle\langle\psi|$ for some $|\psi\rangle \in \mathbb{C}^d$, since the rank of Π equals[1] the number of non-zero eigenvalues of Π, and here $\Pi = |\psi\rangle\langle\psi|$ is a spectral decomposition of Π. Finally, let us develop an intuition for what a projector actually *does* — for any projector $\Pi = \sum_i |\psi_i\rangle\langle\psi_i|$ and state $|\phi\rangle$ to be measured, we have

$$\Pi|\phi\rangle = \left(\sum_i |\psi_i\rangle\langle\psi_i|\right)|\phi\rangle = \sum_i |\psi_i\rangle(\langle\psi_i|\phi\rangle) = \sum_i (\langle\psi_i|\phi\rangle)|\psi_i\rangle \in \mathrm{Span}(\{\psi_i\}),$$

where note $\langle\psi_i|\phi\rangle \in \mathbb{C}$. Thus, Π projects us down onto the span of the vectors $\{|\psi_i\rangle\}$.

Exercise 3.5. Consider three-dimensional vector $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle \in \mathbb{C}^3$ and $\Pi = |0\rangle\langle 0| + |1\rangle\langle 1|$. Compute $\Pi|\phi\rangle$, and observe that the latter indeed lies in the two-dimensional space $\mathrm{Span}(\{|0\rangle, |1\rangle\})$.

Projective measurements. With projectors in hand, we can now define a projective measurement. A *projective measurement* is a set of projectors $B = \{\Pi_i\}_{i=0}^m$ such that $\sum_{i=0}^m \Pi_i = I$. The latter condition is called the *completeness* relation. If each Π_i is rank one, i.e. $\Pi_i = |\psi_i\rangle\langle\psi_i|$, then we say that B models a *measurement in basis* $\{|\psi_i\rangle\}$. Often, we shall measure in the computational basis, which is specified by $B = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ in the case of $\mathbb{C}^2$ (and generalizes as $B = \{|i\rangle\langle i|\}_{i=0}^{d-1}$ for $\mathbb{C}^d$).

[1] This holds since Π is Hermitian, and hence also normal.

Exercise 3.6. Verify that $B = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ is a projective measurement on $\mathbb{C}^2$.

With a projective measurement $B = \{\Pi_i\}_{i=0}^m \subseteq \mathbb{C}^d$ in hand, let us specify how one *uses* B. Suppose our quantum system is in state $|\psi\rangle \in \mathbb{C}^d$. Then, the probability of obtaining outcome $i \in \{0, \ldots, m\}$ when measuring $|\psi\rangle$ with B is given by

$$\Pr(\text{outcome } i) = \mathrm{Tr}(\Pi_i|\psi\rangle\langle\psi|\Pi_i) = \mathrm{Tr}(\Pi_i^2|\psi\rangle\langle\psi|) = \mathrm{Tr}(\Pi_i|\psi\rangle\langle\psi|),$$

where the second equality follows by the cyclic property of the trace and the third since Π_i is a projector.

Exercise 3.7. Let $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \in \mathbb{C}^2$. Show that if we measure in the computational basis, i.e. using $B = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$, then the probabilities of obtaining outcomes 0 and 1 are $|\alpha|^2$ and $|\beta|^2$, respectively.

The exercise above has an important moral — requiring a quantum state $|\psi\rangle$ to be a *unit* vector (i.e. $|\alpha|^2 + |\beta|^2 = 1$) ensures that when measuring $|\psi\rangle$, the distribution over the outcomes is a valid *probability distribution*, i.e. the probabilities for all possible outcomes sum to 1. The other important take-home message here is that measurements in quantum mechanics are inherently *probabilistic* — in general, the outcomes cannot be perfectly predicted!

Finally, we started this lecture by saying that the very act of measuring a quantum state disturbs the system. Let us now formalize this; we will crucially use the fact discussed earlier that a projector *projects* a vector $|\psi\rangle$ down into a smaller subspace. Specifically, upon obtaining outcome Π_i when measuring B, the state of system "collapses" to

$$\frac{\Pi_i|\psi\rangle\langle\psi|\Pi_i}{\mathrm{Tr}(\Pi_i|\psi\rangle\langle\psi|\Pi_i)} = \frac{\Pi_i|\psi\rangle\langle\psi|\Pi_i}{\mathrm{Tr}(\Pi_i|\psi\rangle\langle\psi|)}. \tag{3.1}$$

Note the denominator above is a scalar, and is just the probability of outcome i. There are two points here which may confuse you: Why have we written the output state as a matrix $\Pi_i|\psi\rangle\langle\psi|\Pi_i$ rather than a vector $\Pi_i|\psi\rangle$, and what is the role of the denominator? Let us handle each of these in order.

First, conditioned on outcome Π_i, the output state is indeed a vector, namely $\Pi_i|\psi\rangle$. However, there is a more general formalism which we shall discuss shortly called the *density operator formalism*, in which quantum states are written as matrices, not vectors. Specifically, the "density matrix" representing vector $|\psi\rangle$ would be written as matrix $|\psi\rangle\langle\psi|$. The density operator formalism is more general than the state vector approach we have taken so far, and will be crucial for studying individual subsystems of a larger composite quantum state. Thus, the answer to question 1 is that we have written the output as a matrix simply to slowly ease the transition into the density matrix formalism.

The motive behind question 2 is somewhat less sinister — the problem here is that since we projected out part of $|\psi\rangle$ during the measurement, the output $\Pi_i|\psi\rangle$ may not necessarily be normalized. To renormalize $\Pi_i|\psi\rangle$, we simply divide by its Euclidean norm to obtain

$$|\psi'\rangle = \frac{\Pi_i|\psi\rangle}{\| \Pi_i|\psi\rangle \|_2} = \frac{\Pi_i|\psi\rangle}{\sqrt{\langle\psi|\Pi_i\Pi_i|\psi\rangle}} = \frac{\Pi_i|\psi\rangle}{\sqrt{\langle\psi|\Pi_i|\psi\rangle}}.$$

The state $|\psi'\rangle$ describes the post-measurement state of our system, assuming we have obtained outcome i.

Exercise 3.8. Show that (the density matrix) $|\psi'\rangle\langle\psi'|$ equals the expression in Equation (3.1).

Exercise 3.9. Let $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \in \mathbb{C}^2$. Show that if we measure in the computational basis, i.e. using $B = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$, and obtain outcome $i \in \{0, 1\}$, then the post-measurement state is $|i\rangle$ (or $|i\rangle\langle i|$ in density matrix form).

A final quirk we should iron out is the following — in terms of measurements, what is the consequence of the fact that a projector Π_i satisfies $\Pi_i^2 = \Pi_i$? Well, if you observe a quantum system now, and then again five minutes from now, and if the system has not been subjected to any gates or noise in between the measurements, then the two measurement results you obtain should agree (I realize the study of quantum mechanics has likely distorted your view of when you can trust your intuition, but this is one case in which you can). To model this, suppose we measure using $B = \{\Pi_i\}$ and obtain results i and j in measurements 1 and 2, respectively. Then:

$$\Pr(\text{outcome } j \mid \text{ outcome } i) = \mathrm{Tr}\left(\Pi_j \frac{\Pi_i|\psi\rangle\langle\psi|\Pi_i}{\mathrm{Tr}(\Pi_i|\psi\rangle\langle\psi|)}\Pi_j\right) = \frac{\mathrm{Tr}\left(\Pi_j\Pi_i|\psi\rangle\langle\psi|\Pi_i\Pi_j\right)}{\mathrm{Tr}(\Pi_i|\psi\rangle\langle\psi|)} \tag{3.2}$$

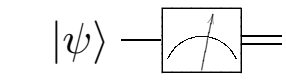
To simplify this expression, we use the fact that if the completeness relation holds for projectors $\{\Pi_i\}$, i.e. $\sum_i \Pi_i = I$, then it turns out that $\Pi_j\Pi_i = \delta_{ij}\Pi_i$, where recall δ_{ij} is the Kronecker delta. Thus, if $i \neq j$, Equation (3.2) equals 0, and if $i = j$, it reduces to

$$\frac{\mathrm{Tr}\left(\Pi_j\Pi_i|\psi\rangle\langle\psi|\Pi_i\Pi_j\right)}{\mathrm{Tr}(\Pi_i|\psi\rangle\langle\psi|)} = \frac{\mathrm{Tr}\left(\Pi_i^2|\psi\rangle\langle\psi|\Pi_i^2\right)}{\mathrm{Tr}(\Pi_i|\psi\rangle\langle\psi|)} = \frac{\mathrm{Tr}\left(\Pi_i|\psi\rangle\langle\psi|\Pi_i\right)}{\mathrm{Tr}(\Pi_i|\psi\rangle\langle\psi|)} = \frac{\mathrm{Tr}\left(\Pi_i|\psi\rangle\langle\psi|\right)}{\mathrm{Tr}(\Pi_i|\psi\rangle\langle\psi|)} = 1,$$

i.e. measuring the state a second time again yields outcome i with probability 1, as desired. Thus, although observing a state for the first time disturbs it, subsequent measurements will consistently return the same measurement result!

Exercise 3.10. Suppose we measure $|0\rangle$ in basis $B = \{|+\rangle\langle +|, |-\rangle\langle -|\}$. What are the probabilities of outcomes $+$ and $-$, respectively? What are the post-measurement states if one obtains outcome $+$ or $-$, respectively?

We close this section by giving the circuit symbol which denotes a measurement of a qubit $|\psi\rangle \in \mathbb{C}^2$ in the computational basis $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$:



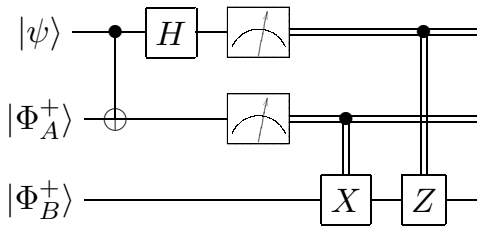
The double-wires on the right side indicate that the output of the measurement is a classical string (indicating which measurement outcome was obtained).

3.2 Quantum teleportation

With the concepts of the Bell state and measurement in hand, we can discuss our first neat computational trick: Quantum teleportation. Suppose you have a single-qubit quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ in your possession (i.e. as a physical system, not on paper), but that you do not know the values of α and β. Your friend Alice now phones you and asks to borrow

your state. How can you send it to her? One obvious way is simply to pop your system in the mail and physically send it over. However, it turns out that by exploiting the phenomenon of entanglement, you can do something incredible — by sending two classical bits over the telephone to Alice, you can "teleport" $|\psi\rangle$ instantly to her!

To teleport $|\psi\rangle$, we assume that you and Alice each share half of a Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ to begin with; specifically, you hold qubit 1 of $|\Phi^+\rangle$, and Alice holds qubit 2. The teleportation circuit is then given as follows:



Let us break this down piece by piece. The first two wires are held by you; wire 1 holds the state to be teleported, $|\psi\rangle$, and wire 2 holds your half of $|\Phi^+\rangle$. The third wire holds Alice's half of $|\Phi^+\rangle$. Note that we have used $|\Phi_A^+\rangle$ and $|\Phi_B^+\rangle$ to denote the two "halves" of $|\Phi^+\rangle$, but this is poor notation — read literally, this diagram suggests $|\Phi^+\rangle = |\Phi_A^+\rangle \otimes |\Phi_B^+\rangle$, which is *not* true since $|\Phi^+\rangle$ is entangled, and hence from last lecture we know that there do not exist states $|\Phi_A^+\rangle, |\Phi_B^+\rangle$ such that $|\Phi^+\rangle = |\Phi_A^+\rangle \otimes |\Phi_B^+\rangle$. This notation is chosen simply because I was unable to find a way to correctly display $|\Phi^+\rangle$ across wires 2 and 3 in the brief time I dedicated to the task.

The circuit can be divided into 5 steps: Step 1 performs the CNOT, Step 2 the Hadamard gate, Step 3 measures qubits 1 and 2, Step 4 applies a conditional X gate, and Step 5 applies a conditional Z gate. The latter two require clarification: The conditional X gate here takes a classical bit b as input (hence the incoming wire at the top is a double line), and applies X if and only if $b = 1$. The conditional Z gate is defined analogously.

Now that we have technically parsed this diagram, let us intuitively parse it. First, you begin in Steps 1 and 2 by performing a CNOT and Hadamard on your qubits, followed by a standard basis measurement in Step 3. Since a measurement in the standard basis maps each qubit to either $|0\rangle$ or $|1\rangle$, the output of your two measurements can jointly be thought of as one of the four bit strings 00, 01, 10, or 11. Call these bits $b_0 b_1$. Now you pick up the telephone, call Alice, and tell her the value of $b_0 b_1$. Conditioned on b_0, she applies X to her half of the Bell pair, followed by Z conditioned on b_1. The claim is that at this point, Alice's qubit's state has been magically converted to $|\psi\rangle$. In fact, as we shall see shortly, $|\psi\rangle$ has also disappeared from your possession! In this sense, teleportation has taken place.

Let us formally analyze the action of this circuit. Denote by $|\psi_i\rangle$ for $i \in \{0, \ldots, 5\}$ the joint state of your and Alice's systems immediately after Step i has taken place. Here, we define $|\psi_0\rangle$ as the initial joint state before any gates are applied; it is given by

$$|\psi_0\rangle = |\psi\rangle|\Phi^+\rangle = (\alpha|0\rangle + \beta|1\rangle)\left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle\right) = \frac{1}{\sqrt{2}}\left(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle\right).$$

After Step 1, i.e. after the CNOT, we have state

$$\frac{1}{\sqrt{2}}\left(\alpha|000\rangle + \alpha|011\rangle + \beta|110\rangle + \beta|101\rangle\right).$$

After Step 2, i.e. after the Hadamard gate, we have

$$\begin{aligned} & \frac{1}{\sqrt{2}}\left(\alpha|+\rangle|00\rangle + \alpha|+\rangle|11\rangle + \beta|-\rangle|10\rangle + \beta|-\rangle|01\rangle\right) \\ = \; & \frac{1}{2}\left(\alpha(|0\rangle+|1\rangle)|00\rangle + \alpha(|0\rangle+|1\rangle)|11\rangle + \beta(|0\rangle-|1\rangle)|10\rangle + \beta(|0\rangle-|1\rangle)|01\rangle\right) \\ = \; & \frac{1}{2}\left(|00\rangle(\alpha|0\rangle+\beta|1\rangle) + |01\rangle(\alpha|1\rangle+\beta|0\rangle) + |10\rangle(\alpha|0\rangle-\beta|1\rangle) + |11\rangle(\alpha|1\rangle-\beta|0\rangle)\right). \quad (3.3) \end{aligned}$$

Let us now pause and analyze the state of affairs. There are four terms in this superposition, each of which begins with a distinct bit string $|00\rangle$, $|01\rangle$, $|10\rangle$, or $|11\rangle$. This means that if you now measure qubits 1 and 2 in the standard basis and obtain outcome (say) $|00\rangle$, then Alice's qubit on wire 3 collapses to the only consistent possibility, $\alpha|0\rangle+\beta|1\rangle$. In this case, teleportation has already taken place.

Exercise 3.11. Let $|\phi\rangle \in (\mathbb{C}^2)^{\otimes 3}$ denote the state in Equation (3.3). Suppose you now measure qubits 1 and 2 in the standard basis. This can be modelled by projective measurement

$$B = \{|00\rangle\langle 00| \otimes I, |01\rangle\langle 01| \otimes I, |10\rangle\langle 10| \otimes I, |11\rangle\langle 11| \otimes I\},$$

where we have I on qubit 3 since we are not measuring it. Show that the probability of outcome 00 is $1/4$. Next, show that conditioned on outcome 00, the post-measurement state collapses to $|00\rangle(\alpha|0\rangle + \beta|1\rangle)$.

More generally, the four possible outcomes upon measuring qubits 1 and 2 result in four distinct residual states on Alice's qubit as follows:

$$00 \mapsto \alpha|0\rangle + \beta|1\rangle \qquad 01 \mapsto \alpha|1\rangle + \beta|0\rangle \qquad 10 \mapsto \alpha|0\rangle - \beta|1\rangle \qquad 11 \mapsto \alpha|1\rangle - \beta|0\rangle.$$

Thus, if you simply send the two bits b_0b_1 encoding the measurement outcome to Alice, then regardless of the value of b_0b_1, she can recover your original state $\alpha|0\rangle + \beta|1\rangle$ via the following identities:

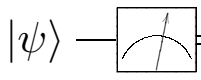
$$X(\alpha|1\rangle+\beta|0\rangle) = \alpha|0\rangle+\beta|1\rangle \qquad Z(\alpha|0\rangle-\beta|1\rangle) = \alpha|0\rangle+\beta|1\rangle \qquad ZX(\alpha|1\rangle-\beta|0\rangle) = \alpha|0\rangle+\beta|1\rangle.$$

Exercise 3.12. Verify that indeed $ZX(\alpha|1\rangle - \beta|0\rangle) = \alpha|0\rangle + \beta|1\rangle$.

In other words, by conditionally applying X and Z based on the outputs b_0b_1 from your measurement, Alice can successfully recover your state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. This is precisely what is depicted in Steps 4 and 5 of the teleportation circuit. Finally, note that since measuring your qubits leaves you in one of the four standard basis states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, the state $|\psi\rangle$ has now "disappeared" from your possession!

3.3 Simulating arbitrary measurements via standard basis measurements

Let us continue to digest the quantum teleportation circuit with an observation. You may have noticed that in Section 3.1, we only described the circuit diagram representation of a *standard basis measurement* on a qubit, i.e.



whereas more generally projective measurements allow measuring in an *arbitrary* basis $B = \{|\psi_1\rangle, |\psi_2\rangle\} \subseteq \mathbb{C}^2$. It turns out that without loss of generality, we can restrict ourselves to measurements in the standard basis as follows. For this, we require an important fact about unitary matrices, which intuitively says that a unitary matrices maps orthonormal bases to orthonormal bases:

Lemma 3.13. *Let $\mathcal{B}_1 = \{|\psi_i\rangle\}_{i=1}^d \subseteq \mathbb{C}^d$ be an orthonormal basis. Then, for any unitary $U \in \mathcal{L}(\mathbb{C}^d)$, $\mathcal{B}_2 = \{U|\psi_i\rangle\}_{i=1}^d \subseteq \mathbb{C}^d$ is an orthonormal basis.*

Proof. Since U is unitary, we have $U^\dagger U = I$. Therefore, for any $i, j \in \{1, \ldots, d\}$

$$(\langle\psi_i|U^\dagger)(U|\psi_j\rangle) = \langle\psi_i|\psi_j\rangle = \delta_{ij},$$

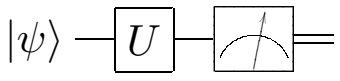
i.e. $\mathcal{B}_2 = \{U|\psi_i\rangle\}_{i=1}^d$ is an orthonormal basis. □

In particular, for *any* orthonormal bases $\mathcal{B}_1 = \{|\psi_i\rangle\}_{i=1}^d \subseteq \mathbb{C}^d$ and $\mathcal{B}_2 = \{|\phi_i\rangle\}_{i=1}^d \subseteq \mathbb{C}^d$, there exists a unitary U mapping $\mathcal{B}_1$ to $\mathcal{B}_2$. This unitary is given by

$$U = \sum_{i=1}^{d} |\phi_i\rangle\langle\psi_i|.$$

Exercise 3.14. Verify that for any $i \in \{1, \ldots, d\}$, $U|\psi_i\rangle = |\phi_i\rangle$. Verify that U is unitary, i.e. that $U^\dagger U = I$. (Hint: Use the fact that for any orthonormal basis $\{|\phi_i\rangle\}_{i=1}^d$, we have $\sum_i |\phi_i\rangle\langle\phi_i| = I$.)

Let us now return to simulating a measurement in an arbitrary basis $B = \{|\psi_1\rangle, |\psi_2\rangle\}$ on $\mathbb{C}^2$ with standard basis measurements. By the exercise above, there exists a unitary U mapping B to the standard basis, i.e. $U|\psi_1\rangle = |0\rangle$ and $U|\psi_2\rangle = |1\rangle$. Then, instead of measuring a state $|\psi\rangle \in \mathbb{C}^2$ in basis B, one can equivalently measure $U|\psi\rangle$ in the standard basis. In other words, we can simulate a measurement in B via the circuit diagram



To see why this works, consider the probability of obtaining outcome $|\psi_1\rangle$ when measuring in B, which is

$$\Pr(\text{outcome } |\psi_1\rangle) = \mathrm{Tr}(|\psi_1\rangle\langle\psi_1| \cdot |\psi\rangle\langle\psi|) = \mathrm{Tr}((U^\dagger|0\rangle\langle 0|U)|\psi\rangle\langle\psi|) = \mathrm{Tr}(|0\rangle\langle 0|(U|\psi\rangle\langle\psi|U^\dagger)).$$

In other words, one might as well measure state $U|\psi\rangle$ against operator $|0\rangle\langle 0|$.

Exercise 3.15. Give a quantum circuit for measuring a qubit in the $\{|+\rangle, |-\rangle\}$ basis.

Returning to the teleportation circuit, let us now observe that Steps 1 and 2 together are simply the *reverse* of our circuit from Lecture 2 which mapped the standard basis to the Bell basis. In other words, Steps 1 and 2 in teleportation map the Bell basis back to the standard basis — thus, Steps 1, 2, and 3 are simply measuring the first two qubits in the Bell basis, i.e. by projectors $B = \{|\Phi^+\rangle\langle\Phi^+|, |\Psi^+\rangle\langle\Psi^+|, |\Phi^-\rangle\langle\Phi^-|, |\Psi^-\rangle\langle\Psi^-|\}$.

3.4 Revisiting Schrödinger's cat

In Lecture 2, we introduced Schrödinger's cat, which through the magic of superposition, managed to be both alive and dead simultaneously, *so long as* the box containing the cat remained sealed. This begged the question: What happens if one opens the box and looks inside? Surely, one does not expect to find both a living and dead cat. The paradox here can finally be resolved via the measurement postulate — the act of opening the box and looking at the cat is itself a *measurement*. This measurement effectively collapses the superposition of dead and alive, leaving simply a cat which is either dead or alive. Note here that the very act of looking at something implies a measurement — this is because in order to "observe" the cat, photons must interact with the cat, and subsequently reach your eyes. This interaction and subsequent processing of the information contained in the photons by your brain "constitutes" a measurement.

4 No Cloning, Entanglement, and Density Matrices

> *"[Entanglement is] not just one of many traits, but the characteristic trait of quantum physics..."*
> — Erwin Schrödinger

The primary focus of the next two lectures is to further practice working with quantum states and measurements, a natural avenue for which is the study of cloning, quantum entanglement, and non-local games. In the process, we will run into a more general formalism for describing quantum states, known as the *density operator* formalism. We begin in this lecture with cloning, quantum entanglement, and density operators.

4.1 No cloning theorem

One of the most natural operations we perform daily on classical computers is the act of copying or *cloning* classical bits. For example, you might make a copy of a file containing your LaTeX-typed homework to keep as a backup. On the negative side, a criminal might copy a classical object like a banknote (though it is not easy) to use as forged money. However, quantumly it turns out that copying quantum bits is impossible — given an arbitrary, unknown quantum state $|\psi\rangle \in \mathbb{C}^2$, there is no quantum circuit which creates a perfect copy of $|\psi\rangle$! How can we formalize such a statement?

We use proof by contradiction. Suppose there exists a unitary $U \in \mathcal{L}(\mathbb{C}^2 \otimes \mathbb{C}^2)$ which, for *any* $|\psi\rangle$, maps $|\psi\rangle \otimes |0\rangle$ to $|\psi\rangle \otimes |\psi\rangle$, i.e. creates a copy of $|\psi\rangle$. To obtain the desired contradiction, we begin by applying U to two arbitrary states $|\psi_1\rangle, |\psi_2\rangle \in \mathbb{C}^2$, obtaining

$$|\phi\rangle = U(|\psi_1\rangle \otimes |0\rangle) = |\psi_1\rangle \otimes |\psi_1\rangle \qquad \text{and} \qquad |\phi'\rangle = U(|\psi_2\rangle \otimes |0\rangle) = |\psi_2\rangle \otimes |\psi_2\rangle.$$

What happens if we now take the inner product $\langle\phi|\phi'\rangle$? The middle terms of these equations yield

$$(\langle\psi_1| \otimes \langle 0|)U^\dagger U(|\psi_2\rangle \otimes |0\rangle) = (\langle\psi_1| \otimes \langle 0|)(|\psi_2\rangle \otimes |0\rangle) = \langle\psi_1|\psi_2\rangle\langle 0|0\rangle = \langle\psi_1|\psi_2\rangle.$$

On the other hand, the right sides of the equations yield

$$(\langle\psi_1| \otimes \langle\psi_1|)(|\psi_1\rangle \otimes |\psi_1\rangle) = \langle\psi_1|\psi_2\rangle\langle\psi_1|\psi_2\rangle = (\langle\psi_1|\psi_2\rangle)^2.$$

Combining these two, we have that $\langle\psi_1|\psi_2\rangle = (\langle\psi_1|\psi_2\rangle)^2$ for some complex number $\langle\psi_1|\psi_2\rangle \in \mathbb{C}$. This is equivalent to saying $c = c^2$ for $c \in \mathbb{C}$, which has only two solutions — either $c = 1$ or $c = 0$. In other words, unless $|\psi_1\rangle$ and $|\psi_2\rangle$ are the same vector (in which case $\langle\psi_1|\psi_2\rangle = 1$) or are orthogonal (i.e. $\langle\psi_1|\psi_2\rangle = 0$), we have a contradiction. Thus, there does not exist a U which can perform the mapping $|\psi\rangle \otimes |0\rangle \mapsto |\psi\rangle \otimes |\psi\rangle$ for arbitrary $|\psi\rangle$.

Exercise 4.1. This section showed that cloning arbitrary states $|\psi\rangle \in \mathbb{C}^2$ is impossible. However, the proof technique failed when the states $|\psi_1\rangle$ and $|\psi_2\rangle$ are orthogonal. Indeed, it turns out in this case cloning is possible. Give a quantum circuit which can clone states $|0\rangle$ and $|1\rangle$, i.e. maps $|0\rangle|0\rangle \mapsto |0\rangle|0\rangle$ and $|1\rangle|0\rangle \mapsto |1\rangle|1\rangle$. How about a circuit which clones $|+\rangle$ and $|-\rangle$? (Hint: Convert the $+/-$ basis into the standard basis first, then apply your cloning circuit for the standard basis.)

4.2 Quantum entanglement

We have already seen the concept of quantum entanglement, and in particular used Bell states such as $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ as a *resource* for performing interesting tasks such as quantum teleportation. But what exactly *is* entanglement, and how can we recognize whether a given state $|\psi\rangle$ is entangled?

History. First, a bit of history. The term "entanglement" was coined in 1935 by physicist Erwin Schrödinger, who used the term "Vershränkung", which in colloquial "non-physicist" German means "folding of the arms". As discussed in a previous lecture, the question of whether entanglement truly exists has been a subject of intense debate. The famous Einstein-Podolsky-Rosen paper of 1935, in particular, argued that quantum mechanics could not be a complete physical theory due to its prediction of entangled states such as $|\Phi^+\rangle$. In 1964, however, physicist John Bell proposed what is now known as a "Bell inequality" or "Bell test"; this test could in principle be run in a lab to confirm whether Nature indeed allows the strong type of correlations between qubits which entanglement would allow. Recently, such tests have been studied under the guise of "non-local games", which is a topic we shall soon visit. Moreover, it is important to note that we nowadays know that any quantum computation (on pure quantum states) cannot achieve an exponential speedup over classical computers *unless* the amount of quantum entanglement in the system is "large". Thus, entanglement is generally regarded as an important resource.

Bipartite entanglement. In this lecture, we study the notion of bipartite entanglement, meaning entanglement between a pair of quantum systems. Let $|\psi\rangle \in \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$ be an arbitrary bipartite state. How can we tell if $|\psi\rangle$ is entangled? Strictly speaking, earlier we defined $|\psi\rangle$ as *entangled* if it cannot be written as the tensor product of two states $|\psi_1\rangle \in \mathbb{C}^{d_1}$ and $|\psi_2\rangle \in \mathbb{C}^{d_2}$, i.e.

$$\forall\, |\psi_1\rangle \in \mathbb{C}^{d_1}, |\psi_2\rangle \in \mathbb{C}^{d_2} \quad \text{we have} \quad |\psi_1\rangle \otimes |\psi_2\rangle \neq |\psi\rangle.$$

For example, let us confirm that $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is entangled. Again, we give a proof by contradiction. Suppose there exist $|\psi_1\rangle \in \mathbb{C}^2, |\psi_2\rangle \in \mathbb{C}^2$ such that $|\psi_1\rangle \otimes |\psi_2\rangle = |\Phi^+\rangle$. If we write $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$, we have

$$|\psi_1\rangle \otimes |\psi_2\rangle = (\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle) = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle.$$

Since this supposed to equal $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, we must have $\alpha_1\beta_2 = \beta_1\alpha_2 = 0$ (since the standard basis vectors $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ are orthogonal). Suppose, without loss of generality, that $\beta_2 = 0$, so that $\alpha_1\beta_2 = 0$. Then, we must also have $\beta_1 = 0$ so that $\beta_1\alpha_2 = 0$ (the other option is to set $\alpha_2 = 0$, but this would yield $\alpha_2 = \beta_2 = 0$, i.e. $|\psi_2\rangle$ is not a unit vector). But if $\beta_1 = \beta_2 = 0$, then $|\psi_1\rangle = |0\rangle$ and $|\psi_2\rangle = |0\rangle$, so that $|\psi_1\rangle \otimes |\psi_2\rangle = |00\rangle \neq |\Phi^+\rangle$! Thus, we have a contradiction, and so $|\Phi^+\rangle$ is entangled.

The Schmidt decomposition. Rather than go through such proofs each time we wish to check if $|\psi\rangle$ is entangled, there is a more elegant tool we can use known as the *Schmidt decomposition*. To keep our discussion simple, we shall restrict our attention to the setting of two qubits; the ideas here extend straightforwardly to arbitrary local dimensions d_1 and d_2. The Schmidt decomposition states that any vector $|\psi\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^2$ can be written in the following special form:

$$|\psi\rangle = \alpha_0 |a_0\rangle|b_0\rangle + \alpha_1 |a_1\rangle|b_1\rangle,$$

where the α_i are real and satisfy $\alpha_i \geq 0$, and where $\{|a_i\rangle\}_{i=1}^2$ and $\{|b_i\rangle\}_{i=1}^2$ are orthonormal bases for $\mathbb{C}^2$. The number of non-zero α_i is called the *Schmidt rank* of $|\psi\rangle$.

Exercise 4.2. Verify that $|00\rangle$ and $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ are already written in terms of Schmidt decompositions. What are the Schmidt ranks of these two states?

Observe now that if $|\psi\rangle$ has Schmidt rank 1, then clearly it is not entangled, as its Schmidt decomposition is in tensor product form. And this is no coincidence — it turns out that an arbitrary state $|\psi\rangle$ is entangled if and only if its Schmidt rank is strictly greater than 1.

Exercise 4.3. Use the notion of Schmidt rank to answer the following: Is $|+\rangle|-\rangle$ entangled? How about $\sqrt{1/1000}|+\rangle|-\rangle + \sqrt{999/1000}|-\rangle|+\rangle$? Finally, how about $\frac{1}{\sqrt{2}}(|0\rangle|+\rangle - |0\rangle|-\rangle)$? (Hint: For this last one, be careful! Is it written in its Schmidt decomposition?)

In this course, we shall not be tasked with *finding* Schmidt decompositions of states. However, like the spectral decomposition for matrices, when doing proofs it is often useful to know that bipartite states $|\psi\rangle$ can always be written in this way. Let us close by pointing out a subtle fact about Schmidt decompositions. Note that any vector $|\psi\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^2$ can be written as

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle.$$

This decomposition requires four terms (i.e. one per standard basis vector for $\mathbb{C}^4$). Part of the power of the Schmidt decomposition is that it manages to write this same state using just *two* terms.

4.3 Density matrices

We have delved further into the structure of entanglement and learned that a bipartite state $|\psi\rangle$ is entangled if and only if its Schmidt rank is at least 2. However, let's take a step back and return to our statement that $|\psi\rangle$ is entangled if it cannot be written as $|\psi_1\rangle \otimes |\psi_2\rangle$ for some $|\psi_1\rangle \in \mathbb{C}^{d_1}$ and $|\psi_2\rangle \in \mathbb{C}^{d_2}$. Intuitively, states of the form $|\psi_1\rangle \otimes |\psi_2\rangle$, called *tensor product states*, are nice because one can immediately read off the state of each qubit — qubit one is in state $|\psi_1\rangle$, and qubit two in $|\psi_2\rangle$. Since entangled states cannot be written in tensor product form, however, this raises the question: How can we describe the state of qubit one in, say, the Bell pair $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$? The answer lies in a more general formalism for describing quantum states, known as the *density matrix formalism*.

Thus far, we have said that an arbitrary d-dimensional quantum state is described by a unit vector $|\psi\rangle$ in $\mathbb{C}^d$. Such states are called *pure* because we know exactly which state $|\psi\rangle$ we have. Now, suppose we play a game in which with probability 1/2, I give you state $|\psi_1\rangle$, and with probability 1/2, I give you state $|\psi_2\rangle$, and I don't tell you *which* state I've given you. How

can you describe the quantum state in your possession, given that you don't know whether you actually have $|\psi_1\rangle$ or $|\psi_2\rangle$? This is done via the *density operator*

$$\rho = \frac{1}{2}|\psi_1\rangle\langle\psi_1| + \frac{1}{2}|\psi_2\rangle\langle\psi_2|. \tag{4.1}$$

More generally, if we play this game with m possible states $|\psi_i\rangle$, each given with probability p_i (i.e. $\sum_i p_i = 1$ and $p_i \geq 0$), then the density operator describing your system is

$$\rho = \sum_{i=1}^{m} p_i |\psi_i\rangle\langle\psi_i|.$$

Such a state is called *mixed* because you don't know with certainty which $|\psi_i\rangle$ you have in your possession. Let us make two important observations here.

1. *Mixtures vs. superpositions.* A mixture of states $\sum_i p_i|\psi_i\rangle\langle\psi_i|$ is entirely different than a superposition of states $\sum_i \alpha_i|\psi_i\rangle$. For starters, the former is a sum of matrices, whereas the latter is a sum of vectors. More importantly, the former models a state of ignorance about which $|\psi_i\rangle$ we actually have in our possession — we know our system is in *precisely one* such $|\psi_i\rangle$, but which one is unknown. In stark contrast, in a superposition, our system is in *all* of the states $|\psi_i\rangle$ simultaneously.

2. *Pure states.* If there is only one state $|\psi_i\rangle$ in the mixture, i.e. $p_i = 1$ for some i, then the mixture simply reads $\rho = |\psi_i\rangle\langle\psi_i|$. This state is called pure and has rank 1. Conversely, for any pure state $|\psi\rangle$, its density matrix is the rank 1 operator $\rho = |\psi\rangle\langle\psi|$.

Let us stress the first point above with a concrete example. The density matrix $\rho = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|$ is:

$$\rho = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} = \frac{I}{2}.$$

In contrast, the state vector $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ is given by

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Note that the former is a matrix, while the latter is a vector, i.e. these are completely different objects!

Exercise 4.4. What is the density matrix for pure state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$? Next, write down the 2×2 density matrix $\rho = \frac{1}{3}|0\rangle\langle 0| + \frac{2}{3}|+\rangle\langle +|$.

Let us now generalize our discussion further. A linear operator $\rho \in \mathcal{L}(\mathbb{C}^d)$ is called a *density matrix* if the following two properties hold:

- ρ is positive semi-definite, i.e. is Hermitian and has non-negative real eigenvalues, and
- ρ has trace 1, i.e. $\mathrm{Tr}(\rho) = 1$.

Let us check that the density operators we have considered thus far satisfy these two properties. For example, for ρ from Equation (4.1), we have

$$\mathrm{Tr}(\rho) = \mathrm{Tr}\left(\frac{1}{2}|\psi_1\rangle\langle\psi_1| + \frac{1}{2}|\psi_2\rangle\langle\psi_2|\right) = \frac{1}{2}\mathrm{Tr}\left(|\psi_1\rangle\langle\psi_1|\right) + \frac{1}{2}\mathrm{Tr}\left(|\psi_2\rangle\langle\psi_2|\right) = \frac{1}{2}\langle\psi_1|\psi_1\rangle + \frac{1}{2}\langle\psi_2|\psi_2\rangle = 1.$$

As for the property of being positive semidefinite, we use the following fact: For any two positive semi-definite matrices A and B and real numbers $p, q \geq 0$, it holds that $pA + qB$ is also positive semi-definite. In particular, since both $|\psi_1\rangle\langle\psi_1|$ and $|\psi_2\rangle\langle\psi_2|$ are positive semi-definite (as you will show in the exercise below), we have that ρ is also positive semi-definite. Thus, ρ is a valid density operator.

Exercise 4.5. Why is $|\psi\rangle\langle\psi|$ positive semi-definite for *any* $|\psi\rangle$? (Hint: Use the spectral decomposition).

In fact, the requirements that $\mathrm{Tr}(\rho) = 1$ and ρ be positive semidefinite allow us to recover exactly the interpretation of mixed states which this section started with — taking the spectral decomposition of ρ, we have

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

Since ρ is positive semidefinite, we know $\lambda_i \geq 0$ for all i. Moreover, recalling that $\mathrm{Tr}(\rho) = \sum_i \lambda_i$, we know that since $\mathrm{Tr}(\rho) = 1$, it holds that $\sum_i \lambda_i = 1$. Thus, $\{\lambda_i\}_i$ forms a probability distribution. This means that we can interpret ρ as follows: With probability λ_i, prepare state $|\lambda_i\rangle$. This is precisely the idea we started this section with! We thus have a good picture of how one can describe probabilistic mixtures over pure quantum states.

Exercise 4.6. Why are the Pauli gates X, Y, and Z not density matrices?

The maximally mixed state. Finally, let us consider a very special density matrix in $\mathcal{L}(\mathbb{C}^d)$, the *maximally mixed* state $\rho = I/d$ (where I denotes the identity matrix). Note that ρ is positive semidefinite since I is positive semidefinite (in fact, I has eigenvalues all equal to 1), and $\mathrm{Tr}(I/d) = \frac{1}{d}\mathrm{Tr}(I) = 1$. Thus, ρ is a valid density operator. But what exactly does I/d *represent*? Here, we use the fact that for any orthonormal basis $\{|\psi_i\rangle\}_{i=1}^d$ for $\mathbb{C}^d$, we have

$$\sum_{i=1}^{d} |\psi_i\rangle\langle\psi_i| = I.$$

In other words, for *any* orthonormal basis $\{|\psi_i\rangle\}_{i=1}^d$, ρ represents the following state: Pick state $|\psi_i\rangle$ with probability $1/d$, and prepare $|\psi_i\rangle$. Since this holds for any basis, we conclude that ρ gives us absolutely no information about which state $|\psi\rangle$ we actually have — *every* state is an eigenvector of ρ, and the eigenvalues of ρ form a uniform distribution. Thus, the maximally mixed state represents the case where we know *nothing* about the state of our system! This will be crucial when we discuss entanglement again below.

4.3.1 The partial trace operation

Our motivation in introducing density matrices stemmed from the fact that we did not know how to describe the state of qubit 1 in the Bell pair $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. We claimed that the answer lies in the framework of density matrices, but we have not yet prescribed how one can *use* density matrices to describe the state of qubit 1 of $|\Phi^+\rangle$. We now do this via the *partial trace* operation.

Intuitively, given a bipartite density matrix ρ_{AB} on systems A and B, the partial trace operation $\mathrm{Tr}_B(\rho_{AB})$ returns a density matrix on system A alone (analogously, $\mathrm{Tr}_A(\rho_{AB})$ returns a density matrix on B alone). Formally, we have that

$$\mathrm{Tr}_B : \mathcal{L}(\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}) \mapsto \mathcal{L}(\mathbb{C}^{d_1}).$$

How exactly is Tr_B defined? Recall that the trace of a matrix $\rho \in \mathcal{L}(\mathbb{C}^d)$ is defined as

$$\mathrm{Tr}(\rho) = \sum_i \rho(i,i) = \sum_{i=1}^{d} \langle i|\rho|i\rangle.$$

For the *partial* trace, we copy this idea, except we only apply it to one subsystem as follows: For $\rho \in \mathcal{L}(\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2})$,

$$\mathrm{Tr}_B(\rho) = \sum_{i=1}^{d_2} (I_A \otimes \langle i|)\rho(I_A \otimes |i\rangle).$$

In other words, we leave system A untouched (hence the I_A terms above), and "trace out" system B. Note that like the trace, the partial trace is a linear map. Let us practice this operation on a number of states, and in the process draw our desired connection to the Bell state which we started this lecture with.

Example 1: Product states. Suppose first that ρ is the density matrix of a pure product state $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$. In other words,

$$\rho = (|\psi_1\rangle \otimes |\psi_2\rangle)(\langle\psi_1| \otimes \langle\psi_2|) = |\psi_1\rangle\langle\psi_1|_A \otimes |\psi_2\rangle\langle\psi_2|_B.$$

Recall that earlier we said the state of $|\psi\rangle$ on qubit 1 is exactly $|\psi_1\rangle$. Let us confirm this by applying the partial trace to ρ to trace out subsystem B:

$$\begin{aligned}
\mathrm{Tr}_B(\rho) &= \mathrm{Tr}_B(|\psi_1\rangle\langle\psi_1| \otimes |\psi_2\rangle\langle\psi_2|) \\
&= \sum_{i=1}^{d} (I \otimes \langle i|)|\psi_1\rangle\langle\psi_1| \otimes |\psi_2\rangle\langle\psi_2|(I \otimes |i\rangle) \\
&= \sum_{i=1}^{d} |\psi_1\rangle\langle\psi_1| \otimes \langle i|\psi_2\rangle\langle\psi_2|i\rangle \\
&= |\psi_1\rangle\langle\psi_1| \left(\sum_{i=1}^{d} \langle i|\psi_2\rangle\langle\psi_2|i\rangle \right) \\
&= |\psi_1\rangle\langle\psi_1|\mathrm{Tr}(|\psi_2\rangle\langle\psi_2|) \\
&= |\psi_1\rangle\langle\psi_1|,
\end{aligned}$$

where the second equality applies the definition of the partial trace, the second-last equality uses the definition of the trace, and the last equality uses the cyclic property of the trace. In

other words, the state of qubit 1 is $|\psi_1\rangle$, as claimed! Note that the calculation above did not use the fact that $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ is pure — indeed, the same calculation yields that for any operator $\rho = \rho_1 \otimes \rho_2$,

$$\mathrm{Tr}_B(\rho) = \rho_1 \cdot \mathrm{Tr}(\rho_2). \tag{4.2}$$

Exercise 4.7. Show that for any $\rho = \rho_1 \otimes \rho_2$, $\mathrm{Tr}_A(\rho) = \mathrm{Tr}(\rho_1) \cdot \rho_2$. (Hint: Follow the chain of equalities in the calculation above.)

Example 2: Separable states. We have said that a pure state $|\psi\rangle \in \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$ is not entangled, or *separable*, if and only if $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$ for some $|\psi_1\rangle \in \mathbb{C}^{d_1}$ and $|\psi_2\rangle \in \mathbb{C}^{d_2}$. This idea extends to the setting of mixed states as follows: A bipartite density matrix $\rho \in \mathcal{L}(\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2})$ is unentangled or *separable* if

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

for some (possibly non-orthogonal) sets of vectors $\{|\psi_i\rangle\} \subseteq \mathbb{C}^{d_1}$ and $\{|\phi_i\rangle\} \subseteq \mathbb{C}^{d_2}$, and where the $\{p_i\}$ form a probability distribution. In other words, ρ is a probabilistic mixture of pure product states. An example of a separable state is

$$\rho = \frac{1}{2}|0\rangle\langle 0| \otimes |0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| \otimes |1\rangle\langle 1|. \tag{4.3}$$

Since the partial trace is a linear map, and since we know that $\mathrm{Tr}_B(\rho_1 \otimes \rho_2) = \rho_1 \cdot \mathrm{Tr}(\rho_2) = \rho_1$ for density matrices ρ_1, ρ_2, computing the partial trace of ρ for separable states is simple:

$$\mathrm{Tr}_B\left(\sum_i p_i |\psi_i\rangle\langle\psi_i| \otimes |\phi_i\rangle\langle\phi_i|\right) = \sum_i p_i \mathrm{Tr}_B\left(|\psi_i\rangle\langle\psi_i| \otimes |\phi_i\rangle\langle\phi_i|\right) = \sum_i p_i |\psi_i\rangle\langle\psi_i| \cdot \mathrm{Tr}(|\phi_i\rangle\langle\phi_i|) = \sum_i p_i |\psi_i\rangle\langle\psi_i|.$$

Exercise 4.8. What is $\mathrm{Tr}_B(\rho)$ for ρ from Equation (4.3)?

Let us make one further comment about mixed separable states. We saw earlier that it is "easy" to check if a bipartite pure state $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ is entangled by determining its Schmidt rank (we did not explicitly state this, but the Schmidt rank can be calculated in time polynomial in the dimension d). In stark contrast, it turns out that determining whether a *mixed* state $\rho \in \mathcal{L}(\mathbb{C}^d \otimes \mathbb{C}^d)$ is separable is NP-hard! This is another strong reminder that the settings of pure versus mixed states are indeed very different; the latter is typically much more difficult to work with.

Example 3: Pure entangled states. We finally arrive at the case we started this lecture with: Pure entangled states such as $|\Phi^+\rangle$. To compute the partial trace of any pure bipartite $|\psi\rangle$, one approach is to simply write out $|\psi\rangle$ in the standard basis, take its density matrix $\rho = |\psi\rangle\langle\psi|$,

and then compute $\mathrm{Tr}_B(\rho)$. For example, for $|\Phi^+\rangle$, we have

$$\begin{aligned}
\mathrm{Tr}_B(|\Phi^+\rangle\langle\Phi^+|) &= \frac{1}{2}\mathrm{Tr}_B((|00\rangle + |11\rangle)(\langle 00| + \langle 11|) \\
&= \frac{1}{2}\mathrm{Tr}_B(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|) \\
&= \frac{1}{2}\mathrm{Tr}_B(|00\rangle\langle 00|) + \frac{1}{2}\mathrm{Tr}_B(|00\rangle\langle 11|) + \frac{1}{2}\mathrm{Tr}_B(|11\rangle\langle 00|) + \frac{1}{2}\mathrm{Tr}_B(|11\rangle\langle 11|) \\
&= \frac{1}{2}\mathrm{Tr}_B(|0\rangle\langle 0| \otimes |0\rangle\langle 0|) + \frac{1}{2}\mathrm{Tr}_B(|0\rangle\langle 1| \otimes |0\rangle\langle 1|) + \frac{1}{2}\mathrm{Tr}_B(|1\rangle\langle 0| \otimes |1\rangle\langle 0|) + \frac{1}{2}\mathrm{Tr}_B(|1\rangle\langle 1| \otimes |1\rangle\langle \\
&= \frac{1}{2}|0\rangle\langle 0|\mathrm{Tr}(|0\rangle\langle 0|) + \frac{1}{2}|0\rangle\langle 1|\mathrm{Tr}(|0\rangle\langle 1|) + \frac{1}{2}|1\rangle\langle 0|\mathrm{Tr}(|1\rangle\langle 0|) + \frac{1}{2}|1\rangle\langle 1|\mathrm{Tr}(|1\rangle\langle 1|) \\
&= \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| \\
&= \frac{1}{2}I,
\end{aligned}$$

where the third equality follows from the linearity of the partial trace, the fifth by Equation (4.2), and the sixth by the cyclic property of the trace and since $|0\rangle$ and $|1\rangle$ are orthogonal. Thus, the reduced state on qubit 1 for the Bell state is *maximally mixed*! In other words, it is a completely random state about which we have *zero* information. In a similar fashion, one can show that $\mathrm{Tr}_A(|\Phi^+\rangle\langle\Phi^+|) = I/2$.

Exercise 4.9. Show that $\mathrm{Tr}_A(|\Phi^+\rangle\langle\Phi^+|) = I/2$.

And here we arrive at one of the most confounding aspects of quantum mechanics: As in the case of the Bell state, it is possible for us to know *absolutely nothing* about the states of qubits 1 and 2 individually (e.g. they have reduced states $I/2$), but when we bring both qubits together, we know *everything* about their joint state (e.g. $|\Phi^+\rangle$ is a pure state, i.e. there is no uncertainty). This is yet another striking example of a quantum phenomenon which does not occur in the classical world.

4.3.2 Using partial trace to detect entanglement

Recall that in Section 4.2, we said that an arbitrary bipartite *pure* state $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ can be written in terms of its Schmidt decomposition

$$|\psi\rangle = \sum_i \alpha_i |a_i\rangle|b_i\rangle,$$

where $\alpha_i \geq 0$ and $\{|a_i\rangle\}$ and $\{|b_i\rangle\}$ are orthonormal sets. We also said that $|\psi\rangle$ is entangled if and only if its Schmidt rank (i.e. number of non-zero Schmidt coefficients α_i) is at least two. However, we did not discuss how to actually *find* the Schmidt decomposition of $|\psi\rangle$, i.e. we have not specified a procedure for computing the Schmidt rank of $|\psi\rangle$. The partial trace allows us to solve this problem.

Exercise 4.10. Prove that the Schmidt rank of $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ equals the rank of $\rho_A = \mathrm{Tr}_B(|\psi\rangle\langle\psi|)$. You should use the fact that in the definition of the partial trace, the standard basis $\{|i\rangle\}$ on B can be replaced by an arbitrary orthonormal basis, $\{|\psi_i\rangle\}$, on B. (Hint: Choose $\{|\psi_i\rangle\}$ as the Schmidt basis for $|\psi\rangle$ on system B.)

The exercise above immediately gives us a method for determining whether a bipartite pure state $|\psi\rangle$ is entangled — namely, compute $\rho_A = \mathrm{Tr}_B(|\psi\rangle\langle\psi|)$, and check if $\mathrm{rank}(\rho_A) > 1$. If so, $|\psi\rangle$ is entangled; otherwise, it is a product state. Finally, observe that this trick only works for *pure* states $|\psi\rangle$ — indeed, for mixed bipartite ρ_{AB}, detecting entanglement is NP-hard, so such a simple criterion for entanglement is highly unlikely to exist!

4.3.3 How the postulates of quantum mechanics apply to density operators

In previous lectures, we introduced the postulates of quantum mechanics in the context of pure states $|\psi\rangle$. All four postulates extend naturally to the setting of mixed quantum states. For example, for Postulate 3 on composite quantum systems, given density operators ρ and σ, we have that $\rho \otimes \sigma$ is also a valid density operator. This follows from the facts that if A and B are positive semi-definite, then so is $A \otimes B$, and that $\mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A)\mathrm{Tr}(B)$. For now, we will avoid revisiting the postulates in the mixed state setting, and rather approach the topic once the need arises in specific applications of future lectures.

5 Non-Local Games

"I don't demand that a theory correspond to reality because I don't know what it is. Reality is not a quality you can test with litmus paper. All I'm concerned with is that the theory should predict the results of measurements."
— Stephen Hawking

5.1 Background

In the past two lectures, we have discussed measurements and entanglement. We now combine these two topics to discuss one of the most fundamental questions in quantum theory: Does entanglement as we understand it actually exist in Nature? Recall that in 1935, Einstein, Podolsky and Rosen (EPR) argued that quantum mechanics could not be a complete physical theory due to its prediction of states such as the Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. For example, as we will discuss shortly, the Bell state *appears* to allow superluminal (i.e. faster than the speed of light) communication, which is impossible by the theory of relativity. Thus, EPR suggested that there must be more to Nature than what quantum theory prescribes — namely, that there must be " hidden variables" which contain extra information missing in quantum mechanics about what is actually happening in the subatomic world. In other words, the moon *is* there even if you do not look at it — it's just that we "do not have access" to the hidden variables prescribing the state of the moon. Moreover, this information must be "local", in the sense that instantaneous communication between variables should not be possible. Such theories are hence called *local hidden variable* theories.

Remarkably, in 1964, John Bell proved that no local hidden variable theory could ever reproduce the statistics predicted by quantum mechanics! Thus, either local hidden variables theories are wrong, or quantum mechanics is wrong. In fact, Bell went a step further — he proposed a "simple" experiment which could be run in a lab to test which of these two cases represents reality. This experiment was based on (what is now called) a *Bell inequality*, and was run for example by the celebrated effort of Aspect, Grangier, and Roger in 1981, who wrote about their findings:

Our results, in excellent agreement with the quantum mechanical predictions, strongly violate the generalized Bell's inequalities, and rule out the whole class of realistic local theories."
— Aspect, Grangier, Roger, PRL 1981.

Thus, quantum mechanics, and not a local hidden variable theory, appears to be correct, and Bell states as we understand them really do exist. [1]

[1] For completeness, however, it should be mentioned that these experiments are not iron-clad results — it turns out there are conceivable "loopholes" in such experimental setups which could explain the outcomes of such experiments *without* ruling out local hidden variable theories. Nevertheless, over the years improved experimental setups have managed to close some of these "loopholes", and it is probably fair to say that the general physics community regards this at least as strong evidence that local hidden variable theories are insufficient to model the subatomic world.

The aim of this lecture is to discuss an equivalent, but more "computer science-oriented" version of Bell inequalities known as *non-local games*. In the process, we will further practice working with measurements, and see another instance of how the non-local nature of entanglement can be harnessed in a computational task to perform a classically impossible feat.

5.2 Does entanglement allow superluminal signalling?

Before discussing what entanglement *can* do, however, let us take a detour to point out what entanglement *cannot* do. Consider the Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Suppose you take the first qubit of $|\Phi^+\rangle$ and run off to Pluto, and your friend Bob keeps the second qubit here on Earth. If you now measure your qubit in the standard basis $\{|0\rangle, |1\rangle\}$, you will obtain outcome $|i\rangle$ with probability

$$\begin{aligned}\Pr(\text{outcome } i) &= \mathrm{Tr}((|i\rangle\langle i| \otimes I)|\Phi^+\rangle\langle\Phi^+|) \\ &= \frac{1}{2}\mathrm{Tr}((|i\rangle\langle i| \otimes I)(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|)) \\ &= \frac{1}{2}.\end{aligned}$$

Moreover, having obtained outcome $|i\rangle$, you know Bob's qubit on Earth has instantly collapsed to state $|i\rangle$ as well, since your joint postmeasurement state (according to *your* state of knowledge, not Bob's!) is given by (the normalized version of)

$$(|i\rangle\langle i| \otimes I)|\Phi^+\rangle = (|i\rangle\langle i| \otimes I)|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|i\rangle\langle i| \otimes I)(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}|i\rangle|i\rangle.$$

Does this mean you have managed to *instantly* communicate the value of i to Bob? This would be a big no-no according to the theory of relativity!

To resolve this paradox, we employ the density operator framework. First, observe that Bob's state of knowledge before your measurement is given by the reduced density matrix of his qubit, which recall from last class satisfies

$$\rho_B = \mathrm{Tr}_A(|\Phi^+\rangle\langle\Phi^+|) = \frac{1}{2}I.$$

In other words, Bob's qubit contains no information by itself. Now let us compute Bob's reduced state after you've performed your measurement (assuming you have not communicated your result to Bob by some classical means such as via satellite phone). Since with probability $1/2$, you obtain measurement outcome $|i\rangle$, resulting in joint state $|ii\rangle$, the density operator describing your joint post-measurement state is

$$\sigma_{AB} = \frac{1}{2}|00\rangle\langle 00| + \frac{1}{2}|11\rangle\langle 11|.$$

If we now check Bob's reduced state, we find that again

$$\mathrm{Tr}_A(\sigma_{AB}) = \frac{1}{2}\mathrm{Tr}_A(|00\rangle\langle 00| + |11\rangle\langle 11|) = \frac{1}{2}(\mathrm{Tr}_A(|00\rangle\langle 00|) + \mathrm{Tr}_A(|11\rangle\langle 11|)) = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) = \frac{1}{2}I.$$

Thus, Bob has learned *nothing* about the outcome of your measurement! In other words, although Bob's state indeed changes instantly once you measure your qubit (i.e. a collapse occurs), he has no way of knowing it until classical information encoding the measurement outcome is sent from Pluto to Earth. Thus, no information is instantly transmitted.

5.3 Non-local games

As we just saw in Section 5.2, measuring half of a Bell state does not allow one to communicate information from Alice to Bob. Surprisingly, though, the story has only just begun — one can nevertheless use local measurements on both halves of a Bell state to generate a probability distribution which contains correlations stronger than those possible classically. It is precisely this principle which is harnessed in *non-local games*.

The setup of a non-local game is as follows. Suppose you and a friend are charged with doing too much homework on school property, and taken to police headquarters, where you are separated into two different rooms (i.e. you cannot communicate with one another). An interrogator takes turns asking each of you a question, and subsequently compares and cross-checks your answers to verify that you are both telling the truth. It turns out that if you and your friend happened to share two halves of a Bell state before being taken to the police station, then by performing appropriate local measurement on your qubits based on the interrogator's questions, you can sometimes convince the interrogator of your honesty with higher probability than you could hope to do so without the use of entanglement. Let us demonstrate this via the concrete example of the CHSH game.

5.3.1 The CHSH game

In 1969, Clauser-Horne-Shimony-Holt gave an important variant of Bell's inequality, nowadays named after its authors as the CHSH inequality. It turns out that this inequality can be recast as a non-local game, called the *CHSH game*. To specify this game, let us denote the two spatially separated players as Alice and Bob, each of which receives a single question consisting of a single bit, $q_A \in \{0,1\}$ for Alice and $q_B \in \{0,1\}$ for Bob. The questions are chosen uniformly at random. Alice and Bob now output a response consisting of a single bit each, $r_A \in \{0,1\}$ for Alice and $r_B \in \{0,1\}$ for Bob. We say they *win* if $q_A \wedge q_B = r_A \oplus r_B$, where $\wedge$ denotes the AND function and $\oplus$ denotes XOR (recall XOR on two bits $b_1, b_2 \in \{0,1\}$ equals one if and only if precisely one of b_1 and b_2 is set to 1). In words, this says the following: If the questions are $q_A q_B \in \{00, 01, 10\}$, then Alice and Bob should answer with the same bit. Only if the questions are $q_A q_B = 11$ should they answer with different bits.

Exercise 5.1. Suppose the questions satify $q_A = 0$ and $q_B = 1$. What possible pairs of answers (r_A, r_B) allow Alice and Bob to win? What if we have $q_A = q_B = 1$?

Limits on classical strategies

Let us convince ourselves that any deterministic classical strategy for this game can win with probability at most 3/4. Specifically, any deterministic strategy for Alice (similarly for Bob) falls into one of four cases, specifying what Alice does when she receives her question, q_A: Output the same bit as q_A, output the opposite of q_A, ignore q_A and always output 0, or ignore q_A and always output 1. Let us consider one such strategy; the analysis of all other cases follows analogously. Suppose Alice and Bob always ignore their questions q_A and q_B and output $r_A = r_B = 0$. What is the probability of this strategy to win? Since $r_A = r_B = 0$, we have $r_A \oplus r_B = 0$, meaning they win whenever $q_A \wedge q_B = 0$. This happens so long as $q_A q_B \in \{00, 01, 10\}$, i.e. for 3 of the 4 possible pairs of questions they can receive. Since the questions are chosen uniformly at random, they win with probability 3/4, as claimed.

Exercise 5.2. Suppose Alice and Bob choose to always output the same bit they receive, i.e. $r_A = q_A$ and $r_B = q_B$. What is the probability of this strategy to win?

Finally, you may wonder whether Alice and Bob can classically do better if they use a *randomized* strategy for selecting answers. It turns out the answer is no, as a randomized strategy may be viewed as an *average* over all deterministic strategies. Thus, they might as well choose the optimal deterministic strategy.

A quantum strategy

We now show that if Alice and Bob share the Bell pair $|\Phi^+\rangle$ before the game starts, then by performing appropriate local measurements based their questions q_A and q_B, they can win the game with a higher probability of $\cos^2(\pi/8) \approx 0.854$ (compared to 3/4 in the classical case). This is equivalent to saying the original CHSH inequality is *violated* by quantum mechanics, meaning a local hidden variable theory cannot explain the measurement statistics predicted by quantum mechanics (though we will not discuss these details here). To model Alice and Bob's strategy, we introduce the concept of an *observable*.

Observables. Recall that a projective measurement is given by a set of projectors $M = \{\Pi_i\}$ such that $\sum_i \Pi_i = I$. To each outcome we associate some label, $i \in S \subset \mathbb{R}$ for some set S such as $S = \{1, \ldots, d\}$. An *observable* is simply the matrix

$$C = \sum_{i \in S} i \cdot \Pi_i.$$

For example, consider a measurement in the standard basis $\{|0\rangle, |1\rangle\}$, where the outcomes are labelled with set $S = \{1, -1\}$, respectively. Then, the corresponding observable is (for Pauli operator Z)

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

Exercise 5.3. Consider measurement $M = \{|+\rangle\langle +|, |-\rangle\langle -|\}$ with outcome labels $S = \{1, -1\}$. What is the corresponding observable?

Observables are useful in that they allow us to quickly calculate the *expected value* of a measurement. Recall that for a random variable X distributed over set Y, the expected value of X is defined

$$E[X] = \sum_{y \in Y} \Pr(X = y) \cdot y,$$

and captures what value X takes on *average*. Similarly, the outcome of a measurement $M = \{\Pi_i\}$ can be modelled by a random variable X over possible outcome labels $i \in S$, and we can ask what $E[X]$ is. For a measurement on density operator ρ, this is given by the formula

$$E[X] = \sum_{i \in S} \Pr(X = i) \cdot i = \sum_{i \in S} \mathrm{Tr}(\rho \Pi_i) \cdot i = \mathrm{Tr}\left(\rho \left(\sum_i \Pi_i \cdot i\right)\right) = \mathrm{Tr}\left(\rho C\right)$$

for observable $C = \sum_i i \cdot \Pi_i$, and where the third equality follows by linearity of the trace.

Exercise 5.4. Suppose we measure in the standard basis $\{|0\rangle, |1\rangle\} \subseteq \mathbb{C}^2$ with corresponding measurement labels $\{1, -1\}$, respectively. Convince yourself that the corresponding observable is Pauli Z. What is the expected value of measuring density operator $\rho = \frac{2}{3}|0\rangle\langle 0| + \frac{1}{3}|+\rangle\langle +|$ with observable Z?

Alice and Bob's strategy. With observables in hand, we state Alice's and Bob's strategy. First, let us change the encoding of their output bits. Namely, instead of outputting $r_A, r_B \in \{0, 1\}$, in their measurements Alice and Bob use label 1 to mean output 0, and label -1 to mean output bit 1. Then, conditioned on questions q_A and q_B, Alice and Bob use observables A_{q_A} and B_{q_B} as follows:

$$A_0 = Z \qquad A_1 = X \qquad B_0 = H \qquad B_1 = ZHZ, \tag{5.1}$$

for H the Hadamard gate. Note that all four of these observables have eigenvalues in set $S = \{1, -1\}$, and so they can be thought of as measurements in their respective eigenbases with outcomes labelled by S.

Exercise 5.5. Intuitively, which bases do Alice's two measurements A_0 and A_1 correspond to?

Calculating the success probability. At first glance, it is likely unclear why such a strategy should be interesting at all. Let us first calculate the success probability of this strategy to demonstrate that it *does* work, and subsequently give an intuitive understanding of *why* it works.

First, we claim that for arbitrary observables with spectral decompositions $A = |a_0\rangle\langle a_0| - |a_1\rangle\langle a_1|$ and $B = |b_0\rangle\langle b_0| - |b_1\rangle\langle b_1|$, the quantity $\mathrm{Tr}(A \otimes B|\Phi^+\rangle\langle\Phi^+|)$ encodes the probability that Alice and Bob output the *same* bits minus the probability they output *different* bits, assuming they measure using A and B. To see this, we have

$$\begin{aligned}
\mathrm{Tr}(A \otimes B|\Phi^+\rangle\langle\Phi^+|) &= \mathrm{Tr}((|a_0\rangle\langle a_0| - |a_1\rangle\langle a_1|) \otimes (|b_0\rangle\langle b_0| - |b_1\rangle\langle b_1|)|\Phi^+\rangle\langle\Phi^+|) \\
&= \mathrm{Tr}(|a_0\rangle\langle a_0| \otimes |b_0\rangle\langle b_0| \cdot |\Phi^+\rangle\langle\Phi^+|) - \mathrm{Tr}(|a_0\rangle\langle a_0| \otimes |b_1\rangle\langle b_1| \cdot |\Phi^+\rangle\langle\Phi^+|) - \\
&\quad \mathrm{Tr}(|a_1\rangle\langle a_1| \otimes |b_0\rangle\langle b_0| \cdot |\Phi^+\rangle\langle\Phi^+|) + \mathrm{Tr}(|a_1\rangle\langle a_1| \otimes |b_1\rangle\langle b_1| \cdot |\Phi^+\rangle\langle\Phi^+|) \\
&= \Pr(\text{output } 00) - \Pr(\text{output } 01) - \Pr(\text{output } 10) + \Pr(\text{output } 11) \\
&= \Pr(\text{output same bits}) - \Pr(\text{output different bits}).
\end{aligned}$$

We conclude that for pairs of questions $q_A q_B \in \{00, 01, 10\}$ (i.e. Alice and Bob should output the same bit), the term $\mathrm{Tr}(A \otimes B|\Phi^+\rangle\langle\Phi^+|)$ denotes the probability Alice and Bob win minus the probability they lose. Similarly, for the question pair $q_A q_B = 11$ (i.e. their answers should *disagree*) the analogous quantity is $-\mathrm{Tr}(A \otimes B|\Phi^+\rangle\langle\Phi^+|)$. It follows that that the probability that Alice and Bob win minus the probability they lose is given by

$$\begin{aligned}
&\Pr(q_A q_B = 00) \cdot \mathrm{Tr}((A_0 \otimes B_0)|\Phi^+\rangle\langle\Phi^+|) + \Pr(q_A q_B = 01) \cdot \mathrm{Tr}((A_0 \otimes B_1)|\Phi^+\rangle\langle\Phi^+|) + \\
&\Pr(q_A q_B = 10) \cdot \mathrm{Tr}((A_1 \otimes B_0)|\Phi^+\rangle\langle\Phi^+|) - \Pr(q_A q_B = 11) \cdot \mathrm{Tr}((A_1 \otimes B_1)|\Phi^+\rangle\langle\Phi^+|) \\
= \quad &\frac{1}{4}\left[\mathrm{Tr}((A_0 \otimes B_0 + A_0 \otimes B_1 + A_1 \otimes B_0 - A_1 \otimes B_1)|\Phi^+\rangle\langle\Phi^+|)\right],
\end{aligned} \tag{5.2}$$

where the factor of $1/4$ appears because each question pair $q_A q_B$ appears with probability $1/4$. A direct calculation now yields that for our choices of A_i and B_i,

$$\mathrm{Tr}((A_0 \otimes B_0)|\Phi^+\rangle\langle\Phi^+|) = \mathrm{Tr}((A_0 \otimes B_1)|\Phi^+\rangle\langle\Phi^+|) = \mathrm{Tr}((A_1 \otimes B_0)|\Phi^+\rangle\langle\Phi^+|) = \frac{1}{\sqrt{2}}, \tag{5.3}$$

whereas $\mathrm{Tr}((A_1 \otimes B_1)|\Phi^+\rangle\langle\Phi^+|) = -1/\sqrt{2}$.

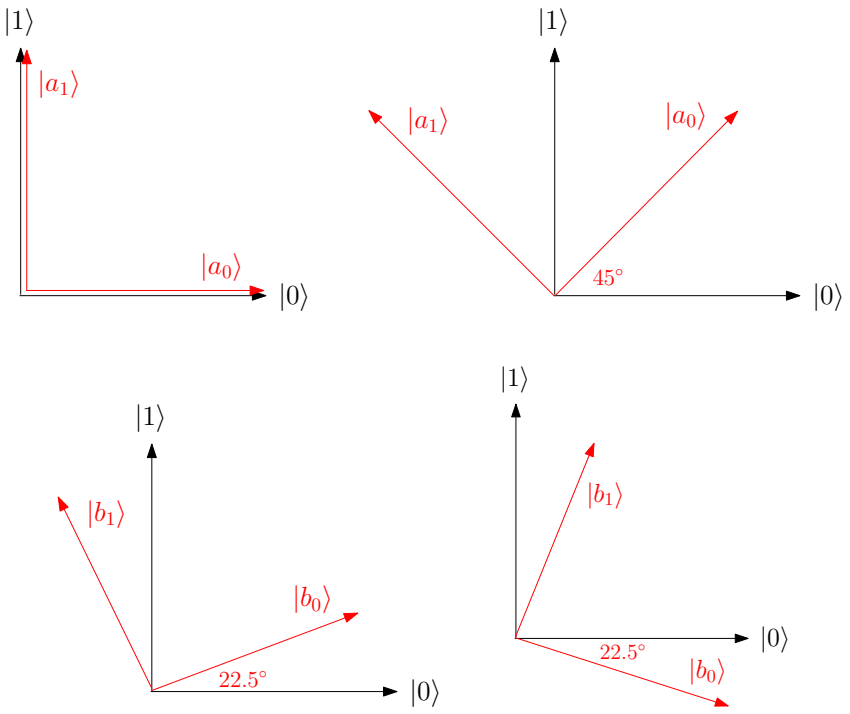


Figure 5.1: Measurement bases for A_0 (top-left), A_1 (top-right), B_0 (bottom-left), B_1 (bottom-right).

Exercise 5.6. Verify that $\mathrm{Tr}((A_0 \otimes B_0)|\Phi^+\rangle\langle\Phi^+|) = 1/\sqrt{2}$ and $\mathrm{Tr}((A_1 \otimes B_1)|\Phi^+\rangle\langle\Phi^+|) = -1/\sqrt{2}$ for A_i and B_i as chosen in Equation (5.1).

Let p be the probability with which Alice and Bob win with this strategy. Equations (5.2) and 5.3 tell us that the probability of winning minus losing, $p - (1-p) = 2p - 1$, equals $1/\sqrt{2}$. Hence, $p = 1/2 + 1/(2\sqrt{2}) = \cos^2(\pi/8) \approx 0.854$, which is strictly better than the optimal classical winning probability of 0.75, as claimed.

Intuition behind Alice and Bob's strategy. Why does this strategy work? To see this, we plot the eigenbases for each observable used (i.e. the measurement bases) in Figure 5.6. Each eigenvector is denoted $|a_0\rangle$ or $|a_1\rangle$ for Alice ($|b_0\rangle$ or $|b_1\rangle$ for Bob), corresponding to Alice outputting 0 or 1, respectively. To begin, intuitively, the Bell state has the special property that if Alice and Bob measure in "similar" bases on their respective qubits, then they should get the same outcome with high probability, whereas if they measure with "very different" bases, then they will get opposite outcomes with high probability. In other words, we want to choose observables A_0, A_1, B_0, and B_1 such that for questions $q_A q_B \in \{00, 01, 10\}$, Alice and Bob measure in bases which are "close" (since their output bits should match), and for $q_A q_B = 11$, they measure in bases which are "far apart" (since their outputs should differ). Figure 5.6 depicts exactly this scenario. For example, for $q_A q_B = 00$, the plots depicting Alice and Bob's measurement bases are the top-left and bottom-left, respectively. Here we see that $|a_0\rangle$ for Alice has high overlap with $|b_0\rangle$ for Bob (similarly for $|a_1\rangle$ and $|b_1\rangle$), so they are likely to obtain the same outcome. Conversely, for $q_A q_B = 11$ (i.e. top-right and bottom-right plots), $|a_0\rangle$ and $|b_0\rangle$ are almost orthogonal (similarly for $|a_1\rangle$ and $|b_1\rangle$), meaning it is unlikely that Alice and Bob's outputs will

agree.

5.3.2 The magic square game

In Section 5.3.1, we studied the CHSH game, whose maximum classical winning probability is 3/4, whereas a quantum strategy based on entanglement achieves probability 0.854. Let us see if this separation can be made stronger — does there exist a non-local game with classical winning probability strictly smaller than 1, but can be won *perfectly* quantumly? It turns out the answer is yes; an example demonstrating this phenomenon is the *magic square* of Mermin and Peres from the early 1990's.

The magic square works as follows: Suppose we have a 3×3 array, labelled with variables x_{ij}:

x_{11}	x_{12}	x_{13}
x_{21}	x_{22}	x_{23}
x_{31}	x_{32}	x_{33}

.

The goal is to assign 0 or 1 to each variable x_{ij} so that in each row and in the first two columns, the number of 1's is even (i.e. they have parity 0). In the last column, the requirement is that the number of 1's is odd (i.e. they have parity 1). Formally, we can express this as the system of equations

$$x_{11} \oplus x_{12} \oplus x_{13} = 0 \qquad x_{11} \oplus x_{21} \oplus x_{31} = 0 \tag{5.4}$$

$$x_{21} \oplus x_{22} \oplus x_{23} = 0 \qquad x_{12} \oplus x_{22} \oplus x_{32} = 0 \tag{5.5}$$

$$x_{31} \oplus x_{32} \oplus x_{33} = 0 \qquad x_{13} \oplus x_{23} \oplus x_{33} = 1. \tag{5.6}$$

where $\oplus$ denotes the XOR, i.e. addition modulo 2. It turns out that it is *impossible* to fill out this square so that all six of these equations are satisfied.

Exercise 5.7. Show that there do not exist assignments to all $x_{ij} \in \{0, 1\}$ satisfying all six equations above. (Hint: Add all the equations to achieve a contradiction.)

Making a game of the magic square. We can turn the magic square into a non-local game between parties Alice and Bob as follows. As her question, Alice receives uniformly at random one of the rows or columns of the square, where we index the rows and columns by $q_A \in \{1, 2, 3, 4, 5, 6\}$. Suppose row/column q_A contains variables x, y, z. Then, Alice must return assignments for x, y, and z — denote her answers x_A, y_A, z_A. As for Bob's question, we now choose uniformly one of x, y, z, and ask Bob to provide an assignment for it. For example, Bob might be asked to provide an assignment for y, which we would denote y_B. We say Alice and Bob win the game if x_A, y_A, z_A satisfies the parity constraint for the corresponding row or column, *and* if Bob's answer matches that of Alice (i.e. $y_B = y_A$ in our example).

Exercise 5.8. Suppose Alice is asked to provide an assignment for row 1, i.e. x_{11}, x_{12}, x_{13}, and Bob is asked to provide an assignment for x_{13}. What response of Alice and Bob can win the game? What response will lose the game?

Because the magic square cannot be filled out perfectly (i.e. Equations (5.4), (5.5), (5.6) cannot be simultaneously satisfied), it is not difficult to argue that no classical strategy of Alice and Bob can win this non-local game with probability 1. There is, however, a perfect quantum strategy, and as for the CHSH game, it relies on the non-local nature of entanglement.

A perfect quantum strategy. As for the CHSH game, we begin by switching from basis $\{0,1\}$ for encoding responses to $\{1,-1\}$, respectively. (As a result, our observables will have eigevalues in $\{1,-1\}$.) Note that this translates an expression of the form $x \oplus y \oplus z = b$ for $x, y, z, b \in \{0,1\}$ to $xyz = (-1)^b$ for $x, y, z \in \{1,-1\}$, i.e. the direct sum condition converts to a multiplication. For example, with this "change of basis", $0 \oplus 1 \oplus 0 = 1$ converts to $(1)(-1)(1) = (-1)^1$. Also as for CHSH, before the game starts, Alice and Bob share an entangled state between them; in this case, we use a higher dimensional analogue of the Bell state,

$$|\psi\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|11\rangle + \frac{1}{2}|22\rangle + \frac{1}{2}|33\rangle \in \mathbb{C}^4 \otimes \mathbb{C}^4,$$

where recall $\{|0\rangle, |1\rangle, |2\rangle, |3\rangle\}$ is an orthonormal basis for $\mathbb{C}^4$. Now recall Alice will receive three cells x, y, z in the magic square (from a single row or column) for which she has to provide values in $\{1,-1\}$, and Bob will receive one of x, y, or z, for which he must provide a value in $\{1,-1\}$. The following chart shows which measurement Alice or Bob should apply to their half of $|\psi\rangle$ depending on which cell(s) they receive questions for:

$Z \otimes I$	$I \otimes Z$	$Z \otimes Z$
$I \otimes X$	$X \otimes I$	$X \otimes X$
$Z \otimes X$	$X \otimes Z$	$Y \otimes Y$

(5.7)

For example, if Alice receives row 1 as her question, she measures observables $Z \otimes I$, $I \otimes Z$, and $Z \otimes Z$ on her half of $|\psi\rangle$. Then, if Bob receives the top middle cell as his question, he measures $I \otimes Z$ on his half of $|\psi\rangle$. Since all observables above have eigenvalues in $\{1,-1\}$, all of Alice and Bob's responses will also be in $\{1,-1\}$.

Why does this work? Suppose Alice gets the first row as a question. Then, she measures according to $Z \otimes I$, $I \otimes Z$, and $Z \otimes Z$, obtaining 3 values from set $\{1,-1\}$, each of which corresponds to one of the cells in row 1. The expected value for the *product* of these (since recall switching to the $\{1,-1\}$ output encoding converted our constraints to multiplication of variables) is given by

$$\mathrm{Tr}((Z \otimes I)(I \otimes Z)(Z \otimes Z)|\psi\rangle\langle\psi|) = \mathrm{Tr}((I \otimes I)|\psi\rangle\langle\psi|) = 1.$$

In other words, Alice *always* outputs a triple from set $\{1,-1\}^{\times 3}$ with product 1 when she is asked row 1, and hence answers her question correctly. (Aside: Note that each operator above, e.g. $Z \otimes I$, acts entirely on Alice's 4-dimensional half of $|\psi\rangle$.) More generally, multiplying out the observables in any row and in the first two columns similarly yields the matrix $I \otimes I$ — this means Alice always outputs values with product 1 in these cases, as desired. As for the last column, the observables multiply out to $-I \otimes I$, meaning Alice outputs values with product -1, again as desired.

Exercise 5.9. Show that multiplying the observables in column 3 above yield $-I \otimes I$.

How about Bob — will his measurement result match Alice's? Note that by definition, Alice and Bob both perform the same measurement for a given cell on their respective halves of $|\psi\rangle$. For example, if Alice is asked row 1 and Bob is asked the top-middle cell, then both of them measure $I \otimes Z$ on their respective halves of $|\psi\rangle$. Here, we use a nice fact:

Fact 5.10. *For $|\psi\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} |ii\rangle$ and any observable A, we have $A \otimes I|\psi\rangle = I \otimes A^T|\psi\rangle$.*

Using this fact, the expected value for the product of Alice and Bob's measurements for the top-middle cell is

$$\mathrm{Tr}((I \otimes Z)_A \otimes (I \otimes Z)_B |\psi\rangle\langle\psi|) = \mathrm{Tr}((I^2 \otimes Z^2)_A \otimes (I \otimes I)_B |\psi\rangle\langle\psi|) = 1,$$

since $Z^2 = I$. In other words, their outputs always agree. A similar analysis applies for any of the nine cells in the magic square. We conclude that Alice and Bob win the magic square game with certainty.

Finally, we mention an important point, but one which we will not dwell on at this point in the course. Recall that in general, measuring a quantum state disturbs it, and so the *order* in which a sequence of measurements is performed is important. There is an exception to this rule, though — if the observables corresponding to the measurements all commute, then the order in which the measurements are performed does not matter. In the strategy above, any pair of observables in the table pairwise commute - thus, e.g., when Alice does three measurements in a row on her half of $|\psi\rangle$, the precise order of the measurements does not matter. This ensures the strategies above are well-defined.

Connections to solving systems of equations. Recall the magic square corresponds to the following inconsistent system of equations (where each $x_{ij} \in \{1, -1\}$):

$$x_{11}x_{12}x_{13} = 1 \qquad x_{11}x_{21}x_{31} = 1 \tag{5.8}$$
$$x_{21}x_{22}x_{23} = 1 \qquad x_{12}x_{22}x_{32} = 1 \tag{5.9}$$
$$x_{31}x_{32}x_{33} = 1 \qquad x_{13}x_{23}x_{33} = -1 \tag{5.10}$$

The reason why the magic square game has a perfect quantum strategy is intuitively that if we allow *higher dimensional assignments*, then the analogous system *does* have a solution! Formally, let us work in $\mathcal{L}(\mathbb{C}^4)$, obtaining system

$$M_{11}M_{12}M_{13} = I \qquad M_{11}M_{21}M_{31} = I \tag{5.11}$$
$$M_{21}M_{22}M_{23} = I \qquad M_{12}M_{22}M_{32} = I \tag{5.12}$$
$$M_{31}M_{32}M_{33} = I \qquad M_{13}M_{23}M_{33} = -I, \tag{5.13}$$

where $M_{ij} \in \mathcal{L}(\mathbb{C}^4)$ and I is the 4×4 identity matrix.

Exercise 5.11. Let M_{ij} be given by the observable in row i and column j of Table 5.7. Show that this choice of assignment satisfies the system in Equations (5.11),(5.12), and (5.13).

It follows immediately from the exercise above that *regardless* of which state $|\psi\rangle$ Alice and Bob share, Alice will always output a correct answer. The only reason we now require $|\psi\rangle$ to specifically be a high dimensional analogue of the Bell state is to apply Fact 5.10, which allows Bob's condition to always be satisfied. Thus, at the heart of the magic square game is the idea that even if a system of equations has no solution over a low dimensional space, its high dimensional analogue may nevertheless have a solution.

5.3.3 Closing thoughts

Although *a priori*, the games considered in this chapter are not obviously interesting in and of themselves, our discussion here has important ramifications. First, as previously mentioned, such non-local games correspond in principle to experiments which can be run in a lab to

demonstrate that, indeed, quantum states can give rise to correlations strictly stronger than those possible with classical states. Alternatively, we can state this as: Assuming experimental loopholes can be closed, Einstein was wrong, and the non-local nature of entanglement appears to, in fact, be real.

Second, in recent years, non-local games such as the CHSH game have been exploited with great success in areas of quantum information research such as device independent cryptography, randomness expansion, and computational complexity theory. The first of these areas, for example, roughly studies the concept of reliable cryptography using quantum devices, *even if those devices may have been tampered with.* At the heart of some of these advances are "rigidity theorems". These state, roughly, that the CHSH game's optimal success probability is *robust* or *rigid* — even if one wants to obtain a success probability for CHSH which is close to optimal (as opposed to exactly optimal), then one must use a strategy which is "close" to the one discussed here (for an appropriate notion of "close").

6 Entropy and Entanglement Distillation

> *"Sea water is rendered potable by evaporation; wine and other liquids can be submitted to the same process, for, after having been converted into vapours, they can be condensed back into liquids."*
> — Aristotle (writing about distillation)

The theme of the last two lectures has been of a quantum information theoretic nature — we have studied cloning (or rather, lack thereof), entanglement, and non-local correlations. Before progressing to our next main theme of quantum algorithms, we now give a brief taste of more advanced ideas in quantum information. In the process, we will continue getting used to working with quantum states in both the state vector and density operator formalisms.

The main questions we ask in this lecture are the following:

- *How can we quantify the "amount" of entanglement in a composite quantum system?*
- *Under what conditions can "less entangled" states be "converted" to "more entangled" states?*

The first question will require the foundational concept of *entropy*, introduced in the context of classical information theory by Claude Shannon in 1948. The entropy is worthy of a lecture in itself, being an extremely important quantity. The second question above is tied to the discovery of *entanglement distillation*, in which, similar to the age-old process of distilling potable water from salt water (or more fittingly for our analogy, "pure" water from "dirty" water), one can "distill" pure entanglement from noisy entanglement.

6.1 Entropy

One of the most influential scientific contributions of the 20th century was the 1948 paper of Claude Shannon, "A Mathematical Theory of Communication", which single-handed founded the field of information theory. Roughly, the aim of information theory is to study information transmission and compression. For this, Shannon introduced the notion of *entropy*, which intuitively quantifies "how random" a data source is, or the "average information content" of the source. It turns out that a *quantum* generalization of entropy will be vital to quantifying entanglement; as such, we begin by defining and motivating the classical *Shannon entropy*.

6.1.1 Shannon entropy

Let X be a discrete random variable taking values from set $\{x_1, \ldots, x_n\}$, where $\Pr(x_i) := \Pr(X = x_i)$ denotes the probability that X takes value x_i. Then, the Shannon entropy $H(X)$ is defined as

$$H(X) = \sum_{i=1}^{n} - \Pr(x_i) \log \left(\Pr(x_i) \right). \tag{6.1}$$

Here, the logarithm is taken base 2, and we define $0 \cdot \log 0 = 0$.

Motivation. Before getting our hands dirty understanding $H(x)$, let us step back and motivate it via data compression. Suppose we have a data source whose output we wish to transmit from (say) Germany to Canada. Naturally, we may wish to first *compress* the data, so that we need to transmit as few bits as possible between the two countries. Furthermore, a compression scheme is useless unless we are later able to *recover* or *uncompress* the data in Canada. This raises the natural question: *How few bits can one transmit, so as to ensure recovery of the data on the receiving end?* Remarkably, Shannon's noiseless coding theorem says that this quantity is given by the entropy! Roughly, the theorem says that in order to reliably transmit N i.i.d. (independently and identically distributed) random variables X_i from a random source X, it is necessary and sufficient to instead send $H(X)$ bits of communication.

Getting our hands dirty. We now explore the sense in which $H(X)$ indeed quantifies the "randomness" or "uncertainty" of X by considering two boundary cases. In the first boundary case, X models a fair coin flip, i.e. X takes value HEADS or TAILS with probability $1/2$ each. Then,

$$H(X) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = \frac{1}{2} + \frac{1}{2} = 1. \tag{6.2}$$

Therefore, we interpret a fair coin as encoding, on average, *one* bit of information. Alternatively, in the information transmission setting, we would need to transmit a single bit to convey the answer of the coin flip from Germany to Canada. This intuitively makes sense — since the outcome of the coin flip is completely random, there is no way to *guess* its outcome in Canada with success probability greater than $1/2$ (i.e. a random guess).

The second boundary case is treated in the exercise below.

Exercise 6.1. Suppose random variable Y models a biased coin, e.g. takes value HEADS and TAILS with probability 1 and 0, respectively. What is $H(Y)$?

In this example, there is no "uncertainty"; we know the outcome will be HEADS. Thus, in the communication setting, one can interpret this as saying *zero* bits of communication are required to transmit the outcome of the coin flip from Germany to Canada (assuming both Germany and Canada know the probabilities of obtaining HEADS and TAILS beforehand). Indeed, the answer to the exercise above is $H(Y) = 0$.

Exercise 6.2. Let random variable Z take values in set $\{0, 1, 2\}$ with probabilities $\{1/4, 1/4, 1/2\}$, respectively. What is $H(Z)$?

Detour: Deriving the formula for entropy. The entropy formula is odd-looking; to understand how it arises, the key observation is the intuition behind the coin flip examples, which says that "when an event is *less* likely to happen, it reveals *more* information". To capture this intuition, Shannon started with a formula for *information content* $I(x_i)$, which for any possible event x_i for random variable X, is given by

$$I(x_i) = \log\left(\frac{1}{\Pr(x_i)}\right) = -\log\left(\Pr(x_i)\right). \tag{6.3}$$

Since the log function is strictly monotonically increasing (i.e. $I(x) > I(y)$ if $x > y$ for $x, y \in (0, \infty)$), it holds that $I(x_i)$ captures the idea that "rare events yield more information". But $I(x)$ also has three other important properties we expect of an "information measure"; here are the first two:

1. (Information is non-negative) $I(x) \geq 0$, and
2. (If an event occurs with certainty, said occurrence conveys no information) if $\Pr(x) = 1$, then $I(x) = 0$.

For the third important property, we ask — why did we use the log function? Why not any other monotonically increasing function satisfying properties (1) and (2) above? Recall that, by definition, two random variables X and Y are *independent* if

$$\Pr(X = x \text{ and } Y = y) = \Pr(X = x) \Pr(Y = y). \tag{6.4}$$

Moreover, if X and Y are independent, then intuitively one expects the information conveyed by the joint event $z := (X = x \text{ and } Y = y)$ to be *additive*, i.e. $I(z) = I(x) + I(y)$. But this is precisely what the information content $I(x)$ satisfies, due to its use of the log function, as you will now verify.

Exercise 6.3. Let X and Y be independent random variables. Then, for $z := (X = x \text{ and } Y = y)$, express $I(z)$ in terms of $I(x)$ and $I(y)$.

We can now use the information content to derive the formula for entropy — $H(X)$ is simply the *expected* information content over all possible events $\{x_1, \ldots, x_n\}$. (Recall here that for random variable X taking values $x \in \{x_i\}$, its expectation $E[x]$ is given by $E[x] = \sum_i \Pr(x_i) \cdot x_i$.) This is precisely why at the start of this section, we referred to $H(x)$ as the *average* information content of a data source.

6.1.2 Von Neumann Entropy

Recall the first aim of this lecture was to use entropy to measure entanglement. For this, we shall require a quantum generalization of the Shannon entropy $H(X)$, denoted the *von Neumann* entropy $S(\rho)$, for density operator ρ. To motivate this definition, let us recall the "hierarchy of matrix classes" we introduced in discussing measurements:

- Hermitian operators, $\text{Herm}\left(\mathbb{C}^d\right)$, which generalize the real numbers.
- Positive semidefinite operators, $\text{Pos}\left(\mathbb{C}^d\right)$, which generalize the non-negative real numbers.
- Orthogonal projection operators, $\Pi(\mathbb{C}^d)$, which generalize the set $\{0, 1\}$.

Note that $\Pi(\mathbb{C}^d) \subseteq \text{Pos}\left(\mathbb{C}^d\right) \subseteq \text{Herm}\left(\mathbb{C}^d\right)$, and that the notion of "generalize" above means the *eigenvalues* of the operators fall into the respective class the operators generalize. (For example, matrices in $\text{Pos}\left(\mathbb{C}^d\right)$ have non-negative eigenvalues.) Applying this same interpretation to the set of *density operators* acting on $\mathbb{C}^d$, $\text{D}\left(\mathbb{C}^d\right)$, we thus have that density operators generalize the notion of a *probability distribution*. Indeed, any probability distribution can be embedded into a diagonal density matrix, as you will now show.

Exercise 6.4. Let $\{p_i\}_{i=1}^d$ denote a probability distribution. Define diagonal matrix $\rho \in \mathcal{L}(\mathbb{C}^d)$ such that $\rho(i, i) = p_i$. Show that ρ is a density matrix.

Since the eigenvalues $\lambda_i(\rho)$ of a density operator $\rho \in \mathrm{D}\left(\mathbb{C}^d\right)$ form a probability distribution, the natural approach for defining a *quantum* entropy formula is to apply the classical Shannon entropy to the spectrum of ρ:

$$S(\rho) := H\left(\{\lambda_i(\rho)\}_{i=1}^d\right) = \sum_{i=1}^d -\lambda_i(\rho)\log\left(\lambda_i(\rho)\right). \tag{6.5}$$

Operator functions. It is important to pause now and take stock of what we have done in defining $S(\rho)$ in Equation (6.5): In order to apply a function $f : \mathbb{R} \mapsto \mathbb{R}$ to a Hermitian matrix $H \in \mathrm{Herm}\left(\mathbb{C}^d\right)$, we instead applied f to the *eigenvalues* of H. Why does this "work"? Let us look at the Taylor series expansion of f, which for e.g. $f = e^x$ is (the series converges for all x)

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots. \tag{6.6}$$

This suggests an idea — to define e^H, perhaps we can substitute H in the right hand side of the Taylor series expansion of e^x:

$$e^H := I + H + \frac{H^2}{2!} + \frac{H^3}{3!} + \cdots. \tag{6.7}$$

Indeed, this leads to our desired definition; that to generalize the function $f(x) = e^x$ to Hermitian matrices, we apply f to the eigenvalues of H, as you will now show.

Exercise 6.5. Let H have spectral decomposition $H = \sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i|$. Show that in Equation (6.7),

$$e^H = \sum_i e^{\lambda_i} |\lambda_i\rangle\langle\lambda_i|.$$

This idea of applying functions $f : \mathbb{R} \mapsto \mathbb{R}$ to the eigenvalues of Hermitian operators is used so frequently in quantum information that we give such "generalized f" a name — *operator functions*. In the case of $S(\rho)$, by setting $f(x) = \log x$, we can rewrite Equation (6.5) as

$$S(\rho) = -\mathrm{Tr}(\rho\log(\rho)). \tag{6.8}$$

Exercise 6.6. Verify that Equations (6.5) and (6.8) are equal.

Exercise 6.7. Let $f(x) = x^2$. What is $f(X)$, for X the Pauli X operator? Why does this yield the same results as multiplying X by itself via matrix multiplication?

Exercise 6.8. Let $f(x) = \sqrt{x}$. For any pure state $|\psi\rangle \in \mathbb{C}^d$, define rank one density operator $\rho = |\psi\rangle\langle\psi|$. What is $\sqrt{\rho}$?

Exercise 6.9. What is $\sqrt{Z}$ for Z the Pauli Z operator? Is it uniquely defined?

Properties of the von Neumann entropy

Let us now see how properties of $H(X)$ carry over to $S(\rho)$. These will prove crucial in our understanding of quantifying entanglement shortly.

1. *When does a quantum state have no entropy?*

   Recall in our biased coin flip example that if an outcome occurs with probability 1 in our distribution, then $H(X) = 0$. Quantumly, the analogue of this statement is that $S(\rho) = 0$ if and only if ρ is a *pure* state, i.e. $\rho = |\psi\rangle\langle\psi|$ for some $|\psi\rangle \in \mathbb{C}^d$. This is because a recall a pure state is the special case of a mixed state in which one of the states in the preparation procedure is picked with certainty.

   Exercise 6.10. Prove that for any pure state $|\psi\rangle$, $S(|\psi\rangle\langle\psi|) = 0$.

2. *When does a quantum state have maximum entropy?*

   We saw that when X represents a fair coin flip, $H(X) = 1$. This is, in fact, the *unique* distribution maximizing H. Applying this directly to the definition of $S(\rho)$, we find that $S(\rho)$ is maximized over all $\rho \in \mathrm{D}\left(\mathbb{C}^2\right)$ if and only if both eigenvalues of ρ are $1/2$. This implies that $\rho = I/2$. Moreover, this statement generalizes to any dimension $d \geq 2$ — for $\rho \in \mathrm{D}\left(\mathbb{C}^d\right)$, $S(\rho)$ is maximized if and only if $\rho = I/d$.

   Exercise 6.11. For $\rho = I/d$, what is $S(\rho)$?

3. *Quantum information is non-negative.*

   Since $H(X) \geq 0$, it immediately follows by definition that $S(\rho) \geq 0$.

4. *What is the quantum analogue of independent probability distributions X and Y?*

   Recall that in defining information content, the log function was chosen so as to ensure information is additive when two random variables X and Y are independently distributed. The quantum analogue of this has a natural expression: Let $\rho, \sigma \in \mathrm{D}\left(\mathbb{C}^d\right)$ be density matrices. Then, ρ and σ are independent if their joint state is $\rho \otimes \sigma$. Below, you will prove that this indeed preserves our desired additivity property of information for independent quantum states.

   Exercise 6.12. Prove that $S(\rho \otimes \sigma) = S(\rho) + S(\sigma)$.

There are other important properties of $S(\rho)$ which we do not wish to focus on at present; for completeness, however, let us briefly mention two more: (1) For arbitrary, possibly entangled, bipartite mixed states ρ_{AB}, $S(\rho_{AB}) \leq S(\rho_A) + S(\rho_B)$ (subadditivity), and (2) $S(\sum_{i=1}^n p_i \rho_i) \geq \sum_{i=1}^p p_i S(\rho_i)$ for $\{p_i\}_{i=1}^n$ a distribution (concavity). Here, and henceforth in this course, we use the shorthand

$$\rho_A := \mathrm{Tr}_B(\rho_{AB}). \tag{6.9}$$

6.2 Quantifying entanglement in composite quantum systems

With the notion of entropy in hand, we return to the following fundamental question. Let $\rho_{AB} \in \mathrm{D}\left(\mathbb{C}^d \otimes \mathbb{C}^d\right)$ be a bipartite quantum state. Can one efficiently determine if ρ_{AB} is

entangled? (Recall this means that ρ_{AB} cannot be written $\rho_{AB} = \sum_i p_i \rho_{A,i} \otimes \rho_{B,i}$ as a convex combination of product states.) Roughly, if one uses d to encode the size of the input (i.e. the input is the entire $d^2 \times d^2$ matrix representing ρ_{AB}), then deciding this question turns out to be NP-hard. This directly implies that quantifying "how much" entanglement is in ρ_{AB} is also NP-hard. However, there is a special case in which we *can* do both tasks efficiently — the case of bipartite *pure* states $|\psi_{AB}\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$. It is here in which the von Neumann entropy plays a role.

In fact, a previous lecture already discussed an efficient test for entanglement for $|\psi_{AB}\rangle$ — the latter is entangled if and only if

$$\rho_A := \mathrm{Tr}_B(|\psi_{AB}\rangle\langle\psi_{AB}|) \tag{6.10}$$

has rank at least 2. This, in turn, followed because it immediately implies the Schmidt rank of $|\psi_{AB}\rangle$ is at least 2. However, we can say more. Suppose we have Schmidt decomposition

$$|\psi_{AB}\rangle = \sum_{i=1}^{d} s_i |a_i\rangle|b_i\rangle. \tag{6.11}$$

Then, intuitively, as with the example of a Bell pair, $|\psi_{AB}\rangle$ is "highly entangled" if all the Schmidt coefficients s_i are approximately equal in magnitude, and $|\psi_{AB}\rangle$ is "weakly entangled" if there exists a single s_i whose magnitude is approximately 1. Do we know of a function which quantifies precisely this sort of behavior on the set $\{s_i\}$? Indeed, the entropy function! This notion of s_i being "spread out" versus "concentrated" is highly reminiscent of our fair versus biased coin flip example for the Shannon entropy. We can therefore use the von Neumann entropy to define an entanglement measure $E(|\psi_{AB}\rangle)$ as

$$E(|\psi_{AB}\rangle) := S(\rho_A). \tag{6.12}$$

Exercise 6.13. What is $E(|\Phi^+_{AB}\rangle)$ for $|\Phi^+_{AB}\rangle = \frac{1}{2}(|00\rangle + |11\rangle)$ a Bell state?

Exercise 6.14. What is $E(|+\rangle_A|-\rangle_B)$?

Exercise 6.15. Unlike the example of the fair coin, the Schmidt coefficients s_i of $|\psi_{AB}\rangle$ are not probabilities, but amplitudes (i.e. we do not have $\sum_i s_i = 1$, but rather $\sum_i s_i^2 = 1$). Show, however, that the notion of probabilities is recovered in the formula $S(\rho_A)$, i.e. show that the eigenvalues of ρ_A are the precisely set $\left\{s_i^2\right\}_{i=1}^d$, which *do* form a distribution.

Finally, let us close this section with a natural question — does $E(|\psi_{AB}\rangle)$ still measure entanglement when its input is allowed to be a mixed state ρ_{AB} (as opposed to a pure state $|\psi_{AB}\rangle$)? The answer is given in the following exercise.

Exercise 6.16. Define $E(\rho_{AB}) := S(\mathrm{Tr}_B(\rho_{AB})) = S(\rho_A)$. Recall that the maximally mixed state on two qubits is a product state, i.e. $I/4 = I/2 \otimes I/2$. Show that $E(I/4) = 1$. Why does this imply when cannot use E as an entanglement measure for bipartite mixed states?

6.3 Entanglement distillation

Now that we have a notion of how to quantify entanglement in pure states, we can become greedy — under what circumstances is it possible to "increase" the amount of entanglement in a composite quantum system? This is a highly non-trivial question, as fundamental communication tasks such as teleportation require highly entangled Bell pairs as a resource. Unfortunately, experimentally producing such pure states is generally a difficult task due to noise from the environment. In other words, in a lab one is typically able to produce *mixed* states, as opposed to pure states. Moreover, *even if* Alice could produce perfect Bell pairs in a lab on Earth, when she sends half of a Bell pair to Bob on Mars, the transmitted qubit will again typically be subject to noise, yielding a shared mixed state ρ_{AB} between Alice and Bob. Do Alice and Bob have any hope of running the teleportation protocol given ρ_{AB}?

Local Operations and Classical Communication (LOCC). To answer the question, it is important to first define the rules of the game. Since Alice and Bob are spatially separated, they are not able to apply joint quantum operations to both systems A and B, e.g. they cannot apply a non-factorizable unitary $U_{AB} \in \mathrm{U}\left(\mathbb{C}^d \otimes \mathbb{C}^d\right)$ to ρ_{AB}. However, they *can* apply local unitaries and measurements, e.g. factorizable unitaries of the form $U_A \otimes U_B$ for $U_A, U_B \in \mathrm{U}\left(\mathbb{C}^d\right)$ (i.e. Alice locally applies U_A, Bob locally applies U_B). They can also pick up the phone and call one another to transmit classical information. Taken together, this set of allowed operations is given a name: Local Operations and Classical Communication (LOCC). The question is thus: *Given a shared mixed state ρ_{AB}, can Alice and Bob use LOCC to "purify" or "distill" Bell states out of ρ_{AB}?* The answer is sometimes *yes*, and protocols accomplishing this are called *distillation protocols*, as they recover "pure" entanglement from "noisy" or mixed state entanglement.

A simple distillation protocol. We shall discuss a simple distillation protocol, known as the *recurrence protocol*. Given as input a mixed two-qubit state $\rho_{AB} \in \mathrm{D}\left(\mathbb{C}^2 \otimes \mathbb{C}^2\right)$, our aim is to distill the Bell state known as the *singlet*, $|\Psi^-\rangle = (|01\rangle - |10\rangle/\sqrt{2}$; note $I \otimes Y|\Psi^-\rangle = i|\Phi^+\rangle$, making it easy to convert one Bell state into the other for the teleportation scheme. There is a required precondition for the protocol to work — the input state ρ_{AB} must have sufficient initial overlap with $|\Psi^-\rangle$, i.e.

$$F(\rho_{AB}) := \langle\Psi^-|\rho_{AB}|\Psi^-\rangle > \frac{1}{2}. \tag{6.13}$$

In other words, in transmitting half of $|\Psi^-\rangle$ from Alice to Bob, the resulting mixed state ρ_{AB} should not have deviated "too far" from $|\Psi^-\rangle$. Henceforth, we shall use shorthand F to denote $F(\rho_{AB})$ for brevity.

Suppose Alice and Bob share two copies of ρ_{AB}; let us label them $\rho_{A_1B_1}$ and $\rho_{A_2B_2}$, where Alice holds systems A_1, A_2, and Bob holds B_1, B_2. Each round of the distillation protocol proceeds as follows.

1. (Twirling operation) Alice picks a Pauli operator U from set $\{I, X, Y, Z\}$ uniformly at random, and communicates this choice to Bob. They each locally apply operator $\sqrt{U}$ to $\rho_{A_iB_i}$ for $i \in \{1, 2\}$ (note that $\sqrt{U}$ is defined using the notion of operator functions, introduced in Section 6.1.2), obtaining

$$\sigma_{A_iB_i} := (\sqrt{U_A} \otimes \sqrt{U_B})\rho_{A_iB_i}(\sqrt{U_A}^\dagger \otimes \sqrt{U_B}^\dagger).$$

This random choice of Pauli and its subsequent local application is together called the *twirling* map $\Phi : \mathrm{D}\left(\mathbb{C}^2 \otimes \mathbb{C}^2\right) \mapsto \mathrm{D}\left(\mathbb{C}^2 \otimes \mathbb{C}^2\right)$, and is not a unitary map (due to the random choice over Pauli operators); nevertheless, it can clearly be implemented given the ability to flip random coins and apply arbitrary single qubit gates. (The formal framework for studying such operations is via *Trace Preserving Completely Positive Maps*, and is beyond the scope of this course.) The nice thing about the twirling operation is that, for any input ρ_{AB}, $\Phi(\rho_{AB})$ can be diagonalized in the Bell basis, i.e. can be written

$$\Phi(\rho_{AB}) = F|\Psi^-\rangle\langle\Psi^-| + \frac{1-F}{3}|\Psi^+\rangle\langle\Psi^+| + \frac{1-F}{3}|\Phi^+\rangle\langle\Phi^+| + \frac{1-F}{3}|\Phi^-\rangle\langle\Phi^-| \quad (6.14)$$

for Bell basis $\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$, where F is the same from Equation (6.13).

2. (Convert from $|\Psi^-\rangle$ to $|\Phi^+\rangle$) Alice applies Pauli Y to her half of each state to obtain states:

$$\sigma_{A_iB_i} = \frac{1-F}{3}|\Psi^-\rangle\langle\Psi^-| + \frac{1-F}{3}|\Psi^+\rangle\langle\Psi^+| + F|\Phi^+\rangle\langle\Phi^+| + \frac{1-F}{3}|\Phi^-\rangle\langle\Phi^-| \quad (6.15)$$

This shifts most of the weight in $\Phi(\rho_{A_iB_i})$ from $|\Psi^-\rangle$ to $|\Phi^+\rangle$, since $F > 1/2$ by Equation (6.13).

3. (Application of CNOT gates) Alice applies a CNOT gate with qubit A_1 as the control and A_2 as the target. Bob does the same for B_1 and B_2.

4. (Local measurements) Alice and Bob each locally measure A_2 and B_2 in the standard basis, obtaining outcomes a and b in $\{0, 1\}$, respectively. They pick up the phone to compare their measurement results a and b. If $a = b$, they keep the remaining composite system on AB, denoted $\sigma'_{A_1B_1}$. Otherwise if $a \neq b$, they throw out all systems and start again.

5. (Convert from $|\Phi^+\rangle$ to $|\Psi^-\rangle$) Alice applies Pauli Y to her half of $\sigma'_{A_1B_1}$ to convert its $|\Phi^+\rangle$ component back to $|\Psi^-\rangle$.

To get a better sense of this protocol in action, let us apply it to a concrete example. Suppose Alice sends half of the singlet to Bob, and along the way, the state $|\Psi^-\rangle$ is injected with some completely random noise, denoted by the identity matrix:

$$\rho_{AB} = \frac{1}{2}|\Psi^-\rangle\langle\Psi^-| + \frac{1}{8}I = \frac{3}{4}|\Psi^-\rangle\langle\Psi^-| + \frac{1}{12}|\Psi^+\rangle\langle\Psi^+| + \frac{1}{12}|\Phi^+\rangle\langle\Phi^+| + \frac{1}{12}|\Phi^-\rangle\langle\Phi^-|, \quad (6.16)$$

where the second equality follows by recalling the identity matrix diagonalizes in any basis, including the Bell basis. (The noise-inducing channel above is formally known as the *depolarizing channel* in quantum information theory.)

Exercise 6.17. Show that $\sqrt{Z} \otimes \sqrt{Z}$ maps $|\Phi^+\rangle$ to $|\Phi^-\rangle$ and vice versa. Using the additional identities that $\sqrt{X} \otimes \sqrt{X}$ maps $|\Phi^+\rangle$ to $|\Psi^+\rangle$ and vice versa, and $\sqrt{Y} \otimes \sqrt{Y}$ maps $|\Phi^-\rangle$ to $|\Psi^+\rangle$ and vice versa, show that the twirling map leaves ρ_{AB} in Equation (6.16) invariant.

Exercise 6.18. Show that applying $Y_A \otimes I$ to ρ_{AB} yields in Step 2 that

$$\sigma_{AB} = \frac{1}{12}|\Psi^-\rangle\langle\Psi^-| + \frac{1}{12}|\Psi^+\rangle\langle\Psi^+| + \frac{3}{4}|\Phi^+\rangle\langle\Phi^+| + \frac{1}{12}|\Phi^-\rangle\langle\Phi^-|. \tag{6.17}$$

Finally, Steps 3 and 4 are a bit messier. The intuition here is that the CNOT entangles $\sigma_{A_1B_1}$ with $\sigma_{A_2B_2}$, and the measurement forces the bits in the second system (formerly in state $\sigma_{A_2B_2}$) to match; via the entanglement just created, this increases the weight in Equation (6.17) on the terms where bits match, i.e. $|\Phi^+\rangle\langle\Phi^+|$ and $|\Phi^-\rangle\langle\Phi^-|$. Thus, the final Step 5 will yield a state with higher overlap on the desired singlet state $|\Psi^-\rangle$.

To run through the full technical analysis for Steps 3 and 4 would be tedious, so we analyze one term of the computation for brevity. Before Step 3 is run, Alice and Bob share state

$$\sigma_{A_1B_1} \otimes \sigma_{A_2B_2} \in \mathrm{D}\left(\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2\right), \tag{6.18}$$

where recall Alice holds qubits A_1, A_2, and Bob holds B_1, B_2. Since each $\sigma_{A_iB_i}$ has 4 terms in its mixture, the tensor product in Equation (6.18) has 16 terms. By linearity, Step 3 then applies gates $\mathrm{CNOT}_{A_1A_2}$ and $\mathrm{CNOT}_{B_1B_2}$ to each of these 16 terms, where our notational convention is that CNOT_{12} has qubit 1 as the control and qubit 2 as the target. Let us analyze one of these 16 terms: $|\Phi^+\rangle\langle\Phi^+|_{A_1B_1} \otimes |\Phi^+\rangle\langle\Phi^+|_{A_2B_2}$.

Exercise 6.19. Show that

$$(\mathrm{CNOT}_{A_1A_2} \otimes \mathrm{CNOT}_{B_1B_2})|\Phi^+\rangle_{A_1B_1} \otimes |\Phi^+\rangle_{A_2B_2} = \frac{1}{2}\left(|0000\rangle + |0011\rangle + |1100\rangle + |1111\rangle\right)_{A_1B_1A_2B_2}.$$

If Alice and Bob run Step 4 on this state and obtain matching outcomes $a = b$, what does the state on qubits A_1B_1 collapse to?

Finally, let us briefly state what this protocol buys us. A careful but tedious analysis yields that with probability at least $1/4$, this protocol maps the input state $\rho_{A_1B_1} \otimes \rho_{A_2B_2}$ to an output state $\sigma'_{A_1B_1}$ such that (for $F_\rho := F(\rho_{A_1B_1})$)

$$F(\sigma'_{A_1B_1}) = \frac{F_\rho^2 + \frac{1}{9}(1-F_\rho)^2}{F_\rho^2 + \frac{2}{3}F_\rho(1-F_\rho) + \frac{5}{9}(1-F_\rho)^2}. \tag{6.19}$$

So long as $F_\rho > 1/2$, one can show $F(\sigma'_{A_1B_1}) > F_\rho$; thus, recursively applying this protocol (using many pairs of input states ρ_{AB}) improves our overlap with our desired target state of $|\Psi^-\rangle$.

7 The Deutsch-Josza and Berstein-Vazirani algorithms

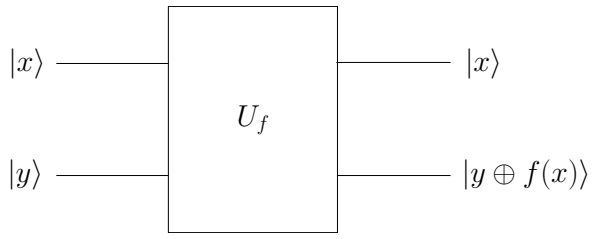
"Computers are physical objects, and computations are physical processes. What computers can or cannot compute is determined by the laws of physics alone. . ."
— David Deutsch

"Where there's smoke, there's fire."
— Latin proverb

In the lectures thus far, we've introduced the postulates of quantum mechanics, and studied them through the lens of quantum information theoretic concepts such as entanglement, nonlocal games, and entropy. We now switch to the theme of *quantum algorithms*, i.e. algorithms harnessing the four postulates of quantum mechanics. We begin with a simple quantum algorithm due to David Deutsch, which is by no means new (it was discovered in 1985), nor does it tackle a particularly important problem (in fact, the problem is quite artificial). Nevertheless, Deutsch's algorithm serves as an excellent proof of concept that, in certain settings, quantum computers are strictly more powerful than classical ones. Moreover, as shown by Berstein and Vazirani, the generalization of this algorithm (dubbed the Deutsch-Josza algorithm) can actually be seen as solving a more natural problem — given a black-box function $f : \{0,1\}^n \mapsto \{0,1\}$ which computes $f(x) = a \cdot x \mod 2$ for some unknown $a \in \{0,1\}^n$, what is a?

7.1 The setup: Functions as oracles

The problem which Deutsch's algorithm tackles is stated in terms of binary functions $f : \{0,1\} \mapsto \{0,1\}$. Thus, the first thing we'll need to do is understand how to *model* such functions in the quantum circuit model. What makes the task slightly non-trivial is that, recall by Postulate 2 of quantum mechanics, all quantum operations must be unitary and hence *reversible*. In general, however, given the output $f(x)$ of a function, it is not always possible to *invert* f to obtain the input x. In other words, we have to compute $f(x)$ in such a way as to guarantee that the computation can be undone. This is achieved via the following setup:



Here, $U_f \in \mathcal{U}((\mathbb{C}^2)^{\otimes 2})$ is a unitary operator mapping $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$ for any $x, y \in \{0,1\}$ (i.e. $|x\rangle, |y\rangle$ denote standard basis states), and where $\oplus$ denotes XOR or addition modulo 2. Note that by linearity, once we define the action of U_f on standard basis states, we immediately know how it acts on *any* input state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes 2}$.

Exercise 7.1. Let $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$. What is the state $U_f|\psi\rangle$?

Observe now that U_f is reversible — this is because running U_f again on its output, $|x\rangle|y \oplus f(x)\rangle$, yields state $|x\rangle|y \oplus f(x) \oplus f(x)\rangle = |x\rangle|y\rangle$, since $f(x) \oplus f(x) = 0$ (adding the same bit twice and dividing by 2 leaves remainder zero). Second, note that we have not specified the inner workings of U_f (i.e. we have not given a circuit implementing the functionality stated above); in this course, we shall treat U_f as a "black box" or "oracle" which we presume we can run, but cannot "look inside" to see its implementation details.

7.2 The problem: Is f constant or balanced?

The problem Deutsch's algorithm tackles can now be stated as follows. Given a block box U_f implementing some unknown function $f : \{0,1\} \mapsto \{0,1\}$, determine whether f is "constant" or "balanced". Here, *constant* means f always outputs the same bit, i.e. $f(0) = f(1)$, and *balanced* means f outputs different bits on different inputs, i.e. $f(0) \neq f(1)$.

Exercise 7.2. Suppose $f(0) = 1$ and $f(1) = 0$. Is f constant or balanced? Given an example of a constant f.

Of course, there is an easy way to determine whether f is constant or balanced — simply evaluate f on inputs 0 and 1, i.e. compute $f(0)$ and $f(1)$, and then check if $f(0) = f(1)$. This naive (classical) solution, however, requires two *queries* or calls to U_f (i.e. one to compute $f(0)$ and one to compute $f(1)$). So certainly at most two queries to U_f suffice to solve this problem. Can we do it with just *one* query? Classically, the answer turns out to be no. But quantumly, Deutsch showed how to indeed achieve this with a single query.

Quantum query complexity. As you may have noticed above, the "cost function" we are interested in minimizing in solving Deutsch's problem is the number of quantum queries to U_f. This is an example of the model of *quantum query complexity*, in which many quantum algorithms have been developed. In the study of quantum query complexity, one is given a black box U_f implementing some function f, and asked what the minimum number of required queries to U_f is in order to determine some desired property of f. Note that the quantum algorithm computing this property can consist of (say) 999999999 quantum gates; if it contains only 2 queries to U_f, then we consider the cost of the algorithm as 2, i.e. all "non-query" operations are considered free.

7.3 The algorithm

7.3.1 A naive idea

Before demonstrating the algorithm itself, let us first attempt a simpler, more naive approach — since we are allowed to query U_f quantumly, what happens if we just query U_f in *superposition* in the input register? In other words, what happens if we run the circuit for U_f with input state $|x\rangle$ replaced with $\alpha|0\rangle + \beta|1\rangle$ and output state $|y\rangle$ with $|0\rangle$? Intuitively, here we have set the input register to *both* possible inputs 0 and 1, and so we expect U_f to return a superposition of both possible outputs, $f(0)$ and $f(1)$. Indeed, by linearity of U_f, the output of the circuit will be

$$|\psi\rangle = U_f(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle = \alpha U_f|0\rangle|0\rangle + \beta U_f|1\rangle|0\rangle = \alpha|0\rangle|f(0)\rangle + \beta|1\rangle|f(1)\rangle.$$

Thus, we seem to have obtained both outputs of f with just a single query! Unfortunately, things in life are rarely free, and this is certainly no exception — although we have both outputs $f(0)$ and $f(1)$ in superposition, we cannot hope to *extract* both answers via measurement. In particular, once we measure both registers in the standard basis, we will collapse to one of the two terms in the superposition, effectively destroying the other term.

Exercise 7.3. Suppose one measures the first qubit of the output state $|\psi\rangle$ (this qubit marks which term in the superposition we have) with a standard basis measurement $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$. Show that the probability of outcome 0 or 1 is $|\alpha|^2$ or $|\beta|^2$, respectively, and that in each case, the state collapses to either $|0\rangle|f(0)\rangle$ or $|1\rangle|f(1)\rangle$, respectively. Thus, only one answer $f(0)$ or $f(1)$ can be extracted this way.

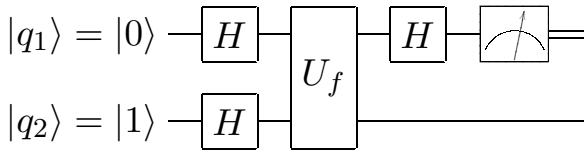
Luckily, our goal is *not* to extract both $f(0)$ and $f(1)$ after a single query. Rather, we want something possibly simpler: To evaluate the expression $f(0) \oplus f(1)$.

Exercise 7.4. Convince yourself that f is constant if $f(0) \oplus f(1) = 0$ and f is balanced if $f(0) \oplus f(1) = 1$. Thus, Deutsch's problem is equivalent to evaluating $f(0) \oplus f(1)$.

It turns out that by a clever twist of the naive approach above, we can indeed evaluate $f(0) \oplus f(1)$ (*without* individually obtaining the values $f(0)$, $f(1)$) via Deutsch's algorithm.

7.3.2 Deutsch's algorithm

The circuit for Deutsch's algoritm is given as follows.



It is not *a priori* obvious at all why this circuit should work, and this is indicative of designing quantum algorithms in general — the methods used are often incomparable to known classical algorithm design techniques, and thus developing an intuition for the quantum setting can be very difficult. Let us hence simply crunch the numbers and see why this circuit indeed computes $f(0) \oplus f(1)$, as claimed. Once we've done the brute force calculation, we will take a step back and talk about the *phase kickback* trick, which is being used here, and which will allow for a much simpler and somewhat more intuitive understanding of why the algorithm works.

As in previous lectures, let us divide the computation into 4 stages denoted by the quantum state in that stage: At the start of the circuit ($|\psi_1\rangle$), after the first Hadamards are applied ($|\psi_2\rangle$), after U_f is applied ($|\psi_3\rangle$), and after the last Hadamard is applied ($|\psi_4\rangle$). It is clear that

$$\begin{aligned} |\psi_1\rangle &= |0\rangle|1\rangle, \\ |\psi_2\rangle &= |+\rangle|-\rangle = \frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle). \end{aligned}$$

After the oracle U_f is applied, we have state

$$|\psi_3\rangle = \frac{1}{2}(|0\rangle|f(0)\rangle - |0\rangle|1 \oplus f(0)\rangle + |1\rangle|f(1)\rangle - |1\rangle|1 \oplus f(1)\rangle).$$

Before we apply the final Hadamard, it will be easier to break our analysis down into two cases: When f is constant and when f is balanced.

Case 1: Constant f. By definition, if f is constant, then $f(0) = f(1)$. Therefore, we can simplify $|\psi_3\rangle$ to

$$\begin{aligned}
|\psi_3\rangle &= \frac{1}{2}(|0\rangle|f(0)\rangle - |0\rangle|1 \oplus f(0)\rangle + |1\rangle|f(0)\rangle - |1\rangle|1 \oplus f(0)\rangle) \\
&= \frac{1}{2}((|0\rangle + |1\rangle) \otimes |f(0)\rangle - (|0\rangle + |1\rangle) \otimes |1 \oplus f(0)\rangle) \\
&= \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|f(0)\rangle - |1 \oplus f(0)\rangle) \\
&= \frac{1}{\sqrt{2}}|+\rangle \otimes (|f(0)\rangle - |1 \oplus f(0)\rangle).
\end{aligned}$$

Thus, qubit 1 is now in state $|+\rangle$. We conclude that

$$|\psi_4\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes (|f(0)\rangle - |1 \oplus f(0)\rangle),$$

i.e. qubit 1 is exactly in state $|0\rangle$. Thus, measuring qubit 1 in the standard basis now yields outcome 0 with certainty.

Case 2: Balanced f. By definition, if f is balanced, then $f(0) \neq f(1)$. Since f is a binary function, this means $f(0) \oplus 1 = f(1)$ and equivalently $f(1) \oplus 1 = f(0)$. Therefore, we can simplify $|\psi_3\rangle$ to

$$\begin{aligned}
|\psi_3\rangle &= \frac{1}{2}(|0\rangle|f(0)\rangle - |0\rangle|f(1)\rangle + |1\rangle|f(1)\rangle - |1\rangle|f(0)\rangle) \\
&= \frac{1}{2}((|0\rangle - |1\rangle) \otimes |f(0)\rangle - (|0\rangle - |1\rangle) \otimes |f(1)\rangle) \\
&= \frac{1}{2}(|0\rangle - |1\rangle) \otimes (|f(0)\rangle - |f(1)\rangle) \\
&= \frac{1}{\sqrt{2}}|-\rangle \otimes (|f(0)\rangle - |f(1)\rangle).
\end{aligned}$$

Thus, qubit 1 is now in state $|-\rangle$. We conclude that

$$|\psi_4\rangle = \frac{1}{\sqrt{2}}|1\rangle \otimes (|f(0)\rangle - |f(1)\rangle),$$

i.e. qubit 1 is exactly in state $|1\rangle$. Thus, measuring qubit 1 in the standard basis now yields outcome 1 with certainty.

Conclusion. If f is constant, the algorithm outputs 0, and if f is balanced, the algorithm outputs 1. Thus, the algorithm decides whether f is constant or balanced, using just a single query!

7.3.3 The phase kickback trick

We've analyzed Deutsch's algorithm using a brute force calculation, but there's a more intuitive view which will be used repeatedly in later algorithms, and which simplifies our calculation here greatly. This view is in terms of the *phase kickback trick*, which Deutsch's algorithm uses. To explain the trick, consider for any $x \in \{0, 1\}$ what happens if we run U_f on input $|x\rangle|-\rangle$:

$$|\psi\rangle = U_f|x\rangle|-\rangle = \frac{1}{\sqrt{2}}(U_f|x\rangle|0\rangle - U_f|x\rangle|1\rangle) = \frac{1}{\sqrt{2}}(|x\rangle|f(x)\rangle - |x\rangle|1 \oplus f(x)\rangle) = |x\rangle \otimes \frac{1}{\sqrt{2}}(|f(x)\rangle - |1 \oplus f(x)\rangle).$$

Now, there are two possibilities: Either $f(x) = 0$, or $f(x) = 1$. If $f(x) = 0$, the equation above simplifies to

$$|\psi\rangle = |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |x\rangle|-\rangle,$$

i.e. the input state is unchanged by the action of U_f. If, on the other hand, $f(x) = 1$, we instead have

$$|\psi\rangle = |x\rangle \otimes \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle) = -|x\rangle|-\rangle,$$

i.e. a -1 phase factor is produced. We can summarize both these cases in a single equation:

$$U_f|x\rangle|-\rangle = (-1)^{f(x)}|x\rangle|-\rangle. \tag{7.1}$$

Exercise 7.5. Convince yourself that the equation above indeed captures both the cases of $f(x) = 0$ and $f(x) = 1$.

Reanalyzing Deutsch's algorithm using phase kickback. Let us return to the state in Deutsch's algorithm just before U_f is applied, i.e.

$$|\psi_2\rangle = |+\rangle|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle|-\rangle + |1\rangle|-\rangle).$$

(Note that we have not expanded out the $|-\rangle$ state as we did previously — this is because with the phase kickback trick, we don't need to go to this level of detail!) By applying phase kickback (Equation (7.1)), we know that after U_f is applied, we have state

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle|-\rangle + (-1)^{f(1)}|1\rangle|-\rangle).$$

Suppose now that f is constant, i.e. $f(0) = f(1)$. Then, above we can factor out the -1 phase factor to simplify $|\psi_3\rangle$ to

$$|\psi_3\rangle = (-1)^{f(0)}\frac{1}{\sqrt{2}}(|0\rangle|-\rangle + |1\rangle|-\rangle) = (-1)^{f(0)}|+\rangle|-\rangle.$$

Thus, applying the final Hadamard to qubit 1 yields

$$|\psi_4\rangle = (-1)^{f(0)}|0\rangle|-\rangle.$$

Measuring the first qubit now yields outcome 0 with certainty, as before.

On the other hand, if f is balanced (i.e. $f(0) \neq f(1)$), then we cannot simply factor out the (-1) term as before! Thus, up to an overall factor of ± 1, $|\psi_3\rangle$ can be written as

$$|\psi_3\rangle = \pm\frac{1}{\sqrt{2}}(|0\rangle|-\rangle - |1\rangle|-\rangle) = \pm|-\rangle|-\rangle.$$

Exercise 7.6. Verify the equation above by considering the two possible balanced functions $f_1(0) = 0$ and $f_1(1) = 1$ and $f_2(0) = 1$ and $f_2(1) = 0$.

We conclude that applying the final Hadamard to qubit 1 yields

$$|\psi_4\rangle = \pm|1\rangle|-\rangle.$$

Measuring the first qubit now yields outcome 1 with certainty, as before.

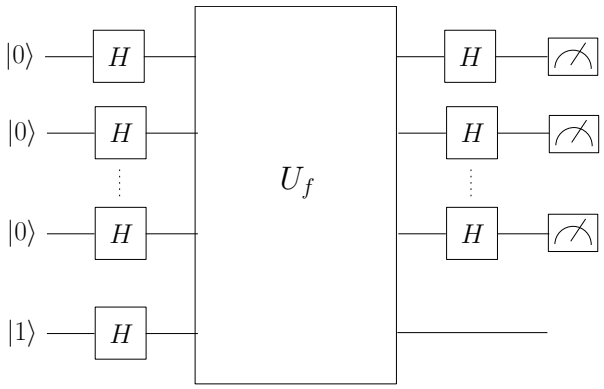


Figure 7.1: Quantum circuit for the Deutsch-Josza algorithm.

7.4 The Deutsch-Josza algorithm

Deutsch's algorithm works in the simple case where $f : \{0,1\} \mapsto \{0,1\}$ acts on a single input bit. However, single-bit functions are not so interesting; our primary area of interest is the design and analysis of quantum algorithms for determining properties of functions $f : \{0,1\}^n \mapsto \{0,1\}$ which act on *many* input bits. This requires some familiarity in handling n-qubit states, and a good way to practice this is by developing the n-bit generalization of Deutsch's algorithm, known as the Deutsch-Josza algorithm.

Specifically, imagine now we have an n-bit function $f : \{0,1\}^n \mapsto \{0,1\}$ which is promised to be constant or balanced, and we wish to determine which is the case. Here, constant means $f(x)$ is the same for all $x \in \{0,1\}$, and balanced means $f(x) = 0$ for precisely half the $x \in \{0,1\}^n$ and $f(x) = 1$ for the remaining inputs?

Exercise 7.7. Give examples of balanced and constant functions on 2 bits. Can you give an example of a 2-bit function which is *neither* constant nor balanced? Finally, can a single-bit function be neither constant nor balanced?

It turns out that Deutsch's algorithm generalizes in an easy manner to this setting; however, its analysis is a bit more tricky, and crucially uses the phase kickback trick. In this more general setting, note that we define the oracle U_f implementing f analogously to before: $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$, where now x is an n-bit string.

The circuit for the Deutsch-Josza algorithm is given in Figure 7.7. As before, each wire denotes a single qubit. The first n qubits are initialized to $|0\rangle$; these are the input qubits. The final, i.e. $(n+1)$st, qubit is initialized to $|1\rangle$. Observe that the algorithm is the straightforward generalization of Deutsch's algorithm to the setting of n input qubits. We claim that using a single query to U_f, the Deutsch-Josza algorithm can determine if f is constant or balanced. Let us now see why this is so.

As before, we divide the computation into 4 stages denoted by the quantum state in that stage: At the start of the circuit ($|\psi_1\rangle$), after the first Hadamards are applied ($|\psi_2\rangle$), after U_f is applied ($|\psi_3\rangle$), and after the last Hadamard is applied ($|\psi_4\rangle$). It is clear that

$$\begin{aligned} |\psi_1\rangle &= |0\rangle \cdots |0\rangle|1\rangle = |0\rangle^{\otimes n}|1\rangle, \\ |\psi_2\rangle &= |+\rangle \cdots |+\rangle|-\rangle = |+\rangle^{\otimes n}|1\rangle. \end{aligned}$$

Since we have defined the action of U_f in terms of the standard basis, i.e. $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$, in order to understand how U_f applies to $|\psi_2\rangle$, we first need to rewrite $|+\rangle^{\otimes n}$ in terms of the standard basis. For this, note that

$$|+\rangle^{\otimes n} = \frac{1}{\sqrt{2}^n}(|0\rangle + |1\rangle) \otimes \cdots \otimes (|0\rangle + |1\rangle) = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle,$$

where the last equality holds since expanding the tensor products out yields 2^n terms in the sum, each of which corresponds to a unique string $x \in \{0,1\}^n$.

Exercise 7.8. Verify that $|+\rangle^{\otimes 3} = \frac{1}{2\sqrt{2}} \sum_{x \in \{0,1\}^3} |x\rangle$.

It follows that we can rewrite $|\psi_2\rangle$ as

$$|\psi_2\rangle = |+\rangle^{\otimes n}|1\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle|-\rangle.$$

Now that we have the first register written with respect to the standard basis, we can analyze the action of U_f using the phase kickback trick, obtaining state

$$|\psi_3\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)}|x\rangle|-\rangle.$$

Finally, we must apply the last set of Hadamard gates, which gives $|\psi_4\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} H^{\otimes n}|x\rangle|-\rangle$. To analyze this state, we first need to understand what $H^{\otimes n}|x\rangle$ equals for arbitrary $x \in \{0,1\}$. For this, we begin with a clean and formal way for writing the action of H on a *single* qubit. Recall that $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$. Equivalently, this means that for $x_1 \in \{0,1\}$,

$$H|x_1\rangle = \frac{1}{\sqrt{2}} \sum_{z_1 \in \{0,1\}} (-1)^{x_1 z_1}|z_1\rangle,$$

where $x_1 z_1$ is just the product of x_1 and z_1.

Exercise 7.9. Verify the statement above by considering the cases $H|0\rangle$ and $H|1\rangle$.

Now that we have a clean way of expressing $H|x_1\rangle$ for single qubit $|x_1\rangle$, we can generalize this to n-qubit states. Specifically, if we write string $x = x_1 \cdots x_n$ as $|x_1\rangle \otimes \cdots \otimes |x_n\rangle$, we have

$$\begin{aligned} H^{\otimes n}|x\rangle &= H|x_1\rangle \otimes \cdots \otimes H|x_n\rangle \\ &= \frac{1}{2^{n/2}} \sum_{z_1 \in \{0,1\}} (-1)^{x_1 z_1}|z_1\rangle \otimes \cdots \otimes \sum_{z_n \in \{0,1\}} (-1)^{x_n z_n}|z_n\rangle \\ &= \frac{1}{2^{n/2}} \sum_{z \in \{0,1\}^n} (-1)^{x_1 z_1 + \cdots + x_n z_n}|z\rangle. \end{aligned}$$

Can we simplify this expression further? There is one small last trick we can apply which will clean it up a bit: Observe that $x_1 z_1 + \cdots x_n z_n$ can be rewritten as the bitwise inner product modulo 2 of strings x and z, i.e. $x_1 z_1 + \cdots x_n z_n = x \cdot z$. (The mod 2 arises since the base is

(-1), so all we care about is if the exponent $x \cdot z$ is even or odd.) Combining these facts, we have that after the final Hadamards are applied, we have

$$\begin{aligned}
|\psi_4\rangle &= \frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} H^{\otimes n}|x\rangle|-\rangle \\
&= \frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} \left(\frac{1}{2^{n/2}} \sum_{z\in\{0,1\}^n} (-1)^{x\cdot z}|z\rangle \right) |-\rangle \\
&= \frac{1}{2^n} \sum_{z\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)+x\cdot z}|z\rangle|-\rangle. \qquad (7.2)
\end{aligned}$$

Finally, a measurement of the first n qubits of $|\psi_4\rangle$ is made in the standard basis. As for Deutsch's algorithm, a good way to analyze this is by individually considering the cases when f is constant or balanced, respectively. The trick in both analyses will be to determine the amplitude on the all zeroes state, $|0\rangle^{\otimes n}$, in the first register.

Case 1: Constant f. Suppose first that f is constant. Then, we can factor out the term $(-1)^{f(x)}$ in $|\psi_4\rangle$, i.e.

$$|\psi_4\rangle = (-1)^{f(x)} \sum_{z\in\{0,1\}^n} \left(\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{x\cdot z} \right) |z\rangle|-\rangle.$$

Now consider the amplitude on $|z\rangle = |0\cdots 0\rangle$, which is given by (up to an insignificant $(-1)^{f(x)}$ global phase out front)

$$\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{x\cdot 0\cdots 0} = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^0 = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} 1 = \frac{1}{2^n} 2^n = 1.$$

In other words, the state $|0\rangle^{\otimes n}|-\rangle$ has amplitude 1. Since $|\psi_4\rangle$ is a unit vector, we conclude that we must have $|\psi_4\rangle = (-1)^{f(x)}|0\cdots 0\rangle|-\rangle$, i.e. all the weight is on this one term. Thus, if f is constant, then measuring the first n qubits in the standard basis yields outcome $|0\cdots 0\rangle$ with certainty.

Case 2: Balanced f. In this case, we cannot simply factor out the $(-1)^{f(x)}$ term, since f can take on different values depending on its input x. However, it still turns out to be fruitful to think about the amplitude on the state $|z\rangle = |0\cdots 0\rangle$. This is given by

$$\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)+x\cdot 0\cdots 0} = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}$$

since $z = 0$. Since f is balanced, we know that for precisely half the terms in this sum, $f(x) = 0$, and for the other half $f(x) = 1$. In other words, all the terms in this sum cancel, i.e. the sum equals 0! We conclude that the amplitude on $|z\rangle = |0\cdots 0\rangle$ is zero, and so we will never see outcome $|0\cdots 0\rangle$ in the final measurement.

Combining our observations for both cases, we find the following: When the final measurement is completed, if the outcome is 0^n, then we output "constant"; for any other n-bit measurement result, we output "balanced".

Classical algorithms for the Deutsch-Josza problem. Finally, you might wonder how *classical* algorithms compete with the Deutsch-Josza algorithm. For the case of *deterministic* classical algorithms, if f is balanced, then there are $2^{n/2}$ inputs x yielding $f(x) = 0$ and $2^{n/2}$ inputs x yielding $f(x) = 1$. Thus, in the worst case, an algorithm might be very unlucky and have its first $2^{n/2}$ queries return value $f(x) = 0$, only to have query $2^{n/2} + 1$ return $f(x) = 1$. For this reason, deterministic algorithms have worst case query complexity $2^{n/2} + 1$. In this setting, the Deutsch-Josza algorithm yields an exponential improvement over classical algorithms, requiring just a single query to f.

However, one can also try a *randomized* classical algorithm. Here is a particularly simple one:

1. Repeat the following K times, for K a parameter to be chosen as needed:
   a) Pick $x \in \{0, 1\}$ uniformly at random.
   b) Call U_f to evaluate $f(x)$.
   c) If $f(x)$ differs from any previous query answer, then halt and output "balanced".
2. Halt and output "Constant".

This algorithm does something straightforward — it repeatedly tries random values of $f(x)$, and if it ever obtains two different answers to its queries, it outputs "balanced"; otherwise, all its queries returned the same answer, and so it outputs "constant". Will this algorithm always be correct? *No.* In fact, it has *one-sided error*, in the following sense. If f is constant, then all queries will always output the same answer. Thus, line 1c will never cause the program to halt, and it will correctly output "constant" on line 2. On the other hand, if f is balanced, then the algorithm might get really unlucky — all of its K queries $f(x)$ might output the same bit, even though f is balanced. Thus, in this case the algorithm will incorrectly output "constant" on line 2.

We are left with two questions: What is the query cost of this randomized algorithm, and what is its probability of error? Clearly, the cost is K queries, since that is the number of times the loop runs. As for the error, note that when f is balanced (which is the only case in which an error might occur), when making a single query, the probability of getting output (say) $f(x) = 0$ is $1/2$. Since all query inputs are uniformly and independently chosen at random, the probability of having all K queries return the same bit is hence $1/2^{K-1}$ (the -1 in the exponent is because there are two such cases, strings 0^K and 1^K). We conclude that the error probability scales inverse exponentially with K.

Exercise 7.10. Suppose we wish our randomized algorithm to have error probability at most $1/n$. What should we set K to?

Finally, let us compare this to the Deutsch-Josza algorithm. Suppose that f is chosen to be constant with probability $1/2$ and balanced with probability $1/2$. Then, the Deutsch-Josza algorithm uses 1 query to determine if f is constant or balanced with certainty. On the other hand, if the randomized algorithm uses $K \in O(1)$ queries, then its probability of success is

$$\begin{aligned}
\Pr[\text{success}] &= \Pr[f \text{ is constant}] \cdot \Pr[\text{output "constant"} \mid \text{f is constant}] + \\
&\quad \Pr[f \text{ is balanced}] \cdot \Pr[\text{output "balanced"} \mid \text{f is balanced}] \\
&= \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot (1 - \frac{1}{2^{K-1}}) \\
&= 1 - \frac{1}{2^K}.
\end{aligned}$$

Exercise 7.11. What is the probability of success for the randomized algorithm if it performs just a single query, i.e. $K = 1$? How does this compare with the Deutsch-Josza algorithm?

7.5 The Berstein-Vazirani algorithm

Although interesting as a proof of principle, the Deutsch-Josza algorithm does not solve a particularly motivated problem. However, as the second quote opening this lecture note suggests[1], "where there's smoke, there's fire", and a closer look at the Deutsch-Josza algorithm will reveal something more interesting at play.

To begin, recall Equation (7.2), which stated that just before measuring in the standard basis, the Deutsch-Josza algorithm yields state

$$|\psi_4\rangle = \frac{1}{2^n} \sum_{z\in\{0,1\}^n} \left(\sum_{x\in\{0,1\}^n} (-1)^{f(x)+x\cdot z} \right) |z\rangle|-\rangle. \tag{7.3}$$

Note that this derivation was *independent* of the definition of f. This raises the question: For what choices of f could measuring $|\psi_4\rangle$ yield interesting information? Since the exponent on -1 contains $x \cdot z$, a naive guess might be to consider linear functions $f(x) = a \cdot x$, where $a, x \in \{0,1\}^n$ (or more generally affine functions $f(x) = a \cdot x + b$ for $b \in \{0,1\}$). This seems a natural guess, as it would allow us to factor out the a term. Specifically, plugging $f(x) = a \cdot x$ into Equation (7.3) yields

$$|\psi_4\rangle = \frac{1}{2^n} \sum_{z\in\{0,1\}^n} \left(\sum_{x\in\{0,1\}^n} (-1)^{(a+z)\cdot x} \right) |z\rangle|-\rangle. \tag{7.4}$$

Thus, the probability of obtaining some outcome z when measuring the first register in the standard basis is $(2^{-n} \sum_{x\in\{0,1\}^n} (-1)^{(a+z)\cdot x})^2$. Since our goal is to extract a, let us ask: What is the probability of observing $z = a$? This is given by (recall we are working modulo 2 in the exponent)

$$\left(\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{(a+a)\cdot x} \right)^2 = \left(\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{0\cdot x} \right)^2 = \left(\frac{1}{2^n} \sum_{x\in\{0,1\}^n} 1 \right)^2 = 1. \tag{7.5}$$

In other words, all of the amplitude in $|\psi_4\rangle$ is concentrated on the $|a\rangle$ term. Equivalently, $|\psi_4\rangle = |a\rangle|-\rangle$. We thus have that measuring the first register yields a with certainty. This procedure is known as the Bernstein-Vazirani algorithm, which solves the problem: Given a black-box linear function $f(x) = a \cdot x$, determine a.

Exercise 7.12. Is it possible to have $z \neq a$ in Equation (7.4)? In other words, does there exist a term in the superposition with non-zero amplitude which satisfies $z \neq a$?

Exercise 7.13. Given an affine function $f(x) = ax + b$, show how the Bernstein-Vazirani algorithm allows one to extract a with a single quantum query.

[1] We are slightly abusing this proverb here, in that it typically has a negative connotation, whereas here our premise is that quantum computation outperforming classical computation is a good thing.

Interpretation via the Deutsch-Josza problem. To close the lecture, let us go full circle and see what Bernstein-Vazirani teaches us about Deutsch-Josza. Suppose first that $f(x) = b$ for $b \in \{0, 1\}$, i.e. we have affine function $f(x) = ax + b$ with $a = 0$ in the Bernstein-Vazirani framework. Then, clearly f is constant in the Deutsch-Josza framework. In both algorithms (which are the same algorithm), we hence obtain all-zeroes in the first register when we measure $|\psi_4\rangle$.

Next, consider $f(x) = ax + b$ for $a \neq 0$. (By an exercise above, we can ignore b, as it simply leads to a global phase in $|\psi_4\rangle$, which cannot be observed.) Since the case of $a = 0$ corresponded to constant f, we now expect the $a \neq 0$ case to correspond to *balanced* f. The following basic, but important, fact in Boolean arithmetic confirms this, and is worth keeping in your toolkit moving forward.

Exercise 7.14. Let $f(x) = a \cdot x + b \mod 2$ be an affine function with $a \neq 0$. Then, for precisely half the inputs $x \in \{0, 1\}^n$, $f(x) = 0$, and on the other half of the inputs, $f(x) = 1$. In other words, f is balanced.

8 Simon's algorithm and applications to cryptography

"Great things are not done by impulse, but by a series of small things brought together."
— Vincent van Gogh.

In the previous lecture, we began our foray into quantum algorithms with the Deutsch, Deutsch-Josza, and Berstein-Vazirani algorithms (the latter two were actually the same algorithm). Although an important proof of concept, the Deutsch-Josza algorithm did not give us a strong separation between classical and quantum query complexity, as recall a classical *randomized* algorithm can solve the Deutsch-Josza problem with only $O(2^{-K})$ probability of error, where K is the number of oracle queries.

In this lecture, we shall study Simon's algorithm, which was the first to yield an exponential separation in query complexity between the quantum and classical randomized settings. *A priori*, Simon's problem will again appear artificial. First appearances, however, can be deceiving: Not only has recent research shown that Simon's algorithm can be used to break certain classical cryptosystems, but the algorithm was a crucial step towards inspiring Peter Shor in his development of the celebrated quantum factoring algorithm.

8.1 Simon's algorithm

Similar to the Deutsch-Josza and Bernstein-Vazirani problems, Simon's problem is interested in computing some property of a given function $f : \{0,1\}^n \mapsto \{0,1\}^n$, where f is specified via a black-box oracle U_f. As before, we work in the query model, meaning we only count the number of queries made to U_f, and all other unitary gates are "free". The specific property Simon's problem is interested in is as follows: Given the promise that there exists a string $s \in \{0,1\}^n$, such that $f(x) = f(y)$ if and only if $x = y$ or $x = y \oplus s$ (where $\oplus$ denotes bitwise XOR), find s.

Exercise 8.1. Suppose $f : \{0,1\}^2 \mapsto \{0,1\}^2$ is such that $f(00) = 01$, $f(01) = 10$, $f(10) = 01$, and $f(11) = 10$. What is s in this case?

8.1.1 Birthdays and a naive classical algorithm

Before giving a quantum algorithm, let us first brainstorm how one might naively try to solve Simon's algorithm classically. Intuitively, assuming $s \neq 0^n$, "all that is required" to find s is to compute $x \neq y \in \{0,1\}$ such that $f(x) = f(y)$. For then we know $x = y \oplus s$, and so

$$0^n = x \oplus x = (x \oplus y) \oplus s,$$

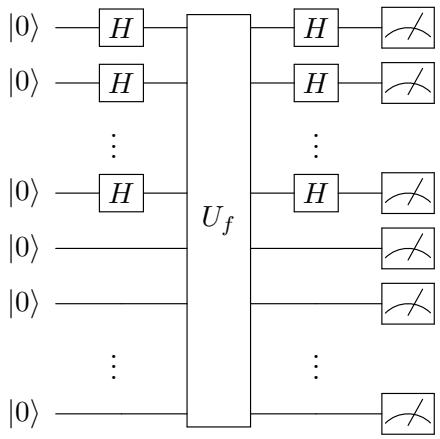
implying $s = x \oplus y$ due to our use of XOR. Thus, we are tasked with finding a *collision*, i.e. a pair of inputs yielding the same output. Let us be exceedingly naive — let us randomly

pick inputs x uniformly at random and compute $f(x)$ until we obtain a collision. What is the expected number of queries this requires if we wish to succeed with probability at least, say, $1/2$? For this, think back to the last birthday party you attended — if there were N attendees present, how large would N have to be before the odds that there existed two attendees with the same birthday was at least $1/2$? The answer is not ≈ 365, but rather $N = 23$, by the Birthday Paradox. More generally, the latter says if there are D days in the year, a collision occurs with probability at least $1/2$ if $N \in O(\sqrt{D})$. Thus, if we equate birthdays with query results $f(x)$, we expect to make $O(\sqrt{2^n})$ queries before we find two inputs with the same output with probability at least $1/2$.

Exercise 8.2. The naive algorithm above assumed $s \neq 0^n$; how can we "complete" the algorithm so that with $O(\sqrt{2^n})$ queries, with probability at least $1/2$, we output the correct s, even if s can equal 0^n?

8.1.2 Simon's algorithm

Interestingly, the "quantum" portion of Simon's algorithm is almost identical to the Bernstein-Vazirani algorithm. It is given by the following circuit, denoted C_n, acting on $2n$ qubits:



Exercise 8.3. What are the differences between the Deutsch-Josza circuit and the one above?

Exercise 8.4. Does the circuit above use phase kickback? Why or why not, intuitively?

For clarity, when a query has n output bits in C_n, we define for $f : \{0,1\}^n \mapsto \{0,1\}^n$:

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$$

for bit-wise XOR $\oplus$.

Despite its similarity to Deutsch-Josza and Berstein-Vazirani, however, this circuit is *not* the entire algorithm for Simon's problem — rather, it will be crucial to repeat C_n multiple times to obtain a set of n bit strings $\{y_i\}$, and classically postprocess the $\{y_i\}$ to find the desired string s. Thus, Simon's algorithm requires not 1 query, but $O(n)$ queries. This is revealing of a general view of quantum algorithms which is worth noting — a quantum circuit (such as C_n) generally allows one to *sample* from an output probability distribution, which might otherwise

be difficult to sample from classically. In Simon's algorithm, as we shall now see, each run of C_n allows us to sample from the uniform distribution over strings

$$Y = \{y \in \{0,1\}^n \mid s \cdot y = 0\}. \tag{8.1}$$

The ability to sample from this distribution, coupled with classical polynomial-time processing, will suffice to solve Simon's problem.

Exercise 8.5. Suppose you are given the ability to sample from the uniform distribution over Y in Equation (8.1). Why might this suffice to efficiently compute s classically?

Analysis of C_n. Let us now trace through the execution of each run of C_n. There are four timesteps before the final measurement, whose states we denote $|\psi_1\rangle, |\psi_2\rangle, |\psi_3\rangle, |\psi_4\rangle \in (\mathbb{C}^2)^{\otimes 2n}$, respectively. Clearly, $|\psi_1\rangle = |0\rangle^{2n}$ and $|\psi_2\rangle = |+\rangle^{\otimes n}|0\rangle^{\otimes n}$. Expanding this in the standard basis, we have that

$$|\psi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle|f(x)\rangle.$$

Normally, we would now apply the last set of Hadamards; however, it will be enlightening to play a technical trick first. Observe that in the circuit diagram above, the Hadamards on the first n qubits and the measurements on the last n qubits act on disjoint sets of qubits; thus, these operations commute, and the order in which we apply them is irrelevant. Thus, let us first measure the second register, obtaining some outcome $y \in \{0,1\}^n$. The key observation is that the postmeasurement state on the first register must be *consistent* with y, i.e. the first register can now only contain $x \in \{0,1\}^n$ such that $f(x) = y$. But by our promise on f, we know something about this — assuming $s \neq 0^n$, there are precisely two preimages x of y, namely x and $x \oplus s$! Our postmeasurement state, which we denote $|\psi_3'\rangle$, is hence:

$$|\psi_3'\rangle = \frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)|f(x)\rangle.$$

This demonstrates one of the strengths of quantum computation — by *postselecting* on a measurement outcome in the second register, we can collapse a superposition over the first register onto some postmeasurement state which reveals some desired structure.

Exercise 8.6. We claimed above that assuming $s \neq 0^n$, there are precisely two preimages x of y, namely x and $x \oplus s$. In other words, f is a 2-to-1 function. Convince yourself that this claim is correct.

Exercise 8.7. Assume $s = 0^n$. Is f still 2-to-1? If not, then what type of function is f?

We are now ready to apply the final set of Hadamards on the first register of $|\psi_3'\rangle$. Recall from Lecture 7 that for any $x \in \{0,1\}^n$, $H^{\otimes n}|x\rangle = (1/\sqrt{2^n}) \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y}|y\rangle$ for $\cdot$ the inner product modulo 2. Thus,

$$\begin{aligned} |\psi_4\rangle &= \frac{1}{\sqrt{2^{n+1}}} \left(\sum_{y \in \{0,1\}^n} (-1)^{x \cdot y}|y\rangle + \sum_{y \in \{0,1\}^n} (-1)^{(x \oplus s) \cdot y}|y\rangle \right) |f(x)\rangle \\ &= \frac{1}{\sqrt{2^{n+1}}} \left(\sum_{y \in \{0,1\}^n} (-1)^{x \cdot y}(1 + (-1)^{s \cdot y})|y\rangle \right) |f(x)\rangle, \end{aligned}$$

where we have used the fact that $(x \oplus s) \cdot y = x \cdot y \oplus s \cdot y$. Observe now that in the exponents, $s \cdot y \in \{0, 1\}$ (since we are working modulo 2), and $s \cdot y = 1$ leads to an amplitude of 0. Thus, we really have state

$$|\psi_4\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{s \cdot y = 0} (-1)^{x \cdot y} |y\rangle |f(x)\rangle,$$

i.e. an equal superposition (up to relative phase) of y in set Y (Equation (8.1)). Thus, measuring the first register now allows us to sample uniformly at random from Y, as claimed.

Exercise 8.8. The prefactor of $|\psi_4\rangle$ of $\sqrt{2^{-n+1}}$ suggests that $|Y| = 2^{n-1}$. Prove the latter statement holds.

Classical postprocessing. We now show how to use the ability to sample from Y to solve Simon's problem. Repeating the circuit C $n-1$ times yields strings $y_1, \ldots, y_{n-1}$, each of which satisfies $s \cdot y_i = 0$. Thus we have a linear system (where $s(i)$ denotes the ith bit of string s)

$$\begin{aligned} s(1)y_1(1) + s(2)y_1(2) + \cdots s(n)y_1(n) &= 0 \\ s(1)y_2(1) + s(2)y_2(2) + \cdots s(n)y_2(n) &= 0 \\ &\vdots \qquad (8.2) \\ s(1)y_{n-1}(1) + s(2)y_{n-1}(2) + \cdots s(n)y_{n-1}(n) &= 0, \qquad (8.3) \end{aligned}$$

where all arithmetic is done over the finite field $\mathbb{F}_2$ with elements $\{0, 1\}$. By Gaussian elimination, which works over any field, we can solve this system in cubic *arithmetic* operations, i.e. $O(n^3)$ time, to obtain the possible solutions s. Here, an "arithmetic" operation means we are counting each field operation (i.e. an addition, subtraction, multiplication, or division) as costing one "unit of time". (One can also consider the *bit complexity* of an algorithm, which further includes how many operations over individual bits each field operation requires.) Note that $s = 0^n$ is *always* a solution to the system in Equation (8.3), since the system is homogeneous (i.e. includes no constant terms). By the promise on f, we know that either $s = 0^n$ is the only solution, or there is precisely one other $s \neq 0^n$, which is what we are interested in.

Probability of failure. Thus far, the algorithm we have described appears to succeed with certainty. However, there is a catch to Gaussian elimination which we have neglected to mention — to obtain a unique non-zero solution s (if it exists), we require the strings y_i to be linearly independent over $\mathbb{F}_2$. Luckily, since we are sampling *uniformly* over Y, it can be shown that the probability of $y_1, \ldots y_{n-1}$ being independent is at least $1/4$. While this may seem a poor success probability *a priori*, it is easy to boost this to something exponentially close to 1, as you will now show.

Exercise 8.9. Show that if the probability of failure in each run of the algorithm is at most $0 < c < 1$, then with K runs of the algorithm, we can guarantee success with probability at least $1 - c^{-K}$.

8.2 Application to cryptography

As mentioned at the outset of this lecture, Simon's algorithm inspired Shor's factoring algorithm, which in turn breaks important cryptosystems such as RSA. However, as we now discuss, even Simon's algorithm can be used to break a cryptographic primitive, *assuming* an appropriate query access model. Our particular exposition will be based on [SS17] (see also [KM10, KLLNP16]).

Feistel networks. A common idea in theoretical cryptography is to use the simplest assumptions possible to build secure cryptographic primitives. One such primitive is the computation of a *random permutation*, i.e. given an input $x \in \{0,1\}^n$, output $\pi(x)$ for π a permutation on n bits chosen uniformly at random. What do we need to implement such a primitive? Intuitively, one might guess that if we assume the ability to at least perform *random functions* $f : \{0,1\}^n \mapsto \{0,1\}^n$ (not necessarily permutations), then this could be bootstrapped to simulate random permutations.

Indeed, this is the aim of a *Feistel network*. Let us focus on the case of 3-round Feistel networks. Such a network takes inputs of the form $(l, r) \in \{0,1\}^n \times \{0,1\}^n$, and each round i produces string $(l_i, r_i) \in \{0,1\}^n \times \{0,1\}^n$, defined recursively as follows for $i \in \{1,2,3\}$:

$$l_i = r_{i-1} \qquad r_i = l_{i-1} \oplus f_i(r_{i-1}). \tag{8.4}$$

Here, the $f_i : \{0,1\}^n \mapsto \{0,1\}^n$ denote the random functions, which we assume the ability to perform. Classically, Feistel networks are provably cryptographically secure, which roughly means given a black box $U : \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ performing either a Feistel network or a truly random permutation, one cannot distinguish in polynomial-time which of the two cases holds (formal definitions are beyond the scope of this course). However, it turns out that if we assume one can quantumly query U in *superposition*, then one can break the security of the 3-round Feistel network efficiently. Specifically, in this setup, one can use Simon's algorithm to efficiently distinguish whether a given U is a Feistel network or a random permutation.

Exercise 8.10. For any fixed $f : \{0,1\}^n \mapsto \{0,1\}^n$, the action of one round of a Feistel network is given by map $g_f(x, y) = (y, x \oplus f(x))$. Prove that g is a permutation on $2n$ bits. (Hint: Prove that g is a bijection.) Conclude that the 3-round network implements some random $2n$-bit permutation. Why does this not immediately yield our desired primitive of generating a "genuinely random permutation"? (Hint: What distribution over $2n$-bit permutations does a Feistel network sample from?)[1]

How to break a Feistel network. Let $U : \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be a black box which either computes a 3-round Feistel network with internal random functions $f_1, f_2, f_3 : \{0,1\}^n \mapsto \{0,1\}^n$, or a truly random permutation on $2n$ bits. Our task is to distinguish which of the two cases holds.

To apply Simon's algorithm, our goal is to ideally define a function f satisfying

$$f(x) = f(y) \text{ if and only if } x = y \oplus s \tag{8.5}$$

when U is a Feistel network. Let us begin by looking at just the first n bits of the output of $U(x, y)$, which define some function $V : \{0,1\}^{2n} \mapsto \{0,1\}^n$. In the case of a Feistel network, note that Equation (8.4) implies

$$V(x, y) = y \oplus f_2(x \oplus f_1(y)).$$

[1] We thank Or Sattath for suggesting this exercise.

Exercise 8.11. Verify the equation above.

To define f, we fix arbitrary distinct strings $\alpha, \beta \in \{0,1\}^n$. The function $f : \{0,1\} \times \{0,1\}^n \mapsto \{0,1\}^n$ will take in two inputs, a *control* bit a, and an *argument* b, such that

$$f(a,b) = \begin{cases} V(b,\alpha) \oplus \alpha & \text{if } a = 0, \\ V(b,\beta) \oplus \beta & \text{if } a = 1. \end{cases}$$

In other words, depending on the control bit a, we feed either α or β as the second input to black box U, and subsequently bitwise XOR on α or β.

Let us now see why f almost satisfies the promise of Simon's problem (i.e. Equation (8.5) when U is a Feistel network. For this, we require the following exercise.

Exercise 8.12. Suppose U is a Feistel network. Show that

$$f(a,b) = \begin{cases} f_2(b \oplus f_1(\alpha)) & \text{if } a = 0, \\ f_2(b \oplus f_1(\beta)) & \text{if } a = 1. \end{cases}$$

With this exercise in hand, we claim that the reverse implication of Equation (8.5) holds, i.e. if $x = y \oplus s$, then $f(x) = f(y)$. To see this, define mask $s = (1, f_1(\alpha) \oplus f_1(\beta))$. Then, for any input $(0,b) \in \{0,1\} \times \{0,1\}^n$ (the case of input $(1,b)$ is analogous),

$$f((0,b) \oplus s) = f(1, b \oplus f_1(\alpha) \oplus f_1(\beta)) = f_2(b \oplus f_1(\alpha)) = f(0,b).$$

We conclude that if U is a Feistel network, then

$$\exists s \in \{0,1\}^{n+1} \text{ such that, for all inputs } (a,b),\ f(a,b) = f((a,b) \oplus s). \tag{8.6}$$

Thus, the reverse implication of Equation (8.5) holds. Unfortunately, the *forward* implication of Equation (8.5) does not in general hold.

Exercise 8.13. Why is it not necessarily true that if U is a Feistel network and if $f(a,b) = f(c,d)$, then $(a,b) = (c,d) \oplus s$? (Hint: Recall the fact that in Simon's problem, the function f was 2-to-1 assuming $s \neq 0^n$. Is f above 2-to-1 in general?)

Luckily, it turns out Simon's algorithm does not need this forward direction of Equation (8.5) to hold.

We can now outline the algorithm for breaking the security of the Feistel network. For this, note that if U is *not* a Feistel network, but rather a random permutation, then the odds of there existing an s satisfying Equation (8.6) can be shown to be negligible. Thus, intuitively, given black box access to U, we can run Simon's algorithm to see if we can compute a mask s satisfying Equation (8.6); if we succeed, we conclude U is a Feistel network, and if we fail, we conclude U is a random permutation.

This outline is more formally stated below; its analysis is left as an exercise. As a first step in the analysis, one needs that given black-box access to U, one can simulate black-box access to f, which you will now show.

Exercise 8.14. Given black-box access to U, i.e. the ability to map input $|x, y\rangle$ to $|x, y \oplus U(x)\rangle$, show how to simulate black box access to f, i.e. how to simulate mapping $|u, y\rangle \mapsto |u, y \oplus f(u)\rangle$.

The algorithm of [SS17] (see also [KM10]) is as follows. It returns either FEISTEL or RANDOM to denote its guess as to the identity of U.

1. Initialize empty set S, whose purpose will be to store samples of strings y obtained via Simon's algorithm.
2. Repeat until S contains n linearly independent strings $y \in \{0,1\}^{n+1}$:
   a) If $|S| \geq 2n$, return FEISTEL.
   b) Run Simon's algorithm on the black box for f to sample string y, and add y to S.
3. (Compute s) Solve the system of equations $\{y \cdot s = 0\}$ to obtain $s \in \{0,1\}^{n+1}$.
4. (Check if s is correct) Pick input $(a, b) \in \{0,1\} \times \{0,1\}^n$ uniformly at random, and check if Equation (8.6) holds. If yes, output FEISTEL.
5. Output RANDOM.

In summary, the algorithm essentially exploits the following observation: If U is a random permutation, it will have no structure. On the other hand, if U is a Feistel network, an appropriately defined f has the structure induced by Equation (8.6). As we know from the first half of this lecture, Simon's algorithm can uncover structure of the type of Equation (8.6), hence breaking the security of Feistel networks. (Again, this assumes one is allowed to query U in superposition, which is a non-trivial assumption.)

9 The Quantum Fourier Transform

"... in mathematics you don't understand things. You just get used to them."
— John von Neumann.

It is hard to understate the impact of the *Fourier transform (FT)* on our everyday lives. Did you listen to an MP3 sound file this week? You used the Fourier Transform. Download JPEG photos of your friends off the internet? You used the Fourier Transform. Get an MRI recently? You guessed it — *Fourier Transform.* So what is the Fourier transform, and why is it so useful?

In a nutshell, the Fourier Transform allows us to efficiently decompose a signal (say, a sound wave) into its constituent frequencies. So, for example, that MP3 file which compresses your favorite song down to a few megabytes — it roughly works by applying the Fourier Transform to break down the song into its constituent frequencies, and then removing or *filtering out* the frequencies whose absence you are unlikely to notice (e.g. very high or low frequencies). Moreover, what makes this so applicable in practice is that the Fourier transform has a speedy near-linear time implementation. In particular, the Discrete Fourier transform (DFT) of a vector $|\psi\rangle \in \mathbb{C}^N$ can be implemented via the Fast Fourier Transform (FFT) algorithm using [1] $O(N \log N)$ field operations over $\mathbb{C}$. These two properties (signal decomposition and highly efficient implementation) are the one-two punch which makes the Fourier Transform so ubiquitous in everyday life.

It is thus perhaps fitting that, in the quantum setting, the *Quantum* Fourier Transform (QFT) underpins one of the greatest achievements of the field — Shor's quantum factoring algorithm. The primary focus of this lecture is hence to introduce the QFT, its implementation, and various applications. With these concepts under our belt, we will be ready to tackle the quantum factoring algorithm.

Now, we should be clear that broadly speaking, the topic of Fourier transforms is a bit confusing (this is roughly what the quote atop this lecture is alluding to). There are, in fact, *four* Fourier transforms (at least from the perspective of signal processing), depending on whether the input signal is continuous or discrete, finite/periodic or infinite/aperiodic. These go by the names of Fourier Series (continuous, finite), Discrete Fourier Transform (discrete, finite), Fourier Transform (continuous, infinite), and Discrete-Time Fourier Transform (discrete, infinite). (The interested reader is referred to [Kul02] for further details.) Here, when we talk about the FFT or QFT, we are referring to fast classical and quantum *implementations*, respectively, of the *DFT*. Thus, our "signals" are discrete and finite, encoded as vectors $|\psi\rangle \in \mathbb{C}^N$. The DFT, in turn, is a linear opeator mapping $\mathbb{C}^N$ to $\mathbb{C}^N$, representable as an $N \times N$ complex matrix. For this reason, we begin with a general class of matrices from which the DFT arises — the *Vandermonde* matrices.

9.1 From Vandermonde matrices to the Discrete Fourier Transform

Recall that given any complex number $c \in \mathbb{C}$, one can define *geometric sequence* $c^0, c^1, c^2, \ldots, c^{N-1}$. The strong structure of such sequences gives rise to nice properties — for example, the *sum* of

[1] Note that the naive implementation of the DFT is $O(N^2)$, which is essentially unusable in practice.

the sequence entries, i.e. the *geometric series* $\sum_{k=0}^{N-1} c^k$, has a closed form. What happens if we embed such sequences as rows of a matrix? We might naively expect to get a structured matrix whose properties can, like the geometric series, be analyzed. Such matrices are called *Vandermonde* matrices.

Vandermonde matrices. As a naive first attempt for $N = 3$, fix any $c \in \mathbb{C}$, and consider Vandermonde matrix

$$M_1 = \begin{pmatrix} 1 & c & c^2 \\ 1 & c & c^2 \\ 1 & c & c^2 \end{pmatrix}.$$

This is not very interesting — since all rows are the same, M is just a rank 1 embedding of the original sequence $(1, c, c^2)$. But something interesting happens when we pick a *different* c for each row. For any distinct $c_1, c_2, c_3 \in \mathbb{C}$, the matrix

$$M_2 = \begin{pmatrix} 1 & c_1 & c_1^2 \\ 1 & c_2 & c_2^2 \\ 1 & c_3 & c_3^2 \end{pmatrix}.$$

turns out to have linearly independent rows. Thus, it is full rank (and hence invertible, assuming M_2 is square).

Exercise 9.1. Suppose above that $c_1 = c_2$, but $c_1 \neq c_3$. Is M_2 full rank? What if we add a fourth row with entries 1, c_4, c_4^2, for c_4 distinct from c_1, c_2, c_3?

To help give intuition as to the usefulness of Vandermonde matrices, one important application (which we revisit at the end of this section) is evaluating polynomials $p : \mathbb{C} \mapsto \mathbb{C}$ at the points (in the case of M_2) $c_1, c_2, c_3 \in \mathbb{C}$.

Exercise 9.2. Consider polynomial $p(x) = 4x^2 - 3x + 1$. Encoding the coefficients of p as vector $|\psi\rangle = (1, -3, 4)^T$, show that the ith coordinate of $M_2|\psi\rangle$ contains $p(c_i)$.

Exercise 9.3. Using the previous exercise, prove the Interpolation Theorm, which states: Any set of $d+1$ point-value pairs $\{(c_i, p(c_i))\}_{i=1}^{d+1}$ identifies a unique degree-d polynomial p, assuming all c_i are distinct.

Returning to the DFT. Since we can choose *any* complex values for the $\{c_i\}$, let us choose the "quintessential" set of N complex numbers, the *Nth roots of unity*. These are generated by taking powers of the "principal" Nth root of unity, $\omega_N := e^{2\pi i/N}$, i.e.

$$\omega_N^0, \omega_N^1, \ldots, \omega_N^{N-1}.$$

For example, the principal 4th root of unity is $\omega_4 = e^{\pi/2} = i$, and the 4th roots are $(1, i, i^2 = -1, i^3 = -i)$.

Exercise 9.4. Draw the 4th roots of unity on the complex unit circle. More generally, what do the Nth roots of unity look like on the circle?

Exercise 9.5. Why are there only N Nth-roots of unity? More precisely, why is sequence $(\omega_N^0, \omega_N^1, \ldots, \omega_N^{N-1}, \omega_N^N)$ redundant?

Plugging the Nth roots of unity into a Vandermonde matrix, we finally arrive at the order-N DFT:

$$\mathrm{DFT}_N = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & (\omega_N^2)^2 & \cdots & (\omega_N^2)^{N-1} \\ 1 & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & (\omega_N^{N-1})^2 & \cdots & (\omega_N^{N-1})^{N-1} \end{pmatrix}.$$

Exercise 9.6. Why is the first row of DFT_N all ones?

Exercise 9.7. What is DFT_2? DFT_3?

Since the roots of unity ω_N^k are distinct, the properties of Vandermonde matrices tell us DFT_N (which is square) is invertible. Can we write down its inverse? Remarkably, as shown in the exercise above, DFT_2 is just (up to scaling) the 2×2 Hadamard matrix we have grown quite fond of. And this is no coincidence — not only is DFT_N invertible, it is actually unitary. Thus, its inverse is simply $\mathrm{DFT}_N^\dagger$ (up to scaling).

Exercise 9.8. Prove that the rows of DFT_N are orthogonal. You may find it useful to look up the closed formula for geometric series, which also applies to complex numbers.

Exercise 9.9. Prove that a square matrix is unitary if and only if its rows (equivalently, columns) form an orthonormal set. Conclude that, up to scaling, DFT_N is unitary.

Exercise 9.10. What is the correct scaling required to make DFT_N unitary?

Since DFT_N is unitary, we can in principle apply it as a quantum gate to a quantum state! In other words, given $|\psi\rangle \in \mathbb{C}^N$, Nature allows the mapping $\mathrm{DFT}_N|\psi\rangle \in \mathbb{C}^N$. The only problem is that for an n-qubit system, the dimension $N = 2^n$ is exponential in n. So the question is: *Can we implement* DFT_N *via a quantum circuit of size polynomial in the number of qubits,* n*?* In the next section, we show the answer is *yes*. It is this quantum implementation of the (normalized) DFT which is henceforth denoted the *Quantum Fourier Transform (QFT)*.

Exercise 9.11. The fastest classical implementation of the DFT is the Fast Fourier Transform (FFT), which requires $O(N \log N)$ time. We shall show in Section 9.2 that the quantum implementation of the DFT, the QFT, requires only time $O(\mathrm{polylog}(N))$. Thus, on the face of it, it looks like one quantum computers exponentially speed up computation of the DFT. Why is this a misleading claim? (Hint: How does the output of the QFT differ from that of the FFT?))

Finally, back to polynomial evaluation. Before closing this section, let us return to our aside into polynomial evaluation.

Exercise 9.12. Using the FFT, argue that an arbitrary polynomial of degree $N-1$ can be evaluated at all Nth roots of unity in just $O(N \log N)$ time. What would be the naive runtime for this if we do not appeal to the FFT?

Exercise 9.13. Given two degree $N-1$ polynomials p and q, suppose we use the FFT (as in the previous exercise) to evaluate p and q at the Nth roots of unity, i.e. to obtain two lists of point-value pairs

$$\begin{array}{llll} (1, p(1)), & (\omega_N, p(\omega_N)), & \ldots & , (\omega_N^{N-1}, p(\omega_N^{N-1})), \text{ and} \\ (1, q(1)), & (\omega_N, q(\omega_N)), & \ldots & , (\omega_N^{N-1}, q(\omega_N^{N-1})). \end{array}$$

Using the fact that for any $(x, p(x))$ and $(x, q(x))$, the tuple $(x, p(x)q(x))$ denotes the value of the product of p and q on input x, show that the coefficients of polynomial $pq(x) := p(x)q(x)$ can be computed in time $O(N \log N)$. (Hint: You will also need the Interpolation Theorem.) How does this compare with the naive runtime for multiplying out two degree $N-1$ polynomials p and q (i.e. without using the FFT)?

Exercise 9.14. Use the FFT-based algorithm you derived in the previous exercise to compute the product of polynomials $p(x) = x^2 - x + 1$ and $q(x) = 2x^2 + 3x$. (Hint: Since $N = 3$ in this case, you can do all computations by hand.)

9.2 The Quantum Fourier Transform (QFT)

Suppose we have an n-qubit system. We now show how to implement the unitary operation $\frac{1}{\sqrt{N}}\mathrm{DFT}_N$ for $N = 2^n$ via $\mathrm{polylog}(N) = \mathrm{poly}(n)$ 1- and 2-qubit gates. To avoid having to repeat the normalization factor each time, we henceforth define $\mathrm{QFT}_N := \frac{1}{\sqrt{N}}\mathrm{DFT}_N$.

The N=4 case. We already know that $\mathrm{QFT}_2 = H$ (i.e. when $n = 1$), so what is QFT_4 ($n = 2$)? Recall that any linear map is completely specified by its action on a basis, such as the standard basis. Thus, since $\mathrm{QFT}_4|j\rangle$ extracts the jth column of QFT_4, we have:

$$\mathrm{QFT}_4|j\rangle \quad = \quad \frac{1}{2}\sum_{k=0}^{3} e^{2\pi i j \frac{k}{4}}|k\rangle, \tag{9.1}$$

where we have intentionally grouped $k/4$ together for the following reason. Recall from your programming courses that, typically, division of two integers a/b is not "cheap", but division by a power of 2, $a/2^k$, *is* cheap — it simply shifts over the bit representation of a to the right by k positions. For example, translating $3/2$ to binary means we shift the bit representation of 3, 11_2, to the right one position to obtain 1.1_2, which should be interpreted as $1 \cdot 2^1 + 1 \cdot 2^{-1}$.

Exercise 9.15. What is the binary representation of $14/29$? Expand it out in terms of powers of 2.

Rewriting $|k\rangle$ in binary as $|k_1k_2\rangle$, we thus have

$$\begin{aligned}
\mathrm{QFT}_4|j\rangle &= \frac{1}{2}\sum_{k=0}^{3} e^{2\pi ij(0.k_1k_2)}|k_1k_2\rangle && (9.2)\\
&= \frac{1}{2}\sum_{k=0}^{3} e^{2\pi ij\left(k_1\cdot 2^{-1}+k_2\cdot 2^{-2}\right)}|k_1k_2\rangle && (9.3)\\
&= \frac{1}{2}\left(\sum_{k_1=0}^{1} e^{2\pi ijk_1\cdot 2^{-1}}|k_1\rangle\right)\left(\sum_{k_2=0}^{1} e^{2\pi ijk_2\cdot 2^{-2}}|k_2\rangle\right) && (9.4)\\
&= \frac{1}{2}\left(|0\rangle + e^{2\pi i\frac{j}{2}}|1\rangle\right)\left(|0\rangle + e^{2\pi i\frac{j}{2^2}}|1\rangle\right) && (9.5)\\
&= \frac{1}{2}\left(|0\rangle + e^{2\pi i(0.j_2)}|1\rangle\right)\left(|0\rangle + e^{2\pi i(0.j_1j_2)}|1\rangle\right). && (9.6)
\end{aligned}$$

Exercise 9.16. In Equation (9.6), why can we omit the bit j_1 in the phase $e^{2\pi i(0.j_2)}$?

Now comes the key observation: Applying the Hadamard to single qubit standard basis state $|j\rangle \in \mathbb{C}^2$ yields

$$H|j\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.j)}|1\rangle\right). \tag{9.7}$$

Exercise 9.17. Convince yourself that Equation (9.7) holds.

Thus, returning to attempting to implement DFT_4, we might naively try

$$H_1 \otimes H_2|j\rangle = H_1 \otimes H_2|j_1\rangle|j_2\rangle = \frac{1}{2}\left(|0\rangle + e^{2\pi i(0.j_1)}|1\rangle\right)\left(|0\rangle + e^{2\pi i(0.j_2)}|1\rangle\right). \tag{9.8}$$

This is *almost* what we want (up to swapping the two output registers), except we are missing the bit j_2 in the phase $\exp(2\pi i(0.j_1j_2))$ of Equation (9.6). Inserting a phase $\theta \in \mathbb{R}$, however, is easily accomplished (at least in theory) via gate

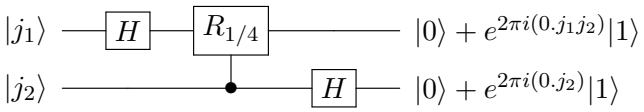
$$R_\theta = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i\theta} \end{pmatrix}. \tag{9.9}$$

Exercise 9.18. Prove that R_θ is unitary for any $\theta \in \mathbb{R}$. What is its action on input $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$?

Exercise 9.19. Consider circuit $(R_{1/4} \otimes I)(H \otimes H)|j\rangle$. Why does this not reproduce $\mathrm{DFT}_4|j\rangle$ (up to swapping the output registers)?

The exercise above highlights that in order to insert phase $\exp(2\pi i(0.0j_2))$ into the *first* term (and hence first qubit) of Equation (9.8), the gate $R_{1/4}$ must also depend on the *second* qubit

state, $|j_2\rangle$. We thus arrive at our QFT_4 circuit by adding a *controlled-$R_{1/4}$* gate (below, we omit swapping the two output qubits for simplicity):



Exercise 9.20. Convince yourself that, up to swapping the output qubits, this is a correct QFT_4 circuit.

The general $N = 2^n$ case. The general $N = 2^n$ case now follows analogously to $N = 4$. Below, we state the action of QFT_N on a standard basis state, followed with the corresponding circuit. Proofs are left as an exercise. First, for standard basis state $|j\rangle \in \mathbb{C}^{2^n}$,

$$\mathrm{QFT}_{2^n}|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j \frac{k}{2^n}} |k\rangle \tag{9.10}$$

$$= \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i(0.j_n)}|1\rangle\right) \left(|0\rangle + e^{2\pi i(0.j_{n-1}j_n)}|1\rangle\right) \cdots \left(|0\rangle + e^{2\pi i(0.j_1 j_2 \cdots j_n)}|1\rangle\right) \tag{9.11}$$

$$=: \frac{1}{\sqrt{2^n}} |\phi_n\rangle|\phi_{n-1,n}\rangle \cdots |\phi_{1,\ldots,n}\rangle, \tag{9.12}$$

where we have defined $|\phi_{k,\ldots,n}\rangle := |0\rangle + e^{2\pi i(0.j_k \cdots j_n)}|1\rangle$ for convenience.

Exercise 9.21. Prove Equation (9.11).

The QFT_N circuit is (up to reordering of output qubits):

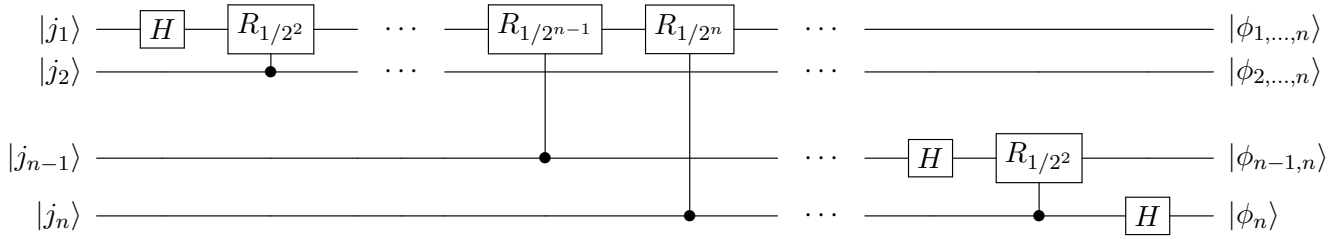
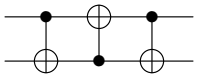


Figure 9.1: The QFT_N circuit (omitting final reordering of output qubits) acting on n qubits, where $N = 2^n$.

Exercise 9.22. Note the output qubits of the circuit above are in the opposite order than in Equation (9.12). Given the ability to perform the two-qubit SWAP gate, which maps any bipartite standard basis state $|i\rangle|j\rangle$ to $|j\rangle|i\rangle$, show how to rearrange the output of the circuit to match Equation (9.12).

Exercise 9.23. Prove that the following sequence of CNOT gates implements the two-qubit SWAP gate:



Do you need to explicitly prove it for any choice of product input state $|\psi\rangle|\phi\rangle \in \mathbb{C}^4$?

Exercise 9.24. How does the SWAP gate act on each of the four Bell basis states for $\mathbb{C}^4$?

Exercise 9.25. Show the circuit of Figure 9.21, together with your use of postprocessing via SWAP gates, correctly computes QFT_N. This quantum circuit *implementation* of DFT_N is what we call here the QFT_N.

Exercise 9.26. What is the size of the circuit in Figure 9.21, where "size" is defined by the number of 1- and 2-qubit gates? (Hint: Think about arithmetic series.) How much overhead does the final postprocessing via SWAP gates incur, and does this change the overall circuit size estimate for Figure 9.21? Give your estimates in terms of both N and n.

Let us close this section by again stressing that, up to renormalization, DFT_N and QFT_N perform precisely the same map. However, while the former is an abstract linear map, the latter is an explicit quantum circuit *implementation* of DFT_N which requires $O(\log^2 N) \in O(n^2)$ gates. In contrast, the best known classical implementation for the DFT_N is the FFT, which requires $O(N \log N)$ time. Thus, it *looks* like QFT_N gives an exponential speedup over the FFT. But this is entirely misleading, as the FFT and QFT_N represent their output in entirely different formats — the former gives an explicit list of all N entires of the output vector $|\psi\rangle$, whereas the latter gives $|\psi\rangle$ as a physical quantum state on $\log N$ qubits, whose amplitudes *cannot* in general be individually recovered.

9.3 Quantum Phase Estimation (QPE)

With QFT_N in hand, we are in principle in a position to discuss Shor's factoring algorithm. However, it will be instructive to take a top-down approach — to start by discussing, at a high level, the *class* of algorithms factoring falls into. Specifically, the heart of the factoring algorithm is a special case of the following general problem.

Definition 9.27 (Quantum Phase Estimation (QPE))**.**

- *Input: A quantum circuit implementing a unitary operator* $U \in \mathrm{L}\left((\mathbb{C}^2)^{\otimes n}\right)$*, and eigenvector* $|\psi_\theta\rangle$ *satisfying* $U|\psi_\theta\rangle = e^{2\pi i\theta}|\psi_\theta\rangle$*, for some* $\theta \in [0,1)$*.*
- *Output: Phase* $\theta \in [0,1)$*.*

Exercise 9.28. Prove that a normal operator U is unitary if and only if all of its eigenvalues have form $e^{i\theta}$.

Exercise 9.29. How would you solve QPE classically, i.e. given full written matrix and vector descriptions of U and $|\psi\rangle$?

Despite its arguably dull appearance, QPE is an important problem. Not only is it at the heart of the factoring algorithm, but QPE has applications throughout quantum algorithms, from quantum walks to solving linear systems of equations. We will briefly comment on such applications in Section 9.3.1. Before then, however, take a moment to break down the statement of Definition 9.27 in your mind, and allow the inevitable myriad of questions to arise — *Can*

we solve QPE in general? What if θ has no finite representation? How do we magically get our hands on an eigenvector $|\psi_\theta\rangle$? And what if we cannot compute such an eigenvector efficiently?

9.3.1 Applications of QPE

To motivate QPE, we now briefly mention three applications, and give a flavor of how QPE is applied.

1. **Factoring.** As we shall see in a subsequent lecture, factoring classically reduces to the *order-finding* problem: Given $x, N \in \mathbb{Z}^+$, what is the smallest $r \in \mathbb{Z}^+$ such that $x^r \equiv 1 \mod N$? For fixed x, we can define unitary U_x to have action $U_x|y\rangle = |xy \mod N\rangle$. Then, Shor's algorithm applies QPE to U_x, and classically postprocesses the result to find the order r.

2. **Quantum random walks.** A classical *random walk* is exactly what it sounds like — given a graph $G = (V, E)$ and a starting vertex $v \in V$, in each time step we pick a random neighbor of our current vertex and "walk" there. A general framework for implementing *quantum* walks uses QPE as follows. Given a vector $|\psi\rangle$, as a subroutine in the quantum walk, we would like to *reflect* about $|\psi\rangle$, i.e. to apply operator $R_\psi = 2|\psi\rangle\langle\psi| - I$.

   Exercise 9.30. Show that R_ψ is indeed a reflection about $|\psi\rangle$, i.e. that $R_\psi|\psi\rangle = |\psi\rangle$, and for any $|\psi^\perp\rangle$ orthogonal to $|\psi\rangle$, $R_\psi|\psi^\perp\rangle = -|\psi^\perp\rangle$.

   One approach for implementing R_ψ, roughly, is to set up a unitary map U for which $|\psi\rangle$ is an eigenvector. Then, to simulate application of R_ψ to an arbitrary state $|\phi\rangle$, one uses QPE to decompose $|\phi\rangle$ into its components: The component proportional to $|\psi\rangle$, and the component orthogonal to $|\psi\rangle$. With this "on-the-fly" decomposition in hand, one can "inject" a phase of $+1$ or -1 as needed, respectively.

3. **Powers of unitaries and linear systems.** Suppose we have access to a circuit implementation of unitary U, but want to instead apply $\sqrt{U}$. Can we do it using just U? QPE gives us an approach for this, based on the following exercise.

   Exercise 9.31. If the kth eigenvalue of U is $e^{i\theta_k}$, what is the kth eigenvalue of $\sqrt{U}$?

   Thus, we can use QPE to, roughly, extract the eigenvalue phases of U into a separate register (i.e. as strings $|\theta_k\rangle$), and coherently apply the square root function to "update" or "customize" the eigenvalue phases to $|\sqrt{\theta_k}\rangle$. Undoing the QPE estimation circuit then effectively "reinserts" the eigenvalues back into the operator to simulate $\sqrt{U}$.

   A striking application of this idea of "eigenvalue surgery" is to solve linear systems of equations $A|\psi\rangle = |b\rangle$ (i.e. given A, $|b\rangle$, what is $|\psi\rangle$?). The setup there is more complicated, as the coefficient matrix A need not even be Hermitian (never mind unitary). But the rough premise is similar — to simulate the inverse A^{-1}, one uses QPE to extract the eigenvalues λ_i of A (via some appropriate unitary implementation of A), inverts the λ_i "manually" to $1/\lambda_i$, and then uses *postselection* to "reinsert" the edited eigenvalues "back into A".

9.3.2 Quantum algorithms for QPE

We now give quantum algorithms for QPE. For simplicity, we assume the desired phase $\theta \in [0,1)$ can be represented exactly using n bits, i.e. in binary we write $\theta = (0.\theta_1 \cdots \theta_n)_2$. As a result, $2^n\theta = (\theta_1 \cdots \theta_n)_2$. Our aim is, given U and $|\psi_\theta\rangle$, to prepare a quantum n-qubit register containing $|\theta_1 \cdots \theta_n\rangle = |2^n\theta\rangle$.

The algorithm. Recall that Equation (9.11) says

$$\begin{aligned} \mathrm{QFT}_{2^n}|\theta\rangle &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k \frac{\theta}{2^n}} |k\rangle && (9.13) \\ &= \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i (0.\theta_n)}|1\rangle\right) \left(|0\rangle + e^{2\pi i (0.\theta_{n-1}\theta_n)}|1\rangle\right) \cdots \left(|0\rangle + e^{2\pi i (0.\theta_1\theta_2\cdots\theta_n)}|1\rangle\right) && (9.14) \end{aligned}$$

Thus, if we could prepare the right hand side above, applying $\mathrm{QFT}_N^\dagger$ would recover $|\theta\rangle$, as desired. There are two ways to do this, depending on how much we want to assume about what we can do with U.

More restrictive: Assume we can apply controlled-U^k gates for any integer $k \geq 0$. The following exercise suggests an approach for solving QPE in this setting.

Exercise 9.32. Show that for non-negative integer k, $U^k|\psi_\theta\rangle = e^{2\pi i k\theta}|\psi_\theta\rangle$.

But now we claim it is easy to prepare the right-hand side of Equation (9.13): Prepare an equal superposition over all $|k\rangle$, and then apply the controlled-U^k gate:

$$|0^n\rangle|\psi_\theta\rangle \xmapsto{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle|\psi_\theta\rangle \xmapsto{c-U^k} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k\theta}|k\rangle|\psi_\theta\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k\frac{(2^n\theta)}{2^n}}|k\rangle|\psi_\theta\rangle \xmapsto{\mathrm{QFT}_N^\dagger} |2^n\theta\rangle|\psi_\theta\rangle,$$

where the first map applies local Hadamards to the first register, the second the controlled-U^k gates with the first register as control and the second register as target, and the last the inverse QFT_N to the first register.

Less restrictive: Assume we can apply controlled-U^{2^k} gates for any integer $k \geq 0$. Suppose we cannot raise U to *arbitrary* powers, but at least we can raise it to powers of 2. Can we still solve QPE? The answer is *yes* (albeit slightly more involved than the previous case), and is given in the following exercise.

Exercise 9.33. Show that by leveraging the expansion in Equation (9.14) instead of Equation (9.13), one can still solve QPE, even with the restricted ability of raising U to powers of 2. (Hint: Each choice of a power of 2 will allow you to prepare a specific term in the tensor product on the right hand side of Equation (9.14).)

Runtime and discussion. There are two major issues we have thus far swept under the rug: What if θ is not representable in n bits, and what is the size of the circuit implementing QPE?

Irrational $\theta \in [0,1)$. Suppose now θ's binary representation is longer than n bits (potentially infinitely long). Can we at least approximate θ to its n most significant bits? *Yes*, and in fact,

the exact same circuit as above works, with the following tweaks. Suppose we wish to compute the n most significant bits of θ's binary expansion with probability of success at least $1-\epsilon$ for $\epsilon > 0$. Then, one can show that running the same procedure as before suffices, *so long as* we are willing to slightly expand the size of the first register to $n + \lceil \log(2 + \frac{1}{2\epsilon}) \rceil$ qubits (all initialized to $|0\rangle$ before the circuit starts).

Size of circuit. It is easy to see that the two dominant components of the QPE algorithm are the QFT_N circuit and the controlled-U^k gate. We know the size of the first; but what about the size of the second? Unfortunately, here we hit a snag — to get n bits of precision in our estimate of θ, we need the gate $c-U^{2^n}$. But in general it will be impossible, given an arbitrary U, to compute U^k efficiently if k is superpolynomial in the input size.

Exercise 9.34. Suppose given an arbitrary unitary U on n qubits, we can compute U^{2^n} efficiently, i.e. with a quantum circuit of size polynomial in n. Prove that this implies quantum computers can efficiently count the number of solutions to a SAT formula $\phi : \{0,1\}^n \mapsto \{0,1\}$ (formally, the complexity class BQP would contain #P). Since it is believed unlikely that quantum computers can solve NP-hard problems (never mind #P problems), conclude it is unlikely for such a powering circuit for U to exist.

Thus, for arbitrary U, the best we can hope to do is estimate θ within $O(\mathrm{polylog}(n))$ bits of precision (since then the largest power of U we need is polynomial in n).

Exercise 9.35. Suppose we approximate $\theta = 0.\theta_1\theta_2\ldots \in [0,1)$ by its first $\lceil \log n \rceil$ bits, i.e. $\widetilde{\theta} = 0.\theta_1 \ldots \theta_{\lceil \log n \rceil}$. What is the largest the additive error $\left|\theta - \widetilde{\theta}\right|$ can be?

The exercise above shows that logarithmically many bits of precision suffices to get inverse polynomial additive error. Note that this suffices in certain applications, since (roughly speaking) polynomial-size quantum circuits cannot distinguish states which are closer than an inverse polynomial in distance (trace distance for density operators, which reduces to Euclidean distance for pure state vectors) anyway.

Conclusion for QPE. We conclude our discussion on QPE by reiterating that, to estimate θ within n bits of precision, we require the controlled-U^{2^k} gate for $k \in O(n)$. The overall runtime of QPE will thus scale with the size of the QFT_N circuit and the powering-of-U circuit, the latter of which generally scales exponentially with k (the precise size is naturally context specific, i.e. depending on which U we have). Second, for the factoring algorithm, being able to raise U to exponentially large powers *will* be important — luckily, as we will see there, the specific U needed will implement a *classical* operation whose precise structure allows such fast exponential powering.

10 Shor's quantum factoring algorithm

> *"Euclid taught me that without assumptions there is no proof. Therefore, in any argument, examine the assumptions."*
> — Eric Temple Bell.

In the previous lecture, we introduced the Quantum Fourier Transform (QFT), and applied it to the Quantum Phase Estimation (QPE) problem. In this lecture, we show how to use QPE to derive a crown jewel of quantum computation — the quantum factoring algorithm. This algorithm, developed by Peter Shor in 1994, forced the classical computer science community to re-evaluate one of its core assumptions. Namely, in 1978, Rivest, Shamir, and Adleman published the public-key cryptosystem known as RSA, which to this day remains widely used in practice. And while the field had grown comfortable assuming RSA is secure, its Achilles heel was hiding in plain sight — the abstract of the original RSA paper itself reads:

> *"The security of the system rests in part on the difficulty of factoring the published divisor, n".*

Indeed, the standard assumption leading into the early 1990's was that factoring is "hard". And this is precisely where Peter Shor made his mark — he showed that a polynomial-time quantum computer could factor a composite integer N into its prime factors, thus breaking RSA.

At a high level, the factoring algorithm proceeds as follows: First, the problem of *integer factorization* is reduced to the problem of *order finding*[1]. Then, the QPE algorithm is used to sample from a certain desired distribution. Finally, these samples are postprocessed using a classical algorithm known as *continued fractions* to solve the order finding problem. Before getting into these details, however, let us begin by formally defining the factoring problem and understanding its context.

10.1 The integer factorization problem

Recall that the Fundamental Theorem of Arithmetic states that any positive integer N has a prime factorization, i.e.

$$N = p_1^{k_1} p_2^{k_2} \cdots p_m^{k_m}, \tag{10.1}$$

for distinct primes p_i and positive integers k_i, and moreover this decomposition is unique. Thus, intuitively the integer factorization problem asks one to, given N, output the prime factors $\{p_1, \ldots, p_m\}$.

Exercise 10.1. Consider the following brute force algorithm for finding a factor p_i: Iteratively try to divide N by each of the numbers in sequence $(2, 3, 4, \ldots, \lfloor\sqrt{N}\rfloor)$, and output the first divisor found. Clearly, this algorithm requires $O(\sqrt{N})$ field operations over $\mathbb{R}$. Why is this not a polynomial-time factoring algorithm? (Hint: What is the number of bits required to represent the input, N? Is an $O(\sqrt{N})$-time algorithm polynomial with respect to this number of bits?)

[1] For clarity, this reduction was known prior to Shor's algorithm, due to the 1976 work of Miller [Mil76].

Exercise 10.2. Given the prime factors $\{p_1, \ldots, p_m\}$, show how to also compute the multiplicities $\{k_1, \ldots, k_m\}$ using polylog(N) field operations. Hint: Prove first that there are at most polylog(N) primes (including multiplicities) in the prime factorization of N.

A formal definition. Although the intuitive definition of the factoring problem is clear, recall that formally the complexity classes P and NP are sets of *decision* problems, meaning roughly that each possible input to the problem is either a YES or NO instance. Thus, we formally define factoring as follows.

Definition 10.3 (FACTOR)**.**

- *Input: A positive integer $N > 1$, and threshold $K \in \mathbb{Z}^+$.*
- *Output: Does N have a prime factor smaller than K?*

Exercise 10.4. Show how a subroutine for solving the decision problem FACTOR can be bootstrapped to efficiently compute the smallest non-trivial prime factor of N.

Complexity theoretic considerations. Interestingly, while FACTOR is not known to be in P, its close cousin, the primality testing problem, *is* in P. The latter is defined: *Given $N \in \mathbb{Z}^+$, is N composite (i.e. not prime)?* While this does not, as far as is known, allow us to also solve FACTOR, it does imply that FACTOR $\in$ NP $\cap$ co-NP.

Exercise 10.5. Why is FACTOR trivially in NP?

Exercise 10.6. Why is FACTOR in co-NP? (Hint: Use the Fundamental Theorem of Arithmetic.)

Exercise 10.7. Prove that if a language $L \in$ NP $\cap$ co-NP is NP-complete, then NP = co-NP.

Combining the exercises above, we conclude that if FACTOR is NP-complete, then NP = co-NP. The latter equality, however, is known to collapse the Polynomial-Time Hierarchy (PH) to its first level, a consequence widely believed almost as unlikely as P = NP. Thus, FACTOR is strongly expected not to be NP-complete.

NP-intermediate problems. While at first glance this may seem an unsatisfying state of affairs, there is a silver lining: FACTOR is one of the few known natural candidate "NP-intermediate" problems. Here, an NP-intermediate problem is a language in NP which is neither in P nor NP-complete. Of course, noone can *prove* that FACTOR is NP-intermediate. However, what we *can* prove (or more accurately, what Richard Ladner proved in 1975) is that, assuming P $\neq$ NP, NP-intermediate problems do exist.

Extended Church-Turing thesis. The backbone on which essentially all of theoretical computer science rests is the *Church-Turing Thesis*. This thesis posits that a Turing machine (TM) can simulate any other model of computing. (Note this is a *thesis*, not a proven theorem.) In other words, TMs tell us everything we need to know about computing. Except that the Church-Turing thesis says nothing about the *efficiency* with which a TM can simulate other models.

For this, we rely on the (less commonly accepted) *Extended Church-Turing Thesis*, which states that TMs can *efficiently* simulate any *physically realizable* model of computing.

The added constraints of "efficiency" and "physical realizability" of the Extended Church-Turing Thesis imply that, thanks to the quantum factoring algorithm, we have arrived at an exciting crossroads, at which one of the following three statements must be *false*:

1. The Extended Church-Turing Thesis is true.
2. FACTOR cannot be solved efficiently on a classical computer.
3. Large-scale universal quantum computers can be built.

Exercise 10.8. Convince yourself that the truth of any two of these statements implies the falsehood of the remaining statement. Where do the keywords "efficiently" and "physically realizable" from the Extended Church-Turing Thesis fit into your arguments?

10.2 The factoring algorithm

Having formally defined FACTOR, we discuss the quantum factoring algorithm, which consists of three high-level steps: Reducing FACTOR to order-finding (Section 10.2.1), and solving order-finding by combining QPE (Section 10.2.2) and continued fractions (Section 10.2.3).

10.2.1 Reducing FACTOR to order-finding

We begin by formally defining order-finding.

Definition 10.9 (Order-finding (ORDER))**.**

- *Input: Positive integers N and $1 \leq x < N$, with x co-prime to N.*
- *Output: The least positive integer r such that $x^r \equiv 1 \mod N$.*

Exercise 10.10. As stated, ORDER is not a decision problem. How can it be converted to one?

The reduction. We now show how FACTOR reduces to ORDER. We begin with a simple observation, given as the following exercise. Define $\gcd(x, y)$ as the greatest common divisor of $x, y \in \mathbb{Z}$. Recall that x and y are *co-prime* if $\gcd(x, y) = 1$.

Exercise 10.11. Let $x \in \mathbb{Z}$ be a non-trivial solution to equation $(x-1)(x+1) \equiv 0 \mod N$, where non-trivial means $x \not\equiv 1 \mod N$ and $x \not\equiv -1 \mod N$. Show that either $\gcd(x-1, N)$ or $\gcd(x+1, N)$ is a non-trivial factor of N. (Hint: Observe that, by assumption, N cannot divide either $x-1$ nor $x+1$.)

The exercise above gives us a route for factoring a given $N \in \mathbb{Z}^+$ — namely, we "just" need to find a non-trivial x satisfying $x^2 \equiv 1 \mod N$. The only question is — where do get our hands on such an x? To solve this difficult-sounding problem, we resort to the most advanced of algorithmic techniques — simply pick an x at random. More accurately, pick (uniformly at random) an integer x in set $Z_N := \{0, 1, \ldots, N-1\}$ which is co-prime to N. It turns out that with high probability, the *order* of x, i.e. the smallest r such that $x^r \equiv 1 \mod N$, allows us

to construct the solution to $x^2 \equiv 1 \mod N$ we are looking for. Before formally proving that random sampling of x works, let us see why the order r of x helps.

Exercise 10.12. Let r be an even positive integer satisfying $x^r \equiv 1 \mod N$. Give a y satisfying $y^2 \equiv 1 \mod N$. (Hint: Note that r is even by assumption.)

Exercise 10.13. Does your y from the previous exercise necessarily allow you to extract a factor of N? (Hint: Is it necessarily true that y will be a *non-trivial* solution to $y^2 \equiv 1 \mod N$?)

The last exercise above suggests that simply picking an x with even order r will not suffice — we will need slightly more structure.

Finding a non-trivial solution to $x^2 \equiv 1 \mod N$. The proof that random sampling can lead to a non-trivial solution for $x^2 \equiv 1 \mod N$ requires three well-known facts, which we first state.

Fact 10.14 (Fermat's Little Theorem (FLT))**.** *For any prime p and integer a, $a^p \equiv a \mod p$. Moreover, if p does not divide a, then $a^{p-1} \equiv 1 \mod p$.*

Fact 10.15 (Chinese Remainder Theorem (CRT))**.** *Let $\{m_1, \ldots, m_n\}$ be a set of integers larger than 1, each pair of which is co-prime. Then, for any integers $\{a_1, \ldots, a_n\}$, the linear system of equations*

$$\begin{aligned} x &\equiv a_1 \mod m_1 && (10.2)\\ x &\equiv a_2 \mod m_2 && (10.3)\\ &\vdots && (10.4)\\ x &\equiv a_n \mod m_n && (10.5) \end{aligned}$$

has a solution $x \in \mathbb{Z}$. Moreover, all solutions are equivalent modulo $M = m_1 \cdots m_n$, implying there is a unique solution $x \in \{0, \ldots, M-1\}$.

Fact 10.16 (Multiplicative group Z_p is cyclic)**.** *Fix any positive prime $p \in \mathbb{Z}$. Then, there exists a generator $g \in Z_p$, such that any non-zero element $e \in Z_p$ can be written $g^k \equiv e \mod p$ for some $k \in \{1, \ldots, p-1\}$.*

Exercise 10.17. Use Fermat's Little Theorem to prove that there is a bijection between elements e and powers k in Fact 10.16 above. Conclude that picking e uniformly at random is equivalent to picking k uniformly at random. (For this exercise, do not assume *a priori* that $k \in \{1, \ldots, p-1\}$ as stated in Fact 10.16.)

Exercise 10.18. Let r be the order of x modulo N, and consider $r' > r$ satisfying $x^{r'} \equiv 1 \mod N$. Prove that r divides r'.

The main theorem of this section is the following. Its proof requires Lemma 10.25, which is stated and proven subsequently.

Theorem 10.19. *Let N be an odd, composite natural number. Define $Z_N^* := \{x \in Z_N \mid x \text{ is co-prime to } N\}$. Choose $x \in Z_N^*$ uniformly at random. Then, with probability at least $1/2$, the order r of x is even and satisfies*

$$x^{r/2} \not\equiv -1 \mod N. \quad (10.6)$$

Exercise 10.20. Why does Theorem 10.19 allow us to factor N with probability at least $1/2$, assuming we have an oracle for order-finding? In your answer, make sure you account for the fact that Theorem 10.19 does not explicitly state that $x^{r/2} \not\equiv 1 \mod N$.

Exercise 10.21. Is $0 \in Z_N^*$? What is Z_p^* for prime p?

Proof of Theorem 10.19. We prove the claim for the special case relevant to RSA (Section 10.3), i.e. $N = pq$ for distinct primes p and q. The proof can be generalized to arbitrary $N = p_1^{k_1} \cdots p_m^{k_m}$ for distinct primes $p_1, \ldots, p_m$.

To begin, the only thing we know about N is its prime factorization $N = pq$, so let us start by rephrasing what it means to "choose $x \in Z_N^*$ uniformly at random" in terms of "local" factors p and q.

Exercise 10.22. Show that choosing $x \in Z_N^*$ uniformly at random is equivalent to choosing $(a, b) \in S_{pq}$ uniformly at random, for set $S_{pq} := \{(a, b) \mid a \in Z_p^* \text{ and } b \in Z_q^*\}$. (Hint: Use the Chinese Remainder Theorem to demonstrate a bijection between Z_N^* and S_{pq}.)

Thus, assume we equivalently sample (a, b) uniformly at random from S_{pq}. We now have to bound the probability that r is even and $x^{r/2} \notin \{\pm 1\}$ modulo N.

Focus: Largest powers of 2. Recall that $x^r \equiv 1 \mod pq$, $x \equiv a \mod p$, and $x \equiv b \mod q$. Defining r_a and r_b as the orders of a modulo p and b modulo q, respectively, we thus have $x^{r_a} \equiv a^{r_a} \equiv 1 \mod p$, and $x^{r_b} \equiv b^{r_b} \equiv 1 \mod q$.

Exercise 10.23. Prove that r_a and r_b both divide r.

To bound the probability of our bad events (i.e. r is odd or $x^{r/2} \in \{\pm 1\}$ modulo N), it turns out the right place to look is the exact *power of* 2 which appears in each of the prime decompositions of r_a and r_b, respectively. Specifically, we proceed by showing two claims:

(A) Either bad event causes r_a and r_b to have precisely the *same* power of 2 in their prime factorizations.

(B) But since (a, b) is drawn uniformly at random from S_{pq}, the odds that r_a and r_b have this same factor of 2 is precisely $1/2$.

Formally, define t_a and t_b as the exact powers of 2 which divide r_a and r_b, respectively (e.g. $r_a = 2^{t_a} p_2^{k_2} p_3^{k_3} \cdots$ is the prime factorization of r_a, with $p_2, p_3, \ldots$ being distinct primes larger than 2). Let us first prove point (A). Consider the first bad event, r is odd. By Exercise 10.23, r_a and r_b divide r, and hence are both odd. Thus, $t_a = t_b = 0$. The second bad event is when r is even but $x^{r/2} \equiv -1 \mod N$. We know from Exercise 10.23 that r_a and r_b both divide r. Thus, t_a and t_b are *at most* the power of 2 in the prime decomposition of r. We show matching lower bounds. Since $x^{r/2} \equiv -1 \mod N$, we have $x^{r/2} \equiv -1 \mod p$ and $x^{r/2} \equiv -1 \mod q$. But by definition $x^{r_a} \equiv a^{r_a} \equiv 1 \mod p$ and $x^{r_b} \equiv b^{r_b} \equiv 1 \mod q$, so r_a and r_b cannot divide $r/2$. (Otherwise, for example, $x^{r/2} \equiv 1 \mod p$.) Since r_a and r_b divide r but not $r/2$, we obtain our desired lower bound — t_a and t_b must be *at least* the power of 2 in the prime decomposition of r. This concludes the proof of point (A). Point (B) now follows from Lemma 10.25, as the following exercise shows.

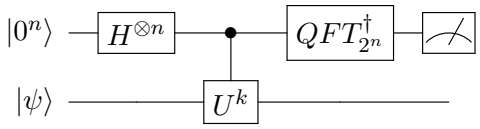


Figure 10.1: A high-level schematic of the QPE algorithm.

Exercise 10.24. Confirm that Point (B) follows from Lemma 10.25. □

Lemma 10.25. *Fix an odd prime p, and choose $a \in Z_p^*$ uniformly at random. Let r_a be the order of a. Then, with probability $1/2$, the largest power of 2 which divides $p-1$ also divides r_a.*

Proof. Let t be the largest power of 2 which divides $p-1$, i.e. 2^t divides $p-1$.

Exercise 10.26. Why can we assume $t > 0$?

The proof strategy exploits Exercise 10.17, which recall states that choosing $a \in Z_p^*$ uniformly at random is equivalent to choosing $k \in \{1, \ldots, p-1\}$ uniformly at random (where $g^k \equiv a \mod p$ for fixed generator g of Z_p). In particular, since Z_p^* has even size, this means that k is odd with probability $1/2$. The key point is that, as we show next, k is odd if and only if 2^t divides r_a, yielding the desired claim regarding r_a.

- (Case 1: k odd) We claim 2^t divides r_a. Since

$$g^{kr_a} \equiv a^{r_a} \equiv 1 \equiv g^{p-1} \mod p,$$

  where the last equivalence follows from FLT, we conclude $p-1$ divides kr_a. But $p-1$ is even and k is odd, implying the factor 2^t which divides $p-1$ does not divide k, but rather r_a, as claimed.

- (Case 2: k even) We claim 2^t does not divide r_a. To show this, note

$$g^{kr_a} \equiv a^{r_a} \equiv 1 \equiv g^{p-1} \equiv g^{k\frac{p-1}{2}} \mod p,$$

  where the third equivalence uses FLT, and the last that k is even. We conclude that r_a divides $(p-1)/2$. But $p-1$ has factor 2^t, meaning $(p-1)/2$ has factor 2^{t-1}, and thus the largest power of 2 appearing in the prime factorization of r_a is also at most 2^{t-1}. Thus, 2^t does not divide r_a, as claimed.

□

10.2.2 Sampling via QPE

Having reduced FACTOR to ORDER, we are now tasked with solving the latter: Given $x, N \in \mathbb{Z}^+$ such that x is co-prime to N, what is the order r of x? It turns out previous lectures have already given us all the toys needed to play this game. Indeed, we now detail the only quantum component of Shor's algorithm, consisting of an application of QPE (a high-level view of QPE is given in Figure 10.1 to jog your memory). This use of QPE requires overcoming three challenges: (1) Which unitary U do we apply QPE to? (2) Where do we find an eigenvector $|\psi\rangle$ of U to plug into QPE? (3) What precision do we need to run QPE to, and can we efficiently compute the corresponding controlled-U^k operation for this precision?

Challenge 1: The choice of U. Recalling that (x, N) is the input to ORDER, we shall run QPE on unitary U_x defined via action

$$U_x|y\rangle = |xy \mod N\rangle. \tag{10.7}$$

Exercise 10.27. Let V be a $d \times d$ complex matrix. Prove that V is unitary if and only if V maps any given orthonormal basis $B \subseteq \mathbb{C}^d$ to an orthonormal basis $B' \subseteq \mathbb{C}^d$. Can you give an outer product representation of V?

Exercise 10.28. Using the previous exercise, prove that U_x is unitary. (Hint: Use the fact that for any x co-prime to N, there exists an inverse $x^{-1} \in Z_N^*$ such that $xx^{-1} \equiv 1 \mod N$.)

The claim now is that certain eigenvalues of U_x encode the order r of x. Let us "design" the corresponding eigenvectors from scratch to build intuition. First, what do we know, and why did we choose U_x the way we did? Well, we know $x^r \equiv 1 \mod N$, and the operation $y \mapsto xy \mod N$ induced by U allows us to iteratively produce the elements of the infinite sequence $(1, x, x^2, x^3, \ldots, x^{r-1}, 1, x, \ldots)$. It follows that a first guess at an eigenvector of U_x is

$$|\eta\rangle := \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |x^j \mod N\rangle. \tag{10.8}$$

Exercise 10.29. Show that $U_x|\eta\rangle = |\eta\rangle$, i.e. $|\eta\rangle$ has eigenvalue 1.

Unfortunately, an eigenvalue of 1 is not very exciting. However, we are only a tweak away from what we want. Recall the other setting in which we have a "cyclic" or "wraparound" property: The complex unit circle. In particular, for principal rth root of unitary $\omega_r := \exp(2\pi i/r)$, we have $\omega_r^r = 1$. Thus, let us try injecting rth roots of unity as relative phases into $|\eta\rangle$ to obtain:

$$|\phi\rangle := \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{2\pi ij/r} |x^j \mod N\rangle. \tag{10.9}$$

Exercise 10.30. Show that $U_x|\phi\rangle = e^{-2\pi i/r}|\phi\rangle$, i.e. $|\phi\rangle$ has eigenvalue $e^{-2\pi i/r}$.

We conclude that running QPE with U_x on $|\phi\rangle$ would yield an approximation to $1/r$, from which we could in principle recover r, as desired. There's only one problem — *how do we prepare* $|\phi\rangle$? Indeed, to prepare $|\phi\rangle$, it would seem we need to know r, which is precisely what we had originally set out to find!

Exercise 10.31. In Lecture 9, we assumed in the definition of QPE that the input phase $\theta \in [0, 1)$, whereas above we have $\theta = -i/r$. Show how a simple modification to the QPE algorithm allows us to nevertheless extract i/r.

Challenge 2: How to prepare $|\phi\rangle$. Unfortunately, it is not known how to construct $|\phi\rangle$ directly. There is, however, a slick way around this problem. Let us extend $|\phi\rangle$ to an orthonormal basis $B := \{|\phi_k\rangle\}$ for the Hilbert space $\mathcal{H}$ which U_x acts on, where we define $|\phi_1\rangle := |\phi\rangle$. Then, any unit vector $|\psi\rangle \in \mathcal{H}$ may be written $|\psi\rangle = \sum_k \alpha_k |\phi_k\rangle$ with $\sum_k |\alpha_k|^2 = 1$. Since QPE is unitary (omitting the final measurement), by linearity we have

$$|0^n\rangle|\psi\rangle = \sum_k \alpha_k |0^n\rangle|\phi_k\rangle \xmapsto{QPE} \sum_k \alpha_k |\theta_k\rangle|\phi_k\rangle, \tag{10.10}$$

where θ_k is an approximation to the phase corresponding to $|\phi_k\rangle$.

Exercise 10.32. Suppose we measure the first register of the right side of Equation (10.10). With what probability do we obtain our desired outcome θ_1 corresponding to $|\phi_1\rangle$?

The takeaway is that we need not prepare $|\phi_1\rangle$ *exactly*; rather, it may be easier to prepare some $|\psi\rangle$ which has non-zero amplitude α_1 on $|\phi_1\rangle$. It turns out this is not only easy, but *trivial*. Indeed, there is a clever choice of the first r basis vectors $|\phi_k\rangle$ in B which satisfies

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\phi_k\rangle = |1 \mod N\rangle. \tag{10.11}$$

(Note that $|1 \mod N\rangle$ is indeed trivial to prepare.) So what are $|\phi_2\rangle, \ldots, |\phi_r\rangle$? In particular, how do we systematically construct states orthonormal to $|\phi_1\rangle$? Observe that $|\phi_1\rangle$ is essentially a column from a Vandermonde matrix, obtained by raising the principal rth root of unity ω_r to increasing powers. Thus, we know precisely how to generate $r-1$ more vectors orthonormal to $|\phi_1\rangle$ — simply consider the Vandermonde matrix whose kth (for $k \in \{0, \ldots, r-1\}$) row exponentiates ω_r^k to increasing powers (up to $r-1$, inclusive). Formally, define

$$|\phi_k\rangle := \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{2\pi i j k/r} |x^j \mod N\rangle. \tag{10.12}$$

Exercise 10.33. Prove that vectors $\{|\phi_k\rangle\}_{k=1}^r$ are orthonormal.

Exercise 10.34. Prove that $U|\phi_k\rangle = e^{-\frac{2\pi i k}{r}} |\phi_k\rangle$.

Exercise 10.35. Show that defining $|\phi_k\rangle$ as in Equation (10.12) yields Equation (10.11).

Plugging the exercise above into Equations (10.10) and (10.11), we have that if we run QPE on $|1 \mod N\rangle$ and measure the first register, with probability $1/r$ we get (an approximation to) phase k/r for some $k \in [r-1]$. This is almost what we want, except for the pesky k variable in the numerator. So we must settle for something less — the ability to sample uniformly at random from set $S := \{1/r, 2/r, \ldots, (r-1)/r\}$. Is this enough to extract r? Incredibly, the answer is yes, and will be covered in Section 10.2.3.

Exercise 10.36. Consider the following naive algorithm for extracting r. Run QPE repeatedly t times to collect t samples from S, for t some polynomial in the input size, $\log N$. Let t^* denote the smallest sample drawn. Return $1/t^*$ as the guess for r. Intuitively, how large might t have to be for this algorithm to extract r with (say) constant probability?

Challenge 3: Precision and the controlled-U^k operation. Before discussing how sampling from S allows us to extract r, we must address a key issue we have thus far swept under the rug: *Can we even efficiently run the QPE algorithm to the precision required for sampling from S, i.e. to represent k/r for $k \in [r-1]$?* To analyze this, recall that the input size to FACTOR is the number of bits needed to encode N, i.e. $n := \lceil \log N \rceil + 1$ bits.

Exercise 10.37. Consider arbitrary $k, r \in \mathbb{Z}^+$ for $k < r$, which can both be expressed exactly using n bits. How many bits are required to express the ratio k/r as a real number, i.e. as a sequence of bits $0.b_1b_2b_3 \dots b_m$? (Hint: Does k/r necessarily have a finite binary expansion, even if k and r have finite expansions?)

As the exercise above suggests, even if k and r have finite binary expansions, it is not necessarily true that k/r has a finite expansion. Thus, *a priori* it is not clear what precision we should require from QPE. Naively, one might "guess" that since kr requires at most $2n+1$ bits to represent, perhaps k/r can be sufficiently well-approximated also using $2n+1$ bits (at least close enough for us to recover k and r as integers). Indeed, it turns out that $2n+1$ bits of QPE precision will suffice for the continued fractions algorithm of Section 10.2.3 to extract k and r.

To obtain $2n+1$ bits of precision in QPE with probability at least $1-\epsilon$ for fixed $\epsilon > 0$, however, we require two things: Expanding the first register of Figure 10.1 to $n' = 2n+1+\lceil \log(2+\frac{1}{2\epsilon}) \rceil$ qubits (which is just a constant factor overhead), and the ability to raise U to power $2^{n'} \in O(2^n)$ in time $\text{poly}(n)$ (this smells like trouble). Stated differently, the latter requires us to compute $x^k y \mod N$ for $k \in O(2^n)$ in time $\text{poly}(n)$. As luck would have it, there is a well-known classical algorithm which accomplishes essentially this, known as *square-and-multiply*.

The square-and-multiply algorithm. This algorithm allows us to compute $a^b \mod M$ using $\text{polylog}(b)$ operations, which is exponentially faster than the brute force method of multiplying a by itself b times.

Basic idea. The basic idea is most easily seen when b is a power of 2, e.g. $b = 2^n$. In this case, note that (trivially) $a^b = (a^2)^{b/2}$. In other words, we can cut the exponent *in half* simply by performing a single squaring of a. So let us repeat this tradeoff, each time squaring the result of the previous square and cutting the remaining exponent in half. How many times can we repeat this before it terminates? Precisely $\log b = n$ times, hence yielding an exponential speedup over brute force, which would have required b operations.

General case. When b is not a power of 2, the principle is similar, but the recursive step requires two cases instead of one:

$$a^b = \begin{cases} (a^2)^{\frac{b}{2}} & \text{if } b \text{ is even} \\ a(a^2)^{\frac{b-1}{2}} & \text{if } b \text{ is odd.} \end{cases} \tag{10.13}$$

The square-and-multiply algorithm repeatedly applies this recursive step until $b = 0$, at which point it terminates. Note that since we are working mod M, in order to prevent the expression being computed from blowing up exponentially quickly, we may apply the mod operation after each application of the recursive rule.

Exercise 10.38. Asymptotically, how many operations does the algorithm require in the general case?

10.2.3 Postprocessing via continued fractions

In Section 10.2.2, we used QPE to approximately sample from set $S := \{1/r, 2/r, \ldots, (r-1)/r\}$. More specifically, defining $n := \lceil \log N \rceil + 1$, each time we ran the QPE subroutine, we obtained (with probability at least $1 - \epsilon$) the $2n + 1$ most significant bits of the binary expansion of some random element k/r of S. Given access to such a sample x, it thus remains to extract a guess for r. To do so, we run the *continued fractions* algorithm, for which we first overcome the following minor obstacle.

A minor obstacle. If we are to have any hope of extracting r given k/r, we must have k and r co-prime (in other words, k/r is already in lowest terms). *A priori*, there is no way of guaranteeing this, since k is drawn uniformly at random from $\{0, \ldots, r-1\}$. However, the uniform sampling is simultaneously the key to the solution. Namely, given positive integer r, the number of positive integers less than r which are *co-prime* to r has been studied since 1763, and goes by the name of the Euler totient function $\phi(r)$. The totient function scales as $\phi(r) \in \Omega(r/\log\log r)$ for $r > 2$. Thus, a randomly chosen k in $\{0, \ldots, r-1\}$ is co-prime to r with probability scaling as $1/\log\log r \geq 1/\log\log N$. It follows that repeating the QPE sampling step $O(\log\log N)$ times (i.e. logarithmic in the encoding size of N) suffices to obtain k/r with k co-prime to r with high probability. Assuming we have such a k/r, we can now apply continued fractions.

The continued fractions algorithm. In a nutshell, the continued fractions expansion is just another representation for a given rational number a/b for $a, b \in \mathbb{Z}^+$. Specifically, given a/b, the algorithm outputs a finite sequence of integers $C := (c_0, c_2, \ldots, c_m)$, such that

$$\frac{a}{b} = c_0 + \cfrac{1}{c_1 + \cfrac{1}{c_2 + \cfrac{1}{\cdots + \cfrac{1}{c_m}}}}. \tag{10.14}$$

Why should this representation of a/b be useful? First, instead of viewing C as representing a *single* rational number, note we may view it as encoding a *list* of m rational numbers, each defined via the continued fraction subsequence (formally, *kth convergent*) $C_k := (c_0, \ldots, c_k)$. Recall now that what QPE gave us was not a random sample k/r, but rather the first $2n + 1$ bits of k/r, denoted x. Thus, x itself is rational, and is "close" to k/r. Amazingly, the following theorem thus tells us that the continued fractions expansion of x will yield a sequence of m convergents, one of which is precisely a/b.

Theorem 10.39. *Let $k, r \in \mathbb{Z}^+$ and $x \in \mathbb{Q}^+$ satisfy*

$$\left| \frac{k}{r} - x \right| \leq \frac{1}{2r^2}. \tag{10.15}$$

Let $C = (c_0, \ldots, c_M)$ be the continued fractions expansion of x. Then, there exists a $k \in \{0, \ldots, M\}$ such that the kth convergent $(c_0, \ldots, c_k)$ represents k/r exactly.

Exercise 10.40. Show that since x is the $2n + 1$ most significant bits of k/r, Equation (10.15) indeed holds, and thus we can apply Theorem 10.39.

This *almost* gives us what we want — instead of an approximation x to k/r, we now actually have k/r exactly. There are three questions to be answered: (1) How large is M in Theorem 10.39?

(2) How do we compute the continued fractions expansion of x? (3) Given the convergent of x which yields k/r, how do we extract r (recall we don't know k)? For (1), it turns out since k and r are n-bit integers, one can choose $M \in O(n)$. The answers to the other two questions we give below.

Computing the continued fractions expansion. The algorithm essentially "implements" Equation (10.14) in a "greedy" fashion. Let us demonstrate with an explicit example, the expansion of 23/9:

$$\frac{23}{9} \xrightarrow{\text{split}} 2 + \frac{5}{9} \xrightarrow{\text{invert}} 2 + \frac{1}{\frac{9}{5}} \xrightarrow{\text{split}} 2 + \frac{1}{1 + \frac{4}{5}} \xrightarrow{\text{invert}} 2 + \frac{1}{1 + \frac{1}{\frac{5}{4}}} \xrightarrow{\text{split}} 2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{4}}}.$$

As demonstrated above, the algorithm repeatly applies the *split* step (extract the integer component) and *invert* step (flip the fractional component) until the final fraction is of form $1/z$ for some $z \in \mathbb{Z}$.

Exercise 10.41. Give the continued fractions expansion for 31/11.

Exercise 10.42. Prove that the continued fractions expansion for any rational number is indeed finite.

Recovering r by "inverting" the continued fractions expansion. Let $C = (c_0, c_1, \ldots, c_m)$ be the continued fractions expansion of the $2n+1$-bit estimate x for k/r given by QPE. Since we assumed k and r are co-prime, we are now in a position to extract r via the following theorem. Note we do not know *a priori* which convergent of C correctly encodes k/r; but since $m \in O(n)$, we can simply try all convergents in polynomial time.

Theorem 10.43. *Given any continued fractions expansion (or more generally, some kth convergent) $C_k := (c_0, c_1, \ldots, c_k)$ where $c_i \in \mathbb{Z}^+$ for all $i \in 0, \ldots, k$, define a_i and b_i recursively as follows:*

$$\begin{array}{llll} a_0 = c_0 & a_1 = 1 + c_0 c_1 & a_i = c_i a_{i-1} + a_{i-2} & \text{for } i \in \{2, \ldots, k\} \\ b_0 = 1 & b_1 = c_1 & b_i = c_i b_{i-1} + b_{i-2} & \text{for } i \in \{2, \ldots, k\}. \end{array}$$

Then, the expansion C_k is equivalent to rational number a_k/b_k, where a_k and b_k are co-prime.

Combining Theorems 10.39 and 10.43, we hence have that trying all m possible convergents in the expansion C for x (which recall is a $2n+1$-bit estimate of k/r) will eventually allow us to recover both k and r.

Exercise 10.44. Apply Theorem 10.43 to $C = (1, 2, 3, 4)$. What are a_3 and b_3? Confirm that they are co-prime.

10.3 Application: Breaking RSA

Finally, let us briefly discuss how the quantum factoring algorithm allows us to efficiently break RSA.

The RSA cryptosystem. RSA, named after its founders Rivest-Shamir-Adleman, is a public key cryptosystem still widely used today. Roughly, in a *public key* cryptosystem, Alice wishes to allow the public to send her private messages. For this, she creates a pair of keys: One public key P, one private key S. The public key P is distributed openly to the world; anyone can use P to encrypt a given message M, obtaining ciphertext $E(M)$ for transmission to Alice. Upon receipt of any such $E(M)$, Alice uses her private key S, which is only known to her, to decrypt $E(M)$ and recover M.

What we thus need to specify for RSA are the public and private keys P and S, respectively, as well as the encryption and decryption algorithms. These are given as follows.

- **How Alice creates the keys:**
  1. Alice chooses two large prime integers p and q, and computes $N = pq$.
  2. She randomly chooses small odd $e \in \mathbb{Z}$ co-prime to $\phi(N) := (p-1)(q-1)$.
  3. She computes d satisfying $ed \equiv 1 \mod \phi(N)$.
  4. She publicly releases $P = (e, N)$, and privately stores $S = (d, N)$.
- **How the public encrypts messages given $P = (e, N)$.** Suppose Bob wishes to encrypt string $M \in \{0,1\}^{\lfloor \log N \rfloor}$. Viewing M as the binary encoding of an integer, he computes ciphertext
$$E(M) := M^e \mod N. \tag{10.16}$$
Note this can be done efficiently even for large e via the square-and-multiply algorithm.

  Exercise 10.45. Why is the length of M assumed to be bounded by $O(\log N)$ above?
- **How Alice decrypts ciphertext $E(M)$.** Given ciphertext $E(M)$, Alice recovers M by computing
$$M = (E(M))^d \mod N. \tag{10.17}$$
The proof of this equality is not difficult, but is omitted to avoid sidetracking the discussion.
- **How factoring allows one to completely break RSA.** Suppose adversary Eve can factor N efficiently into primes p and q. Then, since e is public knowledge, she can recompute d such that $ed \equiv 1 \mod \phi(N)$ (Step 3 of Alice's creation protocol), giving her Alice's private key $S = (d, N)$. Thus, not only can Eve decrypt any incoming ciphertext to Alice, but she in fact has complete knowledge of Alice's secret key.

In sum, what makes RSA classically "hard" to break is that given just $N = pq$, it is not clear how to compute $\phi(N) = (p-1)(q-1)$. But given the factors p and q, recovering $\phi(N)$, and hence the private key d, is easy.

11 Grover search and approximate counting

"The way to find a needle in a haystack is to sit down."
— Beryl Markham.

"After that, work and hope. But never hope more than you work."
— Also Beryl Markham.

Shor's factoring algorithm and Grover's search algorithm are like soccer teammates on different ends of the field — the former, like a striker, is highly specialized to do one thing efficiently (i.e. score), while the latter, akin to a sweeper, has the broad job of neutralizing all threats which slip past the rest of the team. Indeed, Shor's algorithm gives an *exponential* speedup for the *specific* problem of factoring, whereas Grover search can be thrown at just about *anything* to yield a *quadratic* speedup.

Discovered by Lov Grover in 1996, Grover search is more specifically a quantum algorithm for solving the following general problem: Given query access to an oracle containing N items, one of which is "marked", find the marked item. For example, the "oracle" could be implemented by a 3-SAT formula ϕ, indexed by n-bit assignments x, and the aim is to find a satisfying assignment x (which would be considered "marked"). Grover's algorithm solves this problem with high probability using $O(\sqrt{N})$ queries to the database (which turns out to be optimal for a quantum algorithm), whereas a classical algorithm would require $\Omega(N)$ queries in the worst case. Thus, Grover search solves 3-SAT in $O(\sqrt{2^n})$ time.

We begin in Section 11.1 by defining the unstructured search problem. Section 11.2 then gives Grover's algorithm, and Section 11.3 discusses a recent approach for bootstrapping Grover's algorithm for approximate counting.

11.1 The unstructured search problem

We begin by formalizing the unstructured search problem.

Definition 11.1 (Unstructured search (SEARCH))**.**

- *Input: Query access to an oracle U_f for $f : \{0,1\}^n \mapsto \{0,1\}$ an unknown Boolean function, meaning the ability to compute (at unit cost) map*

$$|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle \tag{11.1}$$

  for any $x, y \in \{0,1\}^n$, and $\oplus$ the bit-wise XOR operation.

- *Output: An index $x \in \{0,1\}^n$ such that $f(x) = 1$, if one exists.*

Note that no assumptions about U_f are made, other than the requirement that we have superposition query access to the input/output behavior of f.

Exercise 11.2. SEARCH is often alternatively formulated as follows: The oracle U_f is replaced with an unknown string $z \in \{0,1\}^{2^n}$, such that each query to U_f is equivalent to accessing a single bit of z. The output is to compute the OR function on string z, i.e. $\mathrm{OR}(z) = \bigvee_{i=1}^{2^n} z_i$. Explain why this formulation of SEARCH is equivalent to Definition 11.1.

Application to 3-SAT and the Exponential-Time Hypothesis. As alluded to in the introduction, we may view a 3-SAT formula ϕ as a function $f : \{0,1\}^n \mapsto \{0,1\}$, i.e. which maps n-bit assignment x to $\phi(x)$. Thus, finding a satisfying assignment to ϕ is a special case of SEARCH with $f = \phi$. As we shall see shortly, SEARCH can be solved in $O(\sqrt{2^n})$ time quantumly, and this turns out to be optimal. Thus, quantumly one can apply Grover search as a black box to 3-SAT to find a satisfying assignment (if one exists) in $O(\sqrt{2^n})$ time. Does this saying anything about the complexity of 3-SAT itself?

Maybe, maybe not. It is generally believed that 3-SAT cannot be solved in subexponential time, at least on a classical computer. This is the premise of the *Exponential-Time Hypothesis* of Impagliazzo and Paturi from 1999, stated as follows.

Claim 11.3 (Exponential-Time Hypothesis (ETH))**.** *There exists a constant $\epsilon > 0$ such that 3-SAT requires time $\Omega(2^{\epsilon n})$ on a deterministic Turing machine.*

While the runtime of Grover's algorithm does not contradict ETH, if one *does* believe ETH, then it is perhaps not too surprising that $O(\sqrt{2^n})$ turns out to be the optimal worst-case runtime for a black-box quantum search algorithm.

Is ETH true? Again, this is not clear, but assuming the truth of the ETH has led to matching runtime *lower bounds* in an area known as *fine-grained complexity*. Roughly, the latter aims to pin down the precise complexity of problems which are *known to be in* P (e.g. is the optimal worst-case runtime, say, $O(n^3)$ or $O(n^2)$?). For example, in the Orthogonal Vectors Problem (OV), given sets of vectors $A, B \subseteq \{0,1\}^d$ with $|A| = |B| = n$, one is asked whether there exist $|v\rangle \in A$ and $|w\rangle \in B$ such that $\langle v|w\rangle = 0$?

Exercise 11.4. Show that OV can be solved in $O(n^2 d)$ time.

It turns out that, assuming a stronger variant of ETH known as the Strong Exponential Time Hypothesis, the naive polynomial runtime of Exercise 11.4 is the best possible. The interested reader is referred to the accessible survey of Bringmann [Bri19] for details.

11.2 Grover's algorithm

Define $N := 2^n$. We now show how to solve SEARCH in $O(\sqrt{N})$ time on a quantum computer. For clarity, recall that typically when one discusses problems with black-box access to an oracle (as in SEARCH), the relevant cost model is *query complexity* (i.e. each query has unit cost, and this is the *only* cost we care about). This is also the cost model we adopt here. We begin by revisiting our old friend, the phase kickback trick. Viewing this trick geometrically, in particular, will kickstart the development of the remainder of Grover's algorithm.

Phase kickback. The starting point is the phase kickback trick used in conjunction with oracle U_f. Recall this means using mapping $|x\rangle|-\rangle \mapsto (-1)^{f(x)}|x\rangle|-\rangle$, or for brevity,

$$|x\rangle \mapsto (-1)^{f(x)}|x\rangle. \tag{11.2}$$

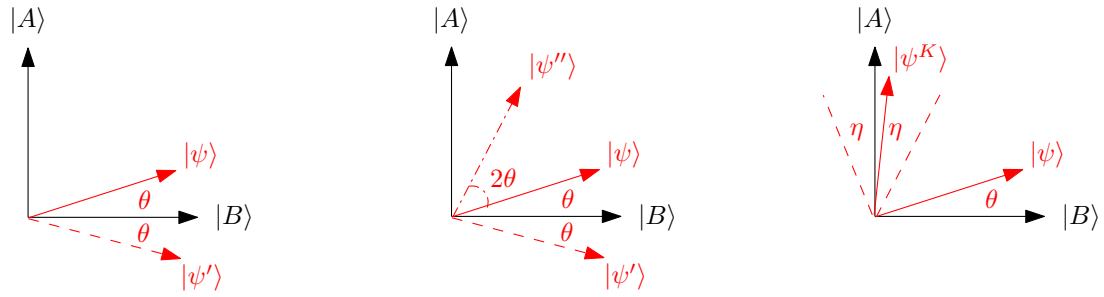


Figure 11.1: (Left) A reflection of $|\psi\rangle$ about $|B\rangle$ to $|\psi'\rangle$. (Middle) A reflection of $|\psi'\rangle$ about $|\psi\rangle$. For clarity, the angle 2θ is between $|\psi''\rangle$ and $|\psi'\rangle$. (Right) The final state $|\psi^K\rangle = (R_\psi R_B)^K|\psi\rangle$ after running the Grover iterate K times.

It will be remarkably helpful to visualize phase kickback *geometrically* in the case of SEARCH. This is done, roughly, by considering superpositions over marked and unmarked items (for clarity, a "marked" ("unmarked") item x satisfies $f(x) = 1$ ($f(x) = 0$)). For this, let $A, B \subseteq \{0,1\}^n$ be the sets of marked and unmarked items, respectively, and define

$$|A\rangle := \frac{1}{\sqrt{|A|}} \sum_{x \in A} |x\rangle \quad \text{and} \quad |B\rangle := \frac{1}{\sqrt{|B|}} \sum_{x \in B} |x\rangle. \tag{11.3}$$

Thus, $|A\rangle$ ($|B\rangle$) is an equal superposition over all marked (unmarked) items. The geometric interpretation follows from the next exercise.

Exercise 11.5. Show that $U_f|A\rangle = -|A\rangle$ and $U_f|B\rangle = |B\rangle$.

In words, restricted to the 2D space defined by $\mathrm{Span}(|A\rangle, |B\rangle)$, U_f acts as a reflection about $|B\rangle$, as depicted in Figure 11.5. And this is no coincidence — digging more deeply into this view will lead us directly to Grover's algorithm (though, for clarity, this geometric view was only discovered *after* Grover's original work). To see this, let us remind ourselves of the second tool we have at our disposal — preparing some initial start state $|\psi\rangle$. Ideally, this state $|\psi\rangle$ should also lie in the span of $|A\rangle$ and $|B\rangle$, so that it "fits" into the 2D picture of Figure 11.5.

Exercise 11.6. Take a moment to guess what might be a "reasonable" choice of $|\psi\rangle$, given our current state of knowledge. What do we know about the location of the marked items? What choice of $|\psi\rangle$ might lie in the span of $|A\rangle$ and $|B\rangle$?

The start state, $|\psi\rangle$. Since *a priori* we have no reason to believe any particular item x is marked, a naive choice of start state is

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle = \sqrt{\frac{|A|}{N}}|A\rangle + \sqrt{1 - \frac{|A|}{N}}|B\rangle. \tag{11.4}$$

Exercise 11.7. Prove the second equality above.

In other words, $|\psi\rangle \in \mathrm{Span}(|A\rangle, |B\rangle)$, as desired, and we may depict it as in Figure 11.5 (Middle). The angle θ therein is given in the following exercise.

Exercise 11.8. Show that for start state $|\psi\rangle$ in Equation (11.4), $\cos\theta = \sqrt{1 - \frac{|A|}{N}}$.

The goal. To guide the rest of the algorithm, we now pause and step back to restate our goal in the 2D picture of Figure 11.5. Given the ability to prepare $|\psi\rangle$, we wish to rotate $|\psi\rangle$ counter-clockwise up to to $|A\rangle$, and subsequently measure in the standard basis, thus obtaining some marked item $x \in A$. Since this is just a rotation map, it is certainly possible, in that there exists a $2^n \times 2^n$ unitary matrix performing this rotation. The question is: *Can this mapping be computed using* poly(n) *queries to* U_f *(and ideally,* poly(n) *auxiliary gates)*?

Two reflections make a rotation. As the saying goes, beggars can't be choosers, and indeed to attain our goal, we must make do with what few tools the generality of SEARCH affords us: We can reflect about $|B\rangle$, and we can prepare $|\psi\rangle$. The key observation is that since we can efficiently prepare $|\psi\rangle$, i.e. $|\psi\rangle = H^{\otimes n}|0^n\rangle$, we can also *reflect* about $|\psi\rangle$.

Exercise 11.9. In Lecture 9, we saw that for $|\psi\rangle$, a reflection about $|\psi\rangle$ is achieved by unitary $U_\psi = 2|\psi\rangle\langle\psi| - I$. Show how to implement U_ψ for our choice of $|\psi\rangle$ from Equation (11.4). (Hint: Begin by showing how to reflect about $|0^n\rangle$, i.e. by implementing $U_{0^n} = 2|0^n\rangle\langle 0^n| - I$.))

Geometrically, this means we can map $|\psi'\rangle$ to $|\psi''\rangle$ in Figure 11.5. In other words, by first reflecting about $|B\rangle$, and then about $|\psi\rangle$, we can effect a counter-clockwise rotation of 2θ in Figure 11.5. Magic! Formally, we shall repeatedly apply the pair of reflections (often dubbed the "Grover iterate")

$$(2|\psi\rangle\langle\psi| - I)(2|B\rangle\langle B| - I) =: R_\psi R_B, \tag{11.5}$$

where R_B is effected by querying the oracle U_f, and R_ψ by undoing the preparation procedure for $|\psi\rangle$, reflecting about $|0^n\rangle$, and then re-doing the preparation for $|\psi\rangle$ (as per Exercise 11.9).

The number of iterations required. We can now state Grover's algorithm as follows:

1. Prepare $|\psi\rangle = H^{\otimes n}|0^n\rangle$.
2. Apply the Grover iterate, $R_\psi R_B$, K times.
3. Measure in the standard basis to obtain string $x \in \{0,1\}^n$.

If there are no solutions, i.e. $M = 0$, this procedure always outputs x satisfying $f(x) = 0$, as expected. If $M > 0$, on the other hand, the question is what to set K, the number of loop iterations, to? Note that it suffices for the algorithm to succeed with any fixed constant success probability p, as then independently repeating the algorithm drives down the overall error probability exponentially. To simplify the analysis, set $p = 1/2$. Without loss of generality, we may assume $|A|/N \leq 1/2$ (as otherwise classically choosing uniformly random inputs to U_f yields success probability at least $1/2$). By Exercise 11.8, we hence have $\theta \leq \pi/4$.

Exercise 11.10. Show that for Step 3 of Grover's algorithm to output $x \in \{0,1\}^n$ satisfying $f(x) = 1$ with probability at least $1/2$, it suffices in Figure 11.5 (Right) that $|\psi^K\rangle$ makes angle at most $\eta \leq \pi/4$ with $|A\rangle$. (Hint: Recall your high school formula that the overlap between real vectors $|v\rangle$ and $|w\rangle$ equals $\langle v|w\rangle = \| |v\rangle \|_2 \| |w\rangle \|_2 \cos\eta$ for η the angle between $|v\rangle$ and $|w\rangle$.)

Thus, by Exercise 11.8, we start with $|\psi\rangle$ at $\theta = \arccos(\sqrt{(1 - |A|)/N}) \leq \pi/4$, and we wish to end up at $|\psi^K\rangle$ with $\eta \in [-\pi/4, \pi/4]$.

Exercise 11.11. Given that $|\psi\rangle$ starts at angle $\theta \leq \pi/4$, do we ever need to wrap around the circle (when applying the Grover iterate) in order to land in range $\eta \in [-\pi/4, \pi/4]$?

Exercise 11.12. Using your answer from the previous exercise, show that setting

$$K = \left\lceil \frac{1}{2\theta} \left(\arccos \sqrt{\frac{|A|}{N}} - \frac{\pi}{4} \right) \right\rceil \tag{11.6}$$

suffices to succeed with probability at least $1/2$, assuming $|A| > 0$.

Exercise 11.13. Show that $\sin\theta = \sqrt{|A|/N}$ and that $\theta \geq \sin\theta$ for $|\theta| \leq 1$. Using these facts, show that

$$K \leq \left\lceil \frac{\pi}{8} \sqrt{\frac{N}{|A|}} \right\rceil \in O\left(\sqrt{\frac{N}{|A|}} \right) \tag{11.7}$$

queries to U_f suffice to find a marked item with probability at least $1/2$.

Exercise 11.14. How many auxiliary gates (i.e. gates other than queries to U_f) does Grover's algorithm use?

In closing, given query access to an oracle U_f for which $|A|$ out of $N = |A| + |B|$ items are marked, a marked item can be found quantumly using $O(\sqrt{|A|/N})$ queries. There is a slight catch, however — the exact number of queries required, as given by Equation (11.6), requires knowledge of $|A|$. Luckily, it turns out that not only do quantum algorithms allow us to check if $|A| > 0$, but additionally to estimate $|A|$ itself to within multiplicative error. This is known as *quantum approximate counting*, covered next.

11.3 Approximate counting

In Section 11.1, we defined the input to SEARCH as an oracle U_f, and the output was to find a marked item x, i.e. satisfying $f(x) = 1$. This can be generalized to the much more difficult question: Can we *count* the number of marked items?

11.3.1 A brief history of counting

Recall that in SEARCH we place no requirements on the complexity of *implementing* U_f, making SEARCH extremely general. However, here on Earth, one typically requires U_f to have an efficient implementation. For example, if $f : \{0,1\}^n \mapsto \{0,1\}$ is efficiently implementable by a deterministic Turing machine M_f, then assuming in SEARCH we are also given a description of M_f (as opposed to just query access), SEARCH is NP-complete. (For example, as done in Section 11.1, U_f could evaluate a 3-SAT formula.) If we now ask the more general question *"how many $x \in \{0,1\}^n$ satisfy $f(x) = 1$?"*, we obtain precisely the complexity class #P, which is believed much harder than NP. For example, while NP is (by definition) the first level of the Polynomial-Time Hierarchy (PH), Toda's theorem tells us $\mathrm{P}^{\#\mathrm{P}}$ contains *all* of PH. Thus, the ability to *count* solutions allows us to solve NP and a whole lot more.

Given the importance of #P to complexity theory, it is worth taking a moment to understand what quantum approximate counting shall buy us, in comparison to what is possible classically (again, assuming f is efficiently implementable by a Turing machine). Let M denote the number

of $x \in \{0,1\}^n$ satisfying $f(x) = 1$. The classic result for approximate counting is Stockmeyer's algorithm, which shows how to approximate M within a multiplicative factor of 2 in randomized polynomial time, *assuming* one has access to an NP oracle. (This factor of 2 can then easily be boosted to $1 + (1/p(|x|))$ for any desired fixed polynomial p.) In contrast, in this section we shall make a tradeoff — by adding quantum computation to the picture, we no longer need an NP oracle, but now the runtime is worst-case exponential: $O(\sqrt{M/N})$ to be precise (assuming we want a constant probability of success). And this tradeoff in some sense is necessary — in the worst case, approximating the Permanent of a matrix (a classic #P-complete problem dating back to Valiant's original 1979 paper on #P) within a constant multiplicative error is *still* #P-*hard* [AA11]. And it is nowadays generally believed quantum computers cannot efficiently solve NP-hard problems, never mind #P-hard problems.

11.3.2 Quantum approximate counting via Grover search

Returning to the setting where U_f is a black-box about which we make no assumptions, there are nowadays multiple approaches for quantumly approximately counting $M := |\{x \in \{0,1\} \mid f(x) = 1\}|$. A classic approach is to run QPE on the Grover iterate (a theme which reappears in more general quantum walk frameworks), as it turns out the *eigenvalues* of the iterate encode M. However, here we shall review a conceptually simpler, more recent approach due to Aaronson and Rall [AR20], which does away with the QPE machinery and whittles the solution down to requiring just a single tool — Grover search itself.

Intuition

The basic idea. Let $p = M/N$, i.e. the fraction of satisfying assignments. Naively, there is a simple classical algorithm for estimating p — simply pick x uniformly at random, and evaluate $f(x)$. By definition, this succeeds with probability p, and so $1/p$ trials are expected before a satisfying assignment is found.

Exercise 11.15. Take a moment to Google for "geometric distribution". Show how to model the sampling experiment above as a geometric distribution. What is the expected value and variance for this distribution? What is the probability that the number of trials needed to see a success deviates significantly from $1/p$? (Hint: Use Chebyshev's inequality, which unlike Markov's inequality, takes the variance into account.)

Of course, in the worst case, $1/p \in O(N)$, and by now we are spoiled with getting faster $O(\sqrt{N})$ runtimes via Grover search. Thus, roughly, the quantum algorithm we discuss will carefully mimic (a refinement of) the idea above in conjunction with Grover search. In the remainder of this section, we state the algorithm, and sketch its proof of correctness. The interested reader is referred to [AR20] for full details.

Quantizing the basic idea. Recall from Exercise 11.13 that in Grover search, the angle made by start state $|\psi\rangle$ with $|B\rangle$ is $\theta = \arcsin(\sqrt{|A|/N})$, or in the terminology of this section, $\theta = \arcsin(\sqrt{M/N})$. One can generalize the analysis of Section 11.2 to show that by making $O(r)$ queries to U_f, Grover search finds a marked item with probability $p = \sin^2(r\theta)$. The smaller M is, the smaller θ is, and hence the larger r needs to be to make p large, as expected.

Exercise 11.16. More accurately, denoting the Grover iterate as G, we have

$$G^{(r-1)/2} = \frac{\sin(r\theta)}{\sqrt{M}} \sum_{x \in A} |x\rangle + \frac{\cos(r\theta)}{\sqrt{N-M}} \sum_{x \in B} |x\rangle. \tag{11.8}$$

Confirm that the probability of extracting a marked item after $O(r)$ uses of G is indeed p.

The beauty of [AR20] is now that we can forget about the word "quantum", and simply think of p as a probability arising from some abstract sampling experiment E with parameter r (we henceforth write $E(r)$ where appropriate). The question is: *Given the ability to choose r in this experiment $E(r)$, how many runs of E do we need to estimate p (thus allowing us to extract θ, which encodes the number of solutions M)*?

The high-level outline for achieving this with a quadratic speedup consists of two steps:

1. (Rough estimate) Repeat $E(r)$ using exponentially increasing values of r until "sufficiently many" marked items are found. This gives a rough estimate of $K_- \leq \theta \leq K_+$.
2. (Finetuning the estimate) Iteratively cut down the interval $[K_-, K_+]$ to zoom in on θ.

The second step, in particular, will require a careful choice of r each time $E(r)$ is run to avoid cutting the candidate interval $[K_-, K_+]$ too much, which in particular is a danger if $\theta \approx K_-$ or $\theta \approx K_+$. This shall rely on the Steady Hands Lemma 11.20, to be stated shortly.

Algorithm statement and analysis

The main theorem of [AR20] is the following.

Theorem 11.17. *Fix any desired $\epsilon, \delta > 0$. Given oracle access to U_f (as in* SEARCH*), there exists a quantum algorithm which:*

1. *outputs $\widetilde{M}$ satisfying $(1-\epsilon)M < \widetilde{M} < (1+\epsilon)M$,*
2. *succeeds with probability at least $1-\delta$,*
3. *uses $O\left(\sqrt{\frac{N}{M}} \frac{1}{\epsilon} \frac{1}{\log \delta}\right)$ queries to U_f,*
4. *uses $O(\log N)$ qubits/space.*

Statement of the algorithm. The algorithm is stated below. It assumes $\theta \leq \pi/1000$, which is without loss of generality since we can always "extend" U_f with dummy indices x which always lead to $f(x) = 0$. For now, we use abstract names c_i to denote parameters in the algorithm, so as to avoid excess detail obscuring the main pedagogical ideas.

1. Set $t = 0$.
2. (Rough estimate) Loop:
   a) Let r be the largest odd integer less than or equal to c_1^t for $c_1 > 1$.
   b) Run $E(r)$ c_2 times, recording each of the c_2 items produced.
   c) If at least $1/3$ of the recorded items x are marked, let $t^* = t$ and exit the loop.
   d) Set $t = t + 1$.

3. Set $\theta_{\min} := \frac{5}{8}\left(\frac{1}{c_1}\right)^{t^*+1}$ and $\theta_{\max} := \frac{5}{8}\left(\frac{1}{c_1}\right)^{t^*-1}$, where recall $c_1 > 1$.

4. Set $t = 0$.

5. (Finetuning the estimate) Loop:
   a) Use the Steady Hands Lemma to choose r.
   b) Run $E(r)$ c_3 times, recording each of the c_3 items produced.
   c) If at least half the items recorded were marked, increase the lower bound via

$$\theta_{\min} = \frac{\theta_{\max}}{\Delta}, \tag{11.9}$$

   where $\Delta := 0.1 + 0.9(\theta_{\max}/\theta_{\min})$. Otherwise, decrease the upper bound via

$$\theta_{\max} = \Delta\theta_{\min}. \tag{11.10}$$

   d) If $\theta_{\max} \leq (1 + \frac{\epsilon}{5})\theta_{\min}$, exit the loop.
   e) Set $t = t + 1$.

6. Return $\widetilde{M} := N \sin^2(\theta_{\max})$.

Formally, setting parameters $c_1 = 12/11$, $c_2 = 10^5 \ln(120/\delta)$, $c_3 = 10^3 \ln(100(0.9)^t/(\delta\epsilon))$ suffices for the proof of Theorem 11.17.

Analysis sketch. We now briefly sketch the proof of Theorem 11.17.

Checkpoint 1: After first loop terminates. The claim here, as suggested by line 3 of the algorithm, is that with probability at least $1 - (\delta/2)$,

$$\frac{5}{8}\left(\frac{1}{c_1}\right)^{t^*+1} = \frac{5}{8}\left(\frac{11}{12}\right)^{t^*+1} \leq \theta \leq \frac{5}{8}\left(\frac{11}{12}\right)^{t^*-1} = \frac{5}{8}\left(\frac{1}{c_1}\right)^{t^*-1}. \tag{11.11}$$

The claim is seen by focusing on a "threshold" t_0 for t, around which the probability of seeing at least $1/3$ of the recorded items marked in Step 2(c) jumps from "insignificant" to "extremely likely". Formally, set t_0 as the largest integer satisfying

$$\left(\frac{12}{11}\right)^{t_0} \theta \leq \frac{5}{8}. \tag{11.12}$$

Then, one can show that if $t < t_0$, the probability p of $E(r)$ returning a marked item is at most

$$p = \sin^2(r\theta) \leq 0.33\left(\frac{12}{11}\right)^{2(1+t-t_0)} < \frac{1}{3}. \tag{11.13}$$

Thus, over m trials, when $t < t_0$ we expect to see strictly less than $1/3$ of the sampled items being marked. Formally, one applies the Chernoff-Hoeffding bound to conclude that since the number of trials is $c_2 = 10^5 \ln(120/\delta)$, the probability of seeing at least $1/3$ of the samples being marked is "small", i.e. at most $\delta/4$.

Exercise 11.18. The Hoeffding bound states the following. Let X_1 through X_n be independent random variables, each satisfying $0 \leq X_i \leq 1$, whose arithmetic mean is $\overline{X} = (X_1 + \cdots + X_n)/n$. Then, the bound states

$$\Pr\left(\left|\overline{X} - E[\overline{X}]\right| \geq t\right) \leq 2e^{-2nt^2}. \tag{11.14}$$

In words, we converge exponentially quickly to the true expectation of $\overline{X}$ by drawing many samples and taking their arithmetic mean. Use the Chernoff bound to show that if a biased coin flip lands HEADS with probability $p - \epsilon$ for fixed $\epsilon > 0$, then it is highly unlikely over n coin flips to have at least pn HEADS instances.

Conversely, as soon as t is "large enough" (formally, $t = t_0 + 1$ suffices), one can show $p > 0.336 > 1/3$, so now a Chernoff bound suffices to conclude that, with probability at least $1 - (\delta/4)$, over $1/3$ of the samples will be marked.

Checkpoint 2: After second loop terminates. By line 5(d), we are guaranteed that if we reach line 6, then

$$\frac{\theta_{\max}}{\theta_{\min}} \leq 1 + \frac{\epsilon}{5}, \tag{11.15}$$

implying any $\widetilde{\theta} \in [\theta_{\min}, \theta_{\max}]$ satisfies

$$\left(1 - \frac{\epsilon}{5}\right)\theta \leq \widetilde{\theta} \leq \left(1 + \frac{\epsilon}{5}\right)\theta. \tag{11.16}$$

Exercise 11.19. Observing that line 6 of the algorithm chooses $\widetilde{\theta} = \theta_{\max}$, show that $\widetilde{M}$ satisfies the first claim of Theorem 11.17.

Roughly, to show that we indeed reach line 6 eventually, observe that after line 3, we have $\Delta = 0.1 + 0.9(\theta_{\max}/\theta_{\min}) \approx 0.1 + 0.9(1.19) > 1$. Thus, intuitively each run of lines 5c and 5d will eliminate a constant fraction of the search space, implying line 5(d) will eventually cause us to exit the second loop, as desired. The only catch is that, each time we remove such a fraction of the search space, we must ensure we do not accidentally "skip" θ, i.e. we must always maintain $\theta \in [\theta_{\min}, \theta_{\max}]$. This is ensured by the following lemma, which shows how to pick r in line $5(a)$ to avoid the situation $\theta \notin [\theta_{\min}, \theta_{\max}]$. Note that the remaining probability of failure of $\delta/2$ in the algorithm arises by applying the union bound over all uses of the following lemma.

Lemma 11.20 (Steady Hands Lemma)**.** *Assume $0 < \theta_{\min} \leq \theta \leq \theta_{\max} \leq \pi/1000$, and that $\theta_{\max}/\theta_{\min} \leq 5/4$. Then, there exists an odd integer r such that, running $E(r)$ at least $1000\ln(1/\delta)$ times and updating $\theta_{\min}$ and $\theta_{\max}$ according to the following rules preserves $\theta_{\min} \leq \theta \leq \theta_{\max}$ with probability at least $1 - \delta$:*

1. *If the majority of samples are marked, update $\theta_{\min} = \theta_{\max}/\Delta$.*
2. *If the majority of samples are unmarked, update $\theta_{\max} = \Delta\theta_{\min}$.*

Further, $r \in \Theta(1/\theta)$, where recall r controls the number of applications of the Grover iterate in $E(r)$.

We will not prove this lemma explicitly, but the rough idea is to select[1] r satisfying

$$r\theta_{\min} \approx \frac{\pi}{2}\left(\frac{\theta_{\min}}{\theta_{\max}-\theta_{\min}}\right) \qquad \text{and} \qquad r\theta_{\max} \approx r\theta_{\min} + \frac{\pi}{2}. \tag{11.17}$$

In words, this not only means r iterations of the Grover iterate "separate" the starting angles $\theta_{\min}$ and $\theta_{\max}$ by an additive angle of approximately $\pi/2$ radians, but that one can additionally show this choice of r aligns $r\theta_{\min}$ with "approximately" $|B\rangle$ (unmarked items) and $r\theta_{\max}$ with $|A\rangle$ (marked items) (recall $|A\rangle$ and $|B\rangle$ have an angle of $\pi/2$ radians). Since before we apply Lemma 11.20, we assume as a precondition that $\theta_{\min} \leq \theta \leq \theta_{\max}$, this means that if $\theta \approx \theta_{\min}$ ($\theta \approx \theta_{\max}$), after r iterations we have $r\theta \approx r\theta_{\min} \approx 0$ modulo 2π ($r\theta \approx r\theta_{\max} \approx \pi/2$ modulo 2π), so we are likely to sample an unmarked (marked) element. Thus, repeating $E(r)$ sufficiently many times and applying a Chernoff bound will ensure that if θ is close to (say) threshold $\theta_{\min}$, then Lemma 11.20 is smart enough to recognize this and to instead update threshold $\theta_{\max}$.

Formally, if $\theta_{\min} \leq \theta \leq \theta_{\max}/\Delta$ (Case 1 of Lemma 11.20), one can show $E(r)$ outputs a marked item with probability

$$p = \sin^2(r\theta) \leq 0.47 < \frac{1}{2}. \tag{11.18}$$

Conversely, if $\Delta\theta_{\min} \leq \theta \leq \theta_{\max}$ (Case 2 of Lemma 11.20), one can show $E(r)$ outputs a marked item with probability

$$p = \sin^2(r\theta) \geq 0.6 > \frac{1}{2}. \tag{11.19}$$

Thus, by the Chernoff-Hoeffding bound, cases 1 and 2 of Lemma 11.20 will correctly update $\theta_{\max}$ or $\theta_{\min}$, respectively, with high probability.

Query complexity. Finally, let us sketch the number of queries to U_f each stage of the algorithm requires.

Exercise 11.21. Recall the first loop is run at most $t_0 + 1$ times, for t_0 defined in Equation (11.12). Recalling that each run of the loops uses $10^5 \ln(120/\delta)$ calls to U_f, show that the total number of queries required by the first loop scales as

$$O\left(\frac{1}{\theta}\log\frac{1}{\delta}\right) \in O\left(\sqrt{\frac{N}{M}}\log\frac{1}{\delta}\right). \tag{11.20}$$

A somewhat similar argument shows that the second loop requires

$$O\left(\frac{1}{\theta\epsilon}\log\frac{1}{\delta}\right) \in O\left(\sqrt{\frac{N}{M}}\frac{1}{\epsilon}\log\frac{1}{\delta}\right) \tag{11.21}$$

queries, which dominates the first loop's runtime, and hence leads to the claimed overall query cost of Theorem 11.17.

[1] Formally, one selects r as the closest integer to $2\pi k/\theta_{\min}$, where k is the closest integer to $\theta_{\min}/(4(\theta_{\max}-\theta_{\min}))$.

12 Stabilizers and the Gottesman-Knill theorem

"So the problem is not so much to see what nobody has yet seen, as to think what nobody has yet thought, concerning that which everybody sees."
— Arthur Schopenhauer, English translation.

According to internet lore, when asked on a physics exam how one should compute the height of a building given just a barometer, a young Nils Bohr allegedly did not give the "expected" answer (e.g. use the barometer to measure pressure at the base versus the top of the building), but rattled off a string of alternate answers (e.g. drop the barometer from the top of the building and measure the time to impact, measure the length of the shadow made by the barometer from atop the roof, etc). While this anecdote is likely more fiction than fact, both it and the quote atop the lecture reveal an important truth — it is often helpful to consider multiple perspectives on a problem.

In particular, in this course, we have used *explicit* representations of quantum states, i.e. throughout our analyses, we tracked the full state vector $|\psi\rangle \in \mathbb{C}^{2^n}$. Naturally, this incurred an exponential classical overhead. But perhaps this overhead is just an artifact of our chosen representation of $|\psi\rangle$? Specifically, we ask:

Is there an alternate representation of quantum states, which allows efficient classical simulation of quantum circuits?

Remarkably, the answer is sometimes *yes*. In this lecture, we introduce a prominent framework for achieving this: The stabilizer formalism. To be clear, the stabilizer formalism does *not* allow efficient simulation of *arbitrary* quantum circuits. However, what it *does* allow is efficient classical simulation of all quantum circuits consisting of so-called *Clifford gates* — this is the content of the celebrated Gottesman-Knill theorem, a proof sketch of which is the main goal of this chapter.

We begin in Section 12.1 by defining the stabilizer formalism, and stating the Gottesman-Knill theorem. Section 12.2 then sketches a proof of the theorem. Note that there is a vast amount further one can say about stabilizers, most importantly in the study of quantum error-correcting codes; however, for reasons of brevity, here we focus on the computational aspect of efficiently simulating (certain classes of) quantum circuits.

12.1 The stabilizer formalism

Briefly, the stabilizer formalism is an alternate, succinct representation of quantum states. We first give intuition in Section 12.1.1. Next, we state the Gottesman-Knill theorem in Section 12.1.2 as the motivating goal of this chapter. Section 12.1.3 then formalizes the stabilizer framework.

12.1.1 Intuition

We build intuition via a sequence of failed attempts (i.e. always cherish the mistakes you make).

Attempt 1: Expansion in the Pauli basis. Just as $|\psi\rangle \in \mathbb{C}^{2^n}$ can be written in terms of a basis for $\mathbb{C}^{2^n}$, the density operator $|\psi\rangle\langle\psi| \in \text{Herm}\left(\mathbb{C}^{2^n}\right)$ can be written in terms of an *operator basis* for $\text{Herm}\left(\mathbb{C}^{2^n}\right)$. For example, any $H \in \text{Herm}\left(\mathbb{C}^2\right)$ can be written as $H = aI + bX + cY + dZ$, for Pauli operators I, X, Y, Z and some $a, b, c, d \in \mathbb{R}$.

Exercise 12.1. Show that $|0\rangle\langle 0| = \frac{1}{2}(I + Z)$. (Hint: Do *not* write out matrices; use spectral decompositions.) What pure state does $\frac{1}{2}(I - X)$ equal?

Exercise 12.2. Show that any single qubit density operator ρ may be written $\rho = \frac{1}{2}(I + r_X X + r_Y Y + r_z Z)$ for *Bloch vector* $\mathbf{r} = (r_X, r_Y, r_Z) \in \mathbb{R}^3$ satisfying $\|\,\mathbf{r}\,\|_2 \leq 1$. What Bloch vector does the maximally mixed state have? How about a pure state $\rho = |\psi\rangle\langle\psi|$?

Thus, any $\rho \in \text{D}\left(\mathbb{C}^2\right)$ can alternatively be expressed via *real* vector $\mathbf{r} \in \mathbb{R}^3$. Does this help? Well, not really — to scale this up to n qubits, we need a corresponding operator basis for $\text{Herm}\left(\mathbb{C}^{2^n}\right)$, such as $P_n := \{\bigotimes_{i=1}^n \sigma_i \mid \sigma_i \in \{I, X, Y, Z\} \text{ for all } i \in [n]\}$, whose properties we now explore.

Exercise 12.3. Write down the operator basis P_2 for $\text{Herm}\left(\mathbb{C}^4\right)$.

Exercise 12.4. To make sense of P_n as a "basis", we require an appropriate notion of "inner product" on this space. For this, we employ the Hilbert-Schmidt inner product[1] on operators, defined as $\langle A, B\rangle := \text{Tr}(A^\dagger B)$ for all Hermitian A, B. Show that for any $M_i, M_j \in P_n$, $\langle A, B\rangle = 2^n \delta_{ij}$. Thus, with respect to the Hilbert-Schmidt inner product, P_n is an orthogonal set.

Exercise 12.5. What is the size of set P_n?

Now, any $H \in \text{Herm}\left(\mathbb{C}^{2^n}\right)$ can be written $H = \sum_{M \in P_n} r_M M$ for $\mathbf{r} \in \mathbb{R}^{4^n}$. The catch is that pesky exponent of 4^n, which follows from Exercise 12.5. In other words, parameterizing $|\psi\rangle\langle\psi|$ via this Pauli operator basis expansion unfortunately still has exponential overhead.

Attempt 2: Casting "shadows[2]" of Pauli basis elements. Attempt 1 used *many* bits ($O(4^n)$ of them, to be precise) to describe *one* state. Attempt 2 asks the opposite: Can we use *few* bits to describe *many states*? Consider, for example, our operator basis element $Z \otimes Z \otimes Z \in \mathcal{L}(\text{Herm}\left(\mathbb{C}^8\right))$. Henceforth, we omit the tensor product and write ZZZ for short.

Exercise 12.6. Prove that ZZZ has precisely two eigenspaces, E_1 and E_{-1}, corresponding to eigenvalues 1 and -1, respectively, each of dimension 4.

[1] This is actually a sleight of hand — by appropriately "reshaping" matrices A and B into vectors v_A and v_B (roughly, v_A is the rows of A concatenated into one big vector, likewise for B), the Hilbert-Schmidt inner product of A and B is actually just the usual inner product of vectors v_A and v_B.

[2] In recent years, the concepts of "shadow tomography" and "classical shadows" have been studied in the quantum computing literature. These are distinct from our use of the term "shadow" here.

In words, we may view ZZZ as "succinctly representing" the 4-dimensional spaces E_1 and E_{-1}, of which we henceforth focus on E_1. More generally, $Z^{\otimes n}$ also has precisely two eigenspaces E_1 and E_{-1}, each of dimension 2^{n-1}. Thus, by writing down just $O(n)$ bits (encoding $Z^{\otimes n}$ requires specifying the Pauli operator at each of the n sites), we can succinctly represent a 2^{n-1}-dimensional space, E_1! This certainly seems interesting, but unfortunately the "shadow" cast by $Z^{\otimes n}$ on $\mathbb{C}^{2^n}$ is "too wide" — we capture *too many* states (i.e. all of E_1). Can we bring the size of this "shadow" under control?

Exercise 12.7. Confirm that $ZZZ|000\rangle = |000\rangle$ and $ZZZ|101\rangle = |101\rangle$, and thus $|000\rangle, |101\rangle \in E_1$. What other two standard basis states are in E_1? What is E_1 more generally, expressed as the span of standard basis states, for arbitrary n?

Attempt 3: Cutting down shadows with set intersections. The natural approach for cutting down the size of a set E_1 is to *intersect* it with another set, E_1'. So consider XX and ZZ. Similar to Exercise 12.7,

$$\begin{aligned} E_1(XX) &= \text{Span}(|x\rangle \mid x \in \{0,1\}^2 \text{ has even Hamming weight }) = \text{Span}(|00\rangle, |11\rangle) \\ E_1(ZZ) &= \text{Span}(H_1 \otimes H_2 |x\rangle \mid x \in \{0,1\}^2 \text{ has even Hamming weight }) = \text{Span}(|{+}{+}\rangle, |{-}{-}\rangle). \end{aligned}$$

Exercise 12.8. Show that $E_1(XX) \cap E_1(ZZ) = \text{Span}(|00\rangle + |11\rangle)$.

Thus, by carefully taking the intersection of two 2-dimensional spaces, $E_1(XX)$ and $E_1(ZZ)$, we obtain a 1-dimensional space. In this particular example, the Bell state $|\phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ is unique state (up to scaling) in the *joint* 1-*eigenspace* of XX and ZZ.

But hold on, you ask — what black magic is this? How can we hope to characterize the types of 1-eigenspaces whose intersection gives rise to an interesting *joint* 1-eigenspace? We will get to the answer shortly, but for now, note that it is crucial that XX and ZZ commute (i.e. $[XX, ZZ] = 0$), and thus *simultaneously diagonalize* (i.e. there exists a basis with respect to which both XX and ZZ are diagonal). The following exercise explicitly reveals this.

Exercise 12.9. Recall the Bell basis $\{|\phi^+\rangle, |\phi^-\rangle, |\psi^+\rangle, |\psi^-\rangle\}$. Show that

$$XX = |\phi^+\rangle\langle\phi^+| - |\phi^-\rangle\langle\phi^-| + |\psi^+\rangle\langle\psi^+| - |\psi^-\rangle\langle\psi^-| \tag{12.1}$$

$$ZZ = |\phi^+\rangle\langle\phi^+| + |\phi^-\rangle\langle\phi^-| - |\psi^+\rangle\langle\psi^+| - |\psi^-\rangle\langle\psi^-|. \tag{12.2}$$

Conclude that XX and ZZ indeed simultaneously diagonalize, and that $|\phi^+\rangle$ is their unique joint 1-eigenvector.

12.1.2 Statement of Gottesman-Knill theorem

With some preliminary intuition in hand, let us now step back and state one of the key applications of the stabilizer formalism.

Theorem 12.10 (Gottesman-Knill Theorem)**.** *Any quantum circuit consisting solely of the following ingredients can be efficiently simulated classically via the stabilizer formalism:*

- *The start state is the all-zeroes string* $|0\rangle^{\otimes n}$.

- *The following gates are allowed: H, CNOT, X, Y, Z, and phase gate*

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{12.3}$$

- *The final state is measured in the standard basis, i.e. via observable $Z^{\otimes n}$.*

We have intentionally stated the theorem in a simpler form; from this, slightly more general statements can be extracted, as demonstrated by the following exercises (neither of which require any knowledge of how the theorem is proven!).

Exercise 12.11. Show that Theorem 12.10 holds even if we start with an arbitrary string $x \in \{0,1\}^n$, or measure an arbitrary Pauli observable $\bigotimes_{i=1}^{\otimes n} \sigma_i$ for $\sigma_i \in \{I, X, Y, Z\}$ at the end.

Exercise 12.12. Show that Theorem 12.10 holds even if we allow intermediate measurements in the standard basis, where all measured qubits are subsequently used as classical controls. (Hint: Use the principle of deferred measurement.)

Thus, amazingly, a non-trivial collection of quantum circuits *can* be simulated efficiently classically, including "genuinely quantum" gates such as H and Z. Of course, we cannot expect to simulate *all* quantum circuits efficiently — most notably, Theorem 12.1.2 does not cover circuits including the T gate

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \tag{12.4}$$

or the Toffoli gate, which has action $|x\rangle|y\rangle|z\rangle \mapsto |x\rangle|y\rangle|z \oplus xy\rangle$ for any $x, y, z \in \{0,1\}$.

Exercise 12.13. What is the relationship between the T and S gates?

12.1.3 Formalizing the stabilizer formalism

We now have intuition (Section 12.1.1) and motivation (Section 12.1.2). It is time to formalize the framework. For this, we denote any operator $\bigotimes_{i=1}^{n} \sigma_i$ with $\sigma_i \in \{I, X, Y, Z\}$ for all $i \in [n]$ as an *n-bit Pauli string*.

The Pauli group. The name of the stabilizer game is to consider joint 1-eigenspaces of Pauli strings. For this, we recall some basic properties: For any $\sigma_i \neq \sigma_j \in \{X, Y, Z\}$, we have $\sigma_i^2 = I$, $\sigma_i\sigma_j = -\sigma_j\sigma_i$ (i.e. Pauli operators anti-commute), and $\sigma_i\sigma_j \propto \sigma_k$ for some $\sigma_k \in \{X, Y, Z\}$ (where the proportionality constant is from set $\{\pm 1, \pm i\}$).

Exercise 12.14. Show that, up to multiplicative factor in $\{\pm 1, \pm i\}$, the product of any two n-bit Pauli strings is another n-bit Pauli string.

Exercise 12.15. Show that, for any Pauli string g, there exists a Pauli string g' such that $gg' = I$ (where I is the Pauli string of identity operators).

Thus, the set of all Pauli strings up to scalars $\{\pm 1, \pm i\}$, i.e.

$$G_n := \left\{ c \cdot \bigotimes_{i=1}^{n} \sigma_i \mid \sigma_i \in \{I, X, Y, Z\} \text{ and } c \in \{\pm 1, \pm i\} \right\}, \tag{12.5}$$

forms a *group* under multiplication. For any subset $S \subseteq G_n$, let $\langle S \rangle \subseteq G_n$ denote the subgroup generated by S (i.e. any element of G_n which may be obtained by taking products of elements in S).

Joint 1-eigenspaces of Pauli strings. As in Section 12.1.1, we now wish to choose a set of generators $S \subseteq G_n$, so that the joint 1-eigenspace of all Pauli strings in S is non-empty, i.e. $E_1(S) \neq \emptyset$ (equivalently, $\dim(E_1(S)) \neq 0$).

Exercise 12.16. Fix any $S \subseteq G_n$. Show that for any $|\psi\rangle$, $|\psi\rangle \in E_1(S)$ if and only if $|\psi\rangle \in E_1(\langle S \rangle)$.

Thus, as far as E_1 is concerned, we may interchangeably talk about a generating set S and the subgroup $\langle S \rangle$ it generates. The set S is then called the *stabilizer* of $E_1(S)$.

Of course, since $E_1(S)$ is implicitly an intersection of sets, the larger S gets, the smaller $E_1(S)$ gets. Thus, we may find ourselves in the predicament that if S is not chosen carefully, $E_1(S) = \emptyset$, as you now show.

Exercise 12.17. Show that if $-I \in S$, then $E_1(S) = \emptyset$.

Exercise 12.18. Show that $S = \{XX, YY, ZZ\}$ has $E_1(S) = \emptyset$. Contrast this with the case of $S = \{XX, ZZ\}$ from Section 12.1.1.

How to pick the generating set S. Luckily, there is a clean characterization of which S indeed produce a non-empty joint 1-eigenspace $E_1(S)$.

Theorem 12.19 (Non-trivial generating set)**.** *Let $S \subseteq G_n$. Then, $E_1(S) \neq \emptyset$ if and only if*

1. *for any $A, B \in S$, $[A, B] = 0$, and*
2. *$-I \notin \langle S \rangle$.*

That these are *necessary* criteria is not difficult to see: We already saw Condition 2's necessity in Exercise 12.17, and now we confirm the same for Condition 1 via the following exercises.

Exercise 12.20. Fix any $S \subseteq G_n$ and any $A, B \in S$. Show that if $[A, B] \neq 0$, then $AB = -BA$. In other words, all elements of S either pairwise commute or anti-commute.

Exercise 12.21. Fix any $S \subseteq G_n$ and any $A, B \in S$, such that $[A, B] \neq 0$, but that there exists $|\psi\rangle \in E_1(S)$. Show that $|\psi\rangle = -|\psi\rangle$, i.e. $|\psi\rangle$ must be the zero vector.

What remains is to show that the criteria in Theorem 12.19 are also *sufficient* to ensure $E_1(S) \neq \emptyset$. Interestingly, one can show a stronger, *quantitative* version of this statement.

Lemma 12.22 (Stabilizer dimension)**.** *For any $1 \leq k \leq n$, let $S = \{g_1, \ldots, g_k\} \subseteq G_n$ be an independent[3] set of generators satisfying the conditions of Theorem 12.19. Then,*

$$\dim(E_1(S)) = 2^{n-k}. \tag{12.6}$$

In other words, not only is it true that increasing S by 1 element shrinks $E_1(S)$, but each added generator cuts the dimension of the encoded space down by a multiplicative factor of 2. Let us now sketch the proof of this lemma.

Proof sketch (Lemma 12.22). The basic approach is rather slick, and best visualized via pizza — if we partition a pizza into 2^k equal slices, one of which represents $E_1(S)$, then our "$E_1(S)$ pizza slice" must span a $1/2^k$-fraction of the pizza. If we now replace "pizza" with $\mathbb{C}^{2^n}$ and "$E_1(S)$ pizza slice" with $E_1(S)$, we obtain $\dim(E_1(S)) = 2^n/2^k = 2^{n-k}$, as claimed.

So how do we partition our Hilbert space pizza into equal-sized spaces? Consider again our example of stabilizer $S = \{XX, ZZ\}$. To begin, let's write down the projector onto our space of interest, $E_1(S)$. For this, define

$$\Pi_{\pm XX} := \frac{I \pm XX}{2} \qquad \text{and} \qquad \Pi_{\pm ZZ} := \frac{I \pm ZZ}{2}. \tag{12.7}$$

Exercise 12.23. Show that Π_{XX} and Π_{ZZ} project onto $E_1(XX)$ and $E_1(ZZ)$, respectively.

The naive way to obtain then the projector onto $E_1(S)$ is to consider $\Pi_{XX}\Pi_{ZZ}$. However, given two projectors P_1 and P_2 onto spaces S_1 and S_2, it is *not* in general true that the projector onto $S_1 \cap S_2$ is P_1P_2. Luckily, here we can leverage the requirement in Lemma 12.22 that all generators *commute*, i.e. $[P_1, P_2] = 0$. In the commuting case, it *is* true that P_1P_2 projects onto $S_1 \cap S_2$. We conclude that since $[\Pi_{XX}, \Pi_{ZZ}] = 0$, $\Pi_{XX}\Pi_{ZZ}$ projects onto $E_1(S)$.

With the projector onto $E_1(S)$ in hand, it remains to devise projectors onto the remaining $2^k - 1 = 3$ ($k = 2$ for $S = \{XX, ZZ\}$) slices of the pizza we claimed existed.

Exercise 12.24. Show that $\Pi_{XX}\Pi_{ZZ}$, $\Pi_{XX}\Pi_{-ZZ}$, $\Pi_{-XX}\Pi_{ZZ}$, and $\Pi_{-XX}\Pi_{-ZZ}$ project onto orthogonal subspaces.

Exercise 12.25. Show that $\Pi_{XX}\Pi_{ZZ} + \Pi_{XX}\Pi_{-ZZ} + \Pi_{-XX}\Pi_{ZZ} + \Pi_{-XX}\Pi_{-ZZ} = I$.

We conclude that $\Pi_{XX}\Pi_{ZZ}$, $\Pi_{XX}\Pi_{-ZZ}$, $\Pi_{-XX}\Pi_{ZZ}$, and $\Pi_{-XX}\Pi_{-ZZ}$ indeed partition $\mathbb{C}^4$ into 4 orthogonal spaces. This part of the proof generalizes easily to all n and k. The last piece of the puzzle is to show that all these orthogonal spaces have *equal* dimension, i.e. are pizza slices of equal size.

Exercise 12.26. Show that conjugation by unitary preserves rank, i.e. for any unitary U and normal operator A, $\operatorname{rank}(UAU^\dagger) = \operatorname{rank}(A)$. Does the same result hold when A is not normal, e.g. when A is not square?

Exercise 12.27. Define $U_1 = X \otimes I \in \mathrm{U}\left(\mathbb{C}^4\right)$. Show that $U_1\Pi_{XX}\Pi_{ZZ}U_1^\dagger = \Pi_{XX}\Pi_{-ZZ}$. Conclude that $\Pi_{XX}\Pi_{ZZ}$ and $\Pi_{XX}\Pi_{-ZZ}$ project onto spaces of the same dimension.

[3] By independent, we mean no generator g_i can be produced as a product of the remaining g_j (and their inverses), i.e. for all i, $g_i \notin \langle\{g_1, \ldots, g_{i-1}, g_{i+1}, \ldots, g_k\}\rangle$.

Exercise 12.28. Find the remaining unitary operators $U_2, U_3 \in \mathrm{U}\left(\mathbb{C}^4\right)$ such that

$$U_2 \Pi_{XX} \Pi_{ZZ} U_2^\dagger = \Pi_{-XX} \Pi_{ZZ} \tag{12.8}$$
$$U_3 \Pi_{XX} \Pi_{ZZ} U_3^\dagger = \Pi_{-XX} \Pi_{-ZZ}. \tag{12.9}$$

Combining these exercises, we have that $\Pi_{XX}\Pi_{ZZ}$, $\Pi_{XX}\Pi_{-ZZ}$, $\Pi_{-XX}\Pi_{ZZ}$, and $\Pi_{-XX}\Pi_{-ZZ}$ indeed partition $\mathbb{C}^4$ into equal-dimensional orthogonal subspaces. We conclude that $\Pi_{XX}\Pi_{ZZ}$ projects onto a 1-dimensional space, as claimed.

Exercise 12.29. Compare Lemma 12.22 with Exercise 12.8. Conclude that the Bell state $|\phi^+\rangle$ is the *unique* state stabilized by $S = \{XX, ZZ\}$. Can you give the stabilizers for the remaining three Bell states?

Exercise 12.30. Use Lemma 12.22 as an alternate way of showing Execise 12.18. (Use the fact that XX, YY, ZZ are independent generators.)

This completes our proof sketch. To formalize this for arbitrary n and k, the main hurdle is systematically finding unitaries analogous to U_1, U_2, U_3 (as in Exercise 12.27), given an arbitrary set of generators satisfying the conditions of Lemma 12.22. This can be accomplished without too much work; however, it goes via the additional formalism of the *check matrix*, which we omit for reasons of brevity.

12.2 Proof of Gottesman-Knill theorem

Finally, we are ready to prove Theorem 12.10. To make good on this claim, we must be able to do three things efficiently: Encode the initial state $|0\rangle^{\otimes n}$, simulate H, $CNOT$, X, Y, and Z gates, and simulate measurements in the standard basis. The first of these is handled in the following exercise.

Exercise 12.31. Give a stabilizer S such that $E_1(S) = \{|0\rangle^{\otimes n}\}$. According to Lemma 12.22, how many generators must S have? (Hint: Do the exercise for the $n = 1$ case first.)

This leaves gate and measurement simulation.

Gate simulation. Normally, for a state $|\psi\rangle$, we would apply a unitary U directly via $U|\psi\rangle$. However, now we are representing $|\psi\rangle$ *implicitly* via a stabilizer S. How does the action of U on $|\psi\rangle$ translate to S? The answer is actually simple: For any $g \in S$,

$$U|\psi\rangle = U(g|\psi\rangle) = Ug(UU^\dagger)|\psi\rangle = (UgU^\dagger)U|\psi\rangle, \tag{12.10}$$

where the first equality holds since $|\psi\rangle \in E_1(S)$. Thus, $U|\psi\rangle$ is stabilized by

$$USU^\dagger = \left\{UgU^\dagger \mid g \in S\right\}. \tag{12.11}$$

In other words, to simulate application of U on $|\psi\rangle$, we *instead apply U to S*.

The second key observation is that the gates listed in Theorem 12.10, H, $CNOT$, X, Y, Z, actually belong to a very special club: They map Pauli strings to Pauli strings, i.e. they act invariantly on group G_n.

Exercise 12.32. Set $g = X^{\otimes n} \in G_n$. Show that $H^{\otimes n} g (H^\dagger)^{\otimes n} \in G_n$. Generalize your proof to arbitrary $g \in G_n$.

Formally, the set of unitaries U mapping Pauli strings to Pauli strings, i.e. which for all $g \in G_n$, $UgU^\dagger \in G_n$, is called the *Clifford group*. Note that to generate the Clifford group, it actually suffices to use just H, $CNOT$, and the S gates (i.e. the Pauli operators listed in Theorem 12.10 are not necessary; they are stated for convenience).

In order to now efficiently simulate gates on our stabilized state, it suffices to explicitly write down how the Clifford group generators $\{H, S, CNOT\}$ act on Pauli strings:

$$\begin{aligned} HXH &= Z && (12.12)\\ HZH &= X && (12.13)\\ SXS^\dagger &= Y && (12.14)\\ SZS^\dagger &= Z && (12.15)\\ CNOT(X_1 \otimes I_2)CNOT &= X_1 \otimes X_2 && (12.16)\\ CNOT(I_1 \otimes X_2)CNOT &= I_1 \otimes X_2 && (12.17)\\ CNOT(Z_1 \otimes I_2)CNOT &= Z_1 \otimes I_2 && (12.18)\\ CNOT(I_1 \otimes Z_2)CNOT &= Z_1 \otimes Z_2, && (12.19) \end{aligned}$$

where we assume the CNOT acts on qubit 1 as control and qubit 2 as target. For example, Pauli string $XZXZ$ is mapped under $CNOT_{12}$ to

$$\begin{aligned} CNOT_{12}(X_1Z_2X_3Z_4)CNOT_{12} &= (CNOT_{12}(X_1I_2I_3I_4)CNOT_{12})(CNOT_{12}(I_1Z_2X_3Z_4)CNOT_{12})\\ &= (X_1X_2I_3I_4)(Z_1Z_2X_3Z_4)\\ &= -Y_1Y_2X_3Z_4, \end{aligned}$$

where the last equality follows since $XZ = -iY$.

Exercise 12.33. Equations 12.12 through 12.19 do not explicitly describe the action of H, S, or $CNOT$ on Pauli Y. How can we recover the action on Y immediately using just the action on X and Z? Use your insight to compute $CNOT(I \otimes Y_2)CNOT$.

Measurement simulation. The last tool we need for Theorem 12.10 is measurement in the standard basis. For this, suppose we have an n-qubit system stabilized by n generators $\langle g_1, \ldots, g_n \rangle$, so that the encoded state $|\psi\rangle$ is unique (up to phase) by Lemma 12.22. We wish to measure a subset of the qubits in the standard (i.e. Z) basis; without loss of generality, assume these are the first k qubits, so that the relevant observable is $Z^{\otimes k} \otimes I^{n-k}$. Recall the projectors onto the $+1$ and -1 eigenspaces of $\overline{Z} := Z^{\otimes k}$, are (respectively)

$$\Pi_+ := \frac{I + Z^{\otimes k}}{2} \qquad \text{and} \qquad \Pi_- := \frac{I - Z^{\otimes k}}{2}. \tag{12.20}$$

where Π_+ (respectively, Π_-) projects onto the span of all even (respectively, odd) Hamming weight k-bit strings. We may now fully characterize the measurement probabilities of Π_+ and Π_- by, again, looking solely at the generators $\langle g_1, \ldots, g_k \rangle$. There are two cases to consider:

- (Case 1: For all $i \in [n]$, $[\overline{Z}, g_i] = 0$.) In this case, either $\overline{Z} \in \langle g_1, \ldots, g_n \rangle$ or $-\overline{Z} \in \langle g_1, \ldots, g_n \rangle$, as you now show.

Exercise 12.34. Show that $\overline{Z}|\psi\rangle$ is stabilized by $\langle g_1, \ldots, g_n\rangle$. Since the stabilized subspace is dimension 1, conclude that $\overline{Z}|\psi\rangle \propto |\psi\rangle$ (Aside: Can $\overline{Z}|\psi\rangle = 0$?)

Exercise 12.35. Why can we now conclude that $\overline{Z}|\psi\rangle = \pm|\psi\rangle$, and why does this yield the desired claim?

Now, if $\overline{Z} \in \langle g_1, \ldots, g_n\rangle$ (respectively, $-\overline{Z} \in \langle g_1, \ldots, g_n\rangle$), then by definition of the stabilizer, $\langle\psi|\Pi_+|\psi\rangle = 1$ (respectively, $\langle\psi|\Pi_+|\psi\rangle = -1$). Thus, we deterministically get outcome $+1$ (respectively, -1) when measuring the stabilized state $|\psi\rangle$ via $\overline{Z}$.

- (Case 2: For some $i \in [n]$, $[\overline{Z}, g_i] \neq 0$.) In this case, you will show that $\langle\psi|\Pi_+|\psi\rangle = \langle\psi|\Pi_-|\psi\rangle = 1/2$, i.e. both outcomes ± 1 occur with equal probability.

  Exercise 12.36. Suppose $[\overline{Z}, g_i] \neq 0$. Then, by Exercise 12.20, $\{\overline{Z}, g_i\} = 0$, i.e. $\overline{Z}$ and g_i anti-commute. Using this, show that

  $$\langle\psi|\Pi_+|\psi\rangle = \langle\psi|\left(\frac{I+\overline{Z}}{2}\right)|\psi\rangle = \langle\psi|\left(\frac{I-\overline{Z}}{2}\right)|\psi\rangle = \langle\psi|\Pi_-|\psi\rangle. \tag{12.21}$$

  (Hint: Use the fact all g_i are Hermitian. Why is this a necessary condition for $\langle g_1, \ldots, g_n\rangle$ to be non-trivial?)

Post-measurement states. It is also easy to update our stabilizer to reflect the post-measurement state. Namely, in Case 1, precisely one of $\overline{Z}$ or $-\overline{Z}$ is in $\langle g_1, \ldots, g_n\rangle$ already, so no update to the stabilizer is necessary. (Intuitively, in Case 1, the measurement outcome occurs with probability 1, meaning $|\psi\rangle$ is not disturbed upon measurement. Thus, its stabilizer need not change.) As for Case 2, we claim that if we assume that $[\overline{Z}, g_j] = 0$ for all $j \neq i$, then whenever we get outcome $+1$ (respectively, -1), the postmeasurement state is stabilized by $\langle g_1, \ldots g_{i-1}, \overline{Z}, g_{i+1}, \ldots, g_n\rangle$ (respectively, $\langle g_1, \ldots g_{i-1}, -\overline{Z}, g_{i+1}, \ldots, g_n\rangle$).

Exercise 12.37. Show the claim above. Where do we use the fact that $[\overline{Z}, g_j] = 0$ for all $j \neq i$?

Exercise 12.38. Why can we assume without loss of generality that $[\overline{Z}, g_j] = 0$ for all $j \neq i$? (Hint: Suppose $\{g_i, \overline{Z}\} = \{g_j, \overline{Z}\} = 0$ for $i \neq j$. What can we say about $[g_i g_j, \overline{Z}]$?)

13 Independent reading - Quantum money

13.1 Quantum money

And now, we have arrived at the end of this course. As the topic of the final lecture, I felt it would be fitting to cover the first known suggested use of quantum information — *quantum money*. Indeed, quantum information processing was *not* first envisioned to solve fancy problems such as factoring or simulation of physical systems. Rather, in the early 1970's, Stephen Wiesner had the idea of using four single-qubit quantum states, $|0\rangle$, $|1\rangle$, $|+\rangle$, $|-\rangle$, to implement an unforgeable notion of money. As the story goes (more accurately, as Gilles Brassard told the story at QIP 2010 in Zurich, and as far as my memory recalls), Wiesner actually proposed his idea as part of a university assignment or project of some sort. His professor did not understand it, and thus it was largely cast aside, only to be published about a decade later (1983, to be precise [Wie83]). In the meantime, however, Wiesner's idea of *conjugate coding* has triumphantly stood the test of time. For example, by happy coincidence, Wiesner happened to share his ideas with Charlie Bennett and Gilles Brassard, who went on to propose their now-famous BB84 protocol in 1984 [BB84] for quantum key distribution (also based on conjugate coding).

Our focus. There is more to quantum money which we will be able to cover in this lecture, as with most topics in this course. Here is what we will focus on. Wiesner's original scheme turns out to indeed be secure, meaning: Given a quantum banknote $|\psi\rangle$ consisting of n conjugate coding states, the probability of creating a duplicate copy of $|\psi\rangle$ is exponentially small in n, i.e. precisely $(3/4)^n$. (Remarkably, it was not until 2012 that this was explicitly shown rigorously by Molina, Vidick, and Watrous [MVW13] using *semidefinite programming*.) Note this security is *information-theoretic*, meaning no computational assumptions are required, other than having the adversary obey the laws of quantum mechanics.

However, security is a fickle thing, and it turns out that if we change the rules of engagement ever so slightly, Wiesner's scheme is *no longer secure*. In particular, as with current-day banknotes, a bank may wish to *verify* that a claimed quantum banknote $|\widetilde{\psi}\rangle$ is genuine. It turns out that, if the bank does what any normal person would, namely return a banknote which passes the authenticity test and confiscate a banknote which fails the authenticity test, then Wiesner's scheme can be *broken*. (Here, the key point is we are now allowing multiple rounds of interaction with an entity, the bank, which performs as its authenticity test the projective measurement $M = \{|\psi\rangle\langle\psi|, I - |\psi\rangle\langle\psi|\}$, for $|\psi\rangle$ the genuine banknote in question.)

Your task. You are to independently read the paper [NSBU16]:

> D. Nagaj, O. Sattath, A. Brodutch, and D. Unruh, An adaptive attack on Wiesner's quantum money. *Quantum Information & Computation*, 16(11&2):1048–1070, 2016. Open-access version at `https://arxiv.org/abs/1404.1507`.

This shows how to use the *quantum Zeno effect* (in the form of the Elitzur-Vaidman bomb testing problem) to successfully create many copies of $|\psi\rangle$, thus breaking Wiesner's scheme. In

particular, you are required to read sections 1 to 3 inclusive of this paper.

Motivation. There are a few reasons for the setup of this final lecture. First, the study of quantum money highlights how careful one need be in formally modelling security in a cryptographic context. Second, Reference [NSBU16] will teach you a neat new trick: Applying the quantum Zeno effect via the Elitzur-Vaidman bomb tester, which is worth having in your toolkit. Finally, the aim of this course is to get you to become independent learners in the field, and so leaving you with an independent reading lecture is arguably quite appropriate. With this in mind, we thus say: Goodbye, farewell, and leave you with one final quote:

> *"Fly my pretties, fly!"*
> — Often (incorrectly) attributed to 1939 film The Wizard of Oz

13.2 What next?

There is much we have not been able to cover given our limited time. Below, we have compiled a short list (which is far from comprehensive) of various sources you may wish to consult to continue your learning (in alphabetical order):

- Andrew Childs `https://www.cs.umd.edu/~amchilds/qa/`: An advanced text focusing on quantum algorithms.
- Ronald de Wolf `https://arxiv.org/abs/1907.09415`: A recent set of introductory course notes to quantum computation, from a theoretical CS perspective.
- Sevag Gharibian `http://groups.uni-paderborn.de/fg-qi/courses/UPB_QCOMPLEXITY/2019/UPB_QCOMPLEXITY_syllabus.html`: An advanced text focusing on quantum complexity theory.
- Griffiths [Gri04]: For those interested in looking beyond our "quantum mechanics sandbox" to what undergraduate quantum mechanics is "really about".
- Nielsen and Chuang [NC00]: The course reference textbook, with much material we did not manage to cover.
- John Watrous `https://cs.uwaterloo.ca/~watrous/TQI/`: An advanced text on quantum information theory, with a computer science orientation.
- Mark Wilde `http://www.markwilde.com/qit-notes.pdf`: An advanced text on quantum information theory, with a strictly information theoretic focus (e.g. entropy, quantum channels, etc).

Bibliography

[AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 333–342, 2011.

[AR20] Scott Aaronson and Patrick Rall. *Quantum Approximate Counting, Simplified*, pages 24–32. 2020.

[BB84] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *IEEE International Conference on Computers, Systems, and Signal Processing*, volume 175, page 8, 1984.

[Bri19] Karl Bringmann. Fine-Grained Complexity Theory (Tutorial). In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019)*, volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:7, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Gri04] D. J. Griffiths. *Introduction to Quantum Mechanics (2nd Edition)*. Pearson Prentice Hall, 2004.

[KLLNP16] M. Kaplan, Gaëtan L., A. Leverrier, and M. Naya-Plascencia. Breaking symmetric cryptosystems using quantum period finding. In *Advances in Cryptology - CRYPTO 2016.*, volume 9815 of *Lecture Notes in Computer Science*, 2016.

[KM10] H. Kuwakado and M. Morii. Quantum distinguisher between the 3-round feistel cipher and the random permutation. In *2010 IEEE International Symposium on Information Theory*, pages 2682–2685, June 2010.

[Kul02] S. R. Kulkarni. Chapter 4: Frequency domain and Fourier transforms. `https://www.princeton.edu/~cuff/ele201/kulkarni_text/frequency.pdf`, 2002.

[Mil76] Gary L. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, 13(3):300 – 317, 1976.

[MVW13] Abel Molina, Thomas Vidick, and John Watrous. Optimal counterfeiting attacks and generalizations for Wiesner's quantum money. In Kazuo Iwama, Yasuhito Kawano, and Mio Murao, editors, *Theory of Quantum Computation, Communication, and Cryptography*, volume 7582 of *Lecture Notes in Computer Science*, pages 45–64. Springer Berlin Heidelberg, 2013.

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[NSBU16] D. Nagaj, O. Sattath, A. Brodutch, and D. Unruh. An adaptive attack on Wiesner's quantum money. *Quantum Information & Computation*, 16(11&12):1048–1070, 2016.

[SS17] T. Santoli and C. Schaffner. Using Simon's algorithm to attack symmetric-key cryptographic primitives. *Quantum Information & Computation*, 17(1&2):65–78, 2017.

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983.