Computing with Quantum Physics

Elisabeth Ann Baseman

April 11, 2011

Submitted to the
Department of Computer Science
of Amherst College
in partial fulfillment of the requirements
for the degree of
Bachelor of Arts with honors

Faculty Advisor: Professor Lyle McGeoch

Abstract

This thesis provides a gentle introduction to the field of quantum computing intended for advanced undergraduate computer science students with little or no physics background. The fundamental goal of quantum computing is to exploit the peculiarities of quantum mechanics in order to develop more powerful computational techniques. Several improvements over what is currently possible classically are known, such as factoring numbers in polynomial time, database search in $O(\sqrt{N})$ time, and provably secure key distribution for public-key cryptosystems. Many questions, including whether quantum computers are fundamentally more powerful than classical computers, remain open.

Acknowledgements

I'd like to thank my advisor, Professor Lyle McGeoch, for his guidance and the many hours we spent in his office working through derivations. I'm grateful to Professor Friedman for access to his lecture transcripts and for encouraging my interest in physics. Thanks to Mom, Dad, Cynthia, Eleanor, Alex, Cathrina, Jacquelyn, Brieanna, Emily, Jenna, Jamie and Gavin for all their help and support.

Contents

Chapter 1

Introduction

Computer science is fundamentally linked to the physical world.

The machines we use to access the Internet, the Java programs written by a student in an introductory computer science class, and indeed everything we generally refer to as a "computer" is based on a set of physical assumptions. Unfortunately, the underlying physical assumptions on which nearly all modern computers rely are outdated. The twentieth century saw the rise of quantum mechanics, revealing strange phenomena including superposition and entanglement, while we are still stuck with computers based on classical physics. The new and incredibly rich field of quantum computation and information seeks to harness the power of quantum mechanics in order to develop significantly more powerful computational techniques [10].

The classical picture of the world is intuitive. We think of electric current as the motion of electrons across a voltage, and of electrons as discrete particles. They cannot be in more than one state simultaneously, and they have a definite location, momentum, energy and time. Quantum mechanics is more bizarre.

1.1 Quantum Weirdness

The well-known "Schrödinger's Cat" thought experiment demonstrates several of the unexpected properties we observe in the world of modern physics.

Suppose we put a live cat in a box that also contains a vial of poisonous gas hooked up to a two-state quantum system. If the system is in the first (ground) state, the poison is released and the cat dies. If the system is in the second (excited) state, the poison is not released and the cat is allowed to live. The system is designed such that at any point in time there is a 50% chance that a transition between the excited and ground states has occurred. The system is initially in the excited state.

We have now sealed the box containing the cat, the poison gas, and the quantum system. For the sake of this thought experiment, assume that we now have no information about what is happening inside the box. We cannot hear the cat or the machinery inside, and we cannot even use very sensitive instruments to detect heat from the cat's body through the walls of the box.

Is the cat alive or dead? Because we gave the cat a 50% chance of living, we cannot know without looking inside the box. So, what can we say about the state of the cat without opening the box?

Because "life" is the quantity we wish to measure, we are interested in finding a representation of the cat's state with respect to the two basis states, $|\text{alive}\rangle$ and $|\text{dead}\rangle$ (recall the notion of a basis from linear algebra). The "$|\ \rangle$" notation, pronounced "ket," indicates a quantum state and is known as *Dirac notation*. The representation of a quantum state is a linear combination of the basis states, with the basis states usually determined by the measurement you plan to make.

Due to the 50%-chance-of-living constraint, our description of the cat's state should indicate a 50% chance of measuring each of the basis states. Clearly, the probabilities of measuring each of the basis states should sum to 1 for any system. It turns out that the proper way to express this requirement is not to use the probabilities as coefficients, but instead to require that the sum of the squares of coefficients is 1. Therefore, we describe the state of the cat inside the closed box as $|\psi\rangle = \frac{1}{\sqrt{2}}|\text{alive}\rangle + \frac{1}{\sqrt{2}}|\text{dead}\rangle$.

The meaning of this state is extremely counterintuitive — the cat is in a "superposition" of $|\text{alive}\rangle$ and $|\text{dead}\rangle$. That is, instead of being either alive or dead, it is both alive and dead simultaneously. The state of the cat only *projects* onto one or the other of these basis states (which we would intuitively expect to see) when we open the box and measure the system. Until we open the box and measure the system, we cannot know whether the cat is alive or dead. In addition, after making a measurement, there is no way to reverse the measurement process and reconstruct the former superposition of the system. Measuring the system causes it to project onto one of the basis states (with probability determined by the square of the coefficients) and destroys the previous state of the system.

While Schrödinger's Cat is just a thought experiment, it demonstrates some fundamental quantum-mechanical behaviors that have never been contradicted by experiment. The next section describes a classic experiment that demonstrates some of the quantum weirdness we have discussed. Throughout this paper, we will be treating quantum-mechanical systems in much the same way we treated Schrödinger's Cat, as a black box that has some interesting properties, but whose inner workings we do not understand.

For a more rigorous discussion of "Schrödinger's Cat," the interested reader should consult Nielsen and Chuang [21].

1.2 The Double Slit Experiment

Imagine firing a stream of classical particles at a barrier that has one slit cut in it just large enough to allow a single particle to pass through and hit a collection apparatus some short distance away on the other side. If the particle gun fires each particle at a random angle, we would expect our data to show that we have the greatest chance of finding a particle directly behind the slit, and that the probability falls off on both sides. Similarly, if we were to have two slits, we would see two peaks in the probability of finding a particle, each directly behind a slit, falling off according to a Gaussian curve elsewhere.

Now imagine doing the same experiment, except instead of shooting classical particles we put the entire setup in a shallow pool of water and send waves toward the two slits. Now we expect to observe an interference pattern. Our results show the greatest received intensity halfway between the two slits, with damped oscillations dropping off on both sides.

What results do we get if we actually perform this experiment with electrons or any other kind of elementary particle? Classical physics says that our data should agree with the first experiment. Indeed, if our collection apparatus is similar to a Geiger counter, emitting a click if it detects an

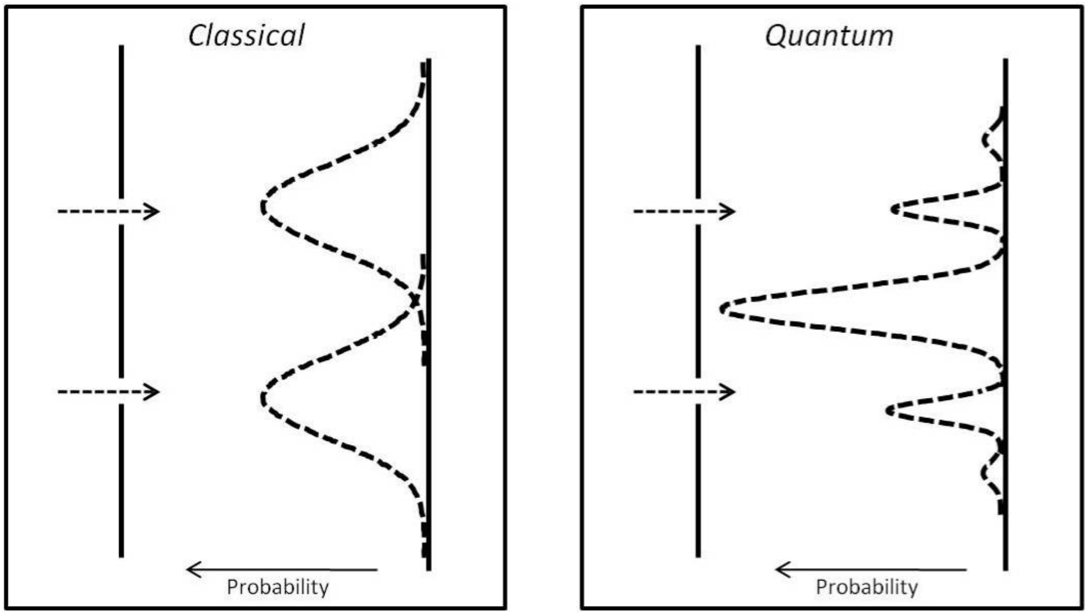

Figure 1.1: The left side shows the results expected classically from the double slit experiment. The particles enter through both slits and have the highest probability of being found directly across from each slit. The right side shows the large-scale actual results that we see because of quantum mechanics. The particles pass through the slits and then have the highest probability of being found halfway between the slits, as if they were interfering waves.

electron, we only hear whole clicks and we never hear two clicks at a time. We always find one electron on the other side of one of the slits, but never behind both. So far, the results are exactly what we would expect classically. However, if we run the experiment multiple times and look at the probabilities of finding an electron on the other side of the barrier, we see a peak halfway between the slits with damped oscillations on both sides, just as if the electrons were waves. Figure 1.1 shows the large-scale structure of both this and the classical result. The aggregate results show electrons interfering like waves, and we find that when we detect an electron on the other side of the barrier, there is no way to determine which slit it travelled through. These observations confirm wave-particle duality and give rise to one of the basic ideas of quantum mechanics: unlike in classical physics, we cannot predict exactly what will happen in a given situation. We can only talk about the probability amplitudes of different outcomes. The experimental results are the same for all elementary particles, including photons (light particles).

For a more rigorous description of the double slit experiment, see Feynman [9].

1.3 Quantum Computing

What if we build a computer using the physical assumptions of quantum mechanics, rather than the very different assumptions of classical physics? Imagine that instead of using a bit which registers either 0 or 1, we use a two-state quantum system, similar to the apparatus in the box with Schrödinger's cat. This two-state system is what we call a *qubit* (quantum bit), the fundamental unit of information in quantum computing [4].

We are interested in whether the qubit is a 0 or a 1 (like a classical bit), so our basis states are $|0\rangle$ and $|1\rangle$. As in our treatment of Schrödinger's cat, this means that qubits can exist in superpositions — they can be both 0 and 1 simultaneously. This feature of quantum mechanics is particularly important to quantum computation because it allows us to work with all possible values of a qubit simultaneously. For example, we can calculate all possible values of a function on a single superposed qubit. Although it may initially sound like quantum computers are obviously more powerful than classical computers, recall that measurement causes the system to project onto one basis state and destroys the overall quantum state of the system. This means that while we may be able to calculate all possible values of a function, we can only measure one of them. Even worse, the one we measure is determined randomly when we take the measurement. This is one challenge in the art of designing quantum algorithms: figuring out how to manipulate the qubits so that the results of our measurements have a meaningful interpretation, which hopefully provides us with more information than we would have gotten using a classical computing paradigm.

Chapter 2

Qubits and Quantum Gates

2.1 Representing a Single-Qubit System

As we have seen, qubits can behave in surprising ways.[1] Before we get to performing computation, we need to formally develop some of the ideas we have discussed. Unlike classical bits, which register either 0 or 1, the quantum state of a qubit can be any linear combination of the 0 state and the 1 state, written

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

where α, $\beta \in \mathbb{C}$.[2]

α^2 and β^2 correspond to the probabilities with which a measurement of the state of the qubit would result in $|0\rangle$ and $|1\rangle$, respectively. Therefore, we require that

$$\alpha^2 + \beta^2 = 1$$

[1]The reader may be skeptical that qubits actually exist outside of a mathematical construct. In fact, there are many ways to implement qubits in the real world. A common example of a qubit is an electron. Electrons have the peculiar property that they have some intrinsic angular momentum (*spin*), which happens to always be either $+\frac{1}{2}$ or $-\frac{1}{2}$ when we measure it. To use an electron as a qubit, we can just interpret the positive and negative spins as $|0\rangle$ and $|1\rangle$. Designing and building the physical apparatus to manipulate and store qubits is difficult, but we will leave that to the experimental physicists and engineers. Other common ways to implement qubits include using photon polarization or the excited and ground states of two-level atoms.

[2]We let α, $\beta \in \mathbb{C}$ instead of $\mathbb{R}$ so that we can account for phase differences, a physical phenomenon that is best represented by use of complex numbers. We will see phase factors play a role later in this section.

For the purposes of quantum computing, we choose to use the Heisenberg formulation of quantum mechanics, which makes heavy use of matrices. In the Heisenberg approach, the $|0\rangle$ and $|1\rangle$ states can be represented as 2×1 matrices known as *spinors*:

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

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

Similarly, it will be helpful to think of a general qubit state as a matrix:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Our third option for describing the state of a single qubit is the *Bloch sphere representation*, which provides us with a geometric interpretation of the state of a qubit. In this representation, we re-write the state of the qubit as

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

To see that this can be done, first note that multiplying the state by an overall phase factor has no observable effects. Consider what we get if we multiply each of the coefficients by a global phase factor, $e^{i\gamma}$ [13]:

$$\begin{aligned} |e^{i\gamma}\alpha|^2 + |e^{i\gamma}\beta|^2 &= (e^{i\gamma}\alpha)^*(e^{i\gamma}\alpha) + (e^{i\gamma}\beta)^*(e^{i\gamma}\beta) \\ &= e^{-i\gamma}\alpha^* e^{i\gamma}\alpha + e^{-i\gamma}\beta^* e^{i\gamma}\beta \\ &= e^0\alpha^*\alpha + e^0\beta^*\beta \\ &= \alpha^*\alpha + \beta^*\beta \\ &= |\alpha|^2 + |\beta|^2 \end{aligned}$$

So, we see that an overall phase factor does not affect the values of the squares of the coefficients, and therefore does not affect the overall result of a qubit measurement. This means that we can neglect any global phase factors that we come across in our derivations.[3]

[3] A global phase factor is different from a *relative* phase factor. Two quantum states differ by a relative phase factor in some particular basis if there exists a real θ such that each of the coefficients, a and b, in that basis are related by $a = e^{i\theta}b$. Because relative phase is basis-dependent, it is observable and cannot be neglected in the way that global phase can [21].

Because α and β are complex, we will re-write $|\psi\rangle$ as:

$$|\psi\rangle = r_\alpha e^{i\varphi_\alpha}|0\rangle + r_\beta e^{i\varphi_\beta}|1\rangle$$

with r_α, φ_α, r_β, $\varphi_\beta \in \mathbb{R}$.

Then for simplification we can multiply by an overall phase factor of $e^{-i\varphi_\alpha}$:

$$\begin{aligned} |\psi\rangle &= e^{-i\varphi_\alpha}(r_\alpha e^{i\varphi_\alpha}|0\rangle + r_\beta e^{i\varphi_\beta}|1\rangle) \\ &= r_\alpha|0\rangle + r_\beta e^{i\varphi}|1\rangle \end{aligned}$$

where we let $\varphi = \varphi_\beta - \varphi_\alpha$.

Now, consider what we get if we switch back to the Cartesian representation of the coefficients, so β becomes $(x+iy)$, and apply our probability requirement (known as the *normalization constraint*):

$$\begin{aligned} \alpha^2 + \beta^2 &= |r_\alpha|^2 + |x+iy|^2 \\ &= r_\alpha^2 + (x+iy)^*(x+iy) \\ &= r_\alpha^2 + (x-iy)(x+iy) \\ &= r_\alpha^2 + x^2 + y^2 = 1 \end{aligned}$$

This gives us the equation of a unit sphere in 3-dimensional space! Transforming to the usual spherical polar coordinate system yields:

$$|\psi\rangle = \cos\theta|0\rangle + e^{i\varphi}\sin\theta|1\rangle$$

with θ, $\varphi \in \mathbb{R}$.

A little further investigation reveals that θ need only vary between 0 and $\frac{\pi}{2}$ in order to generate all points on the sphere, due to our ability to neglect multiplication by overall phase factors. Then, we arrive at our Bloch sphere representation of a qubit, as we saw before,

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

where θ, $\varphi \in \mathbb{R}$, $0 \le \theta \le \pi$, $0 \le \varphi \le 2\pi$. We can imagine the state of a qubit as a vector from the origin to a point on the Bloch sphere – a qubit in the $|0\rangle$ state is a vector pointing straight up, while a qubit in the $|1\rangle$ state is a vector pointing straight down. It should be clear then that a qubit in a 50/50 superposition of $|0\rangle$ and $|1\rangle$ is a vector pointing horizontally, possibly rotated in the horizontal plane due to a phase difference, because intuitively this 50/50 superposition is "halfway between" the two basis states. When we apply operations, or gates, to qubits, it is sometimes useful to envision the operations as rotations or reflections within the Bloch sphere. For example, a gate that applies a negation operation would take $|0\rangle$ to $|1\rangle$, which is a reflection.

We have seen three ways to think about and write down qubits: Dirac notation, matrix notation and the Bloch sphere representation. The Dirac bra-ket notation[4] is useful for concise notation of qubit states, and comes in handy for writing down systems of many qubits, as we will see later. It is closely tied to the matrix notation because each ket is really just shorthand for a spinor. The matrix notation is most useful to us when we are trying to figure out the result of applying some operations to a qubit (or a system of qubits), and the Bloch sphere representation helps to give us some intuitive physical notion of how we might imagine the state of a qubit.

2.2 Single-Qubit Gates

Now that we have some machinery for thinking about qubits, let's investigate how we might be able to apply some simple operations. It turns out that in quantum mechanics all operations must be reversible. This is not true for classical computing. Consider a classical AND gate. If the AND gate outputs a 0, we cannot run the computation backwards and definitively retrieve the two inputs. In order to obey the rules of quantum mechanics, we must make sure that all of our quantum gates are reversible.[5] This reversibility constraint, combined with our requirement that the squares of the coefficients of the states sum to 1, ends up meaning that all of our gates must be unitary matrices. That is, we require that the gate's conjugate transpose also be its inverse:

$$U^\dagger U = (U^T)^* U = (U^*)^T U = \mathbb{I}$$

[4]So far we have only come across kets. Later on we will encounter their complex conjugates, known as *bras* and written $\langle|$. These are named so that when they are combined, which corresponds to taking a dot product, we can pronounce $\langle|\rangle$ as "bracket."

[5]The terms "operations" and "gates" will be used interchangeably throughout this thesis.

where U is a square matrix of complex numbers and $\mathbb{I}$ is the identity matrix.

In order to apply an operation, we simply multiply the state of the qubit by the corresponding unitary matrix. Three unitary matrices that will be of particular interest to us are the *Pauli spin matrices*:

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

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

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

Let's take a look at what happens to a general qubit if we apply the σ_x gate:

$$\sigma_x|\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

The σ_x gate flips the coefficients of the qubit state. This means that we've found our negation operation! As we mentioned before, the negation operation can also be thought of as a reflection in the Bloch sphere.

In fact, *any* unitary matrix can be a quantum gate.[6] For example, a general rotation by θ of a qubit state about an axis in the Bloch sphere, say $\hat{j}$, is [18]:

$$\begin{aligned}
R_{\hat{j}}(\theta) &= e^{-i\frac{\theta}{2}\sigma_{\hat{j}}} \\
&= \mathbb{I} - i\frac{\theta}{2}\sigma_{\hat{j}} - \frac{(\frac{\theta}{2}\sigma_{\hat{j}})^2}{2!} + i\frac{(\frac{\theta}{2}\sigma_{\hat{j}})^3}{3!} + \frac{(\frac{\theta}{2}\sigma_{\hat{j}})^4}{4!} - i\frac{(\frac{\theta}{2}\sigma_{\hat{j}})^5}{5!} + \ldots \\
&= \mathbb{I} - i\frac{\theta}{2}\sigma_{\hat{j}} - \frac{\left(\frac{\theta}{2}\right)^2\mathbb{I}}{2!} + i\frac{\left(\frac{\theta}{2}\right)^3\sigma_{\hat{j}}}{3!} + \frac{\left(\frac{\theta}{2}\right)^4\mathbb{I}}{4!} - i\frac{\left(\frac{\theta}{2}\right)^5\sigma_{\hat{j}}}{5!} + \ldots \\
&= \left(1 - \frac{\left(\frac{\theta}{2}\right)^2}{2!} + \frac{\left(\frac{\theta}{2}\right)^4}{4!} + \ldots\right)\mathbb{I} - i\left(\frac{\theta}{2} - \frac{\left(\frac{\theta}{2}\right)^3}{3!} + \frac{\left(\frac{\theta}{2}\right)^5}{5!} + \ldots\right)\sigma_{\hat{j}} \\
&= \cos\left(\frac{\theta}{2}\right)\mathbb{I} - i\sin\left(\frac{\theta}{2}\right)\sigma_{\hat{j}}
\end{aligned}$$

[6] This means that there is an uncountable (continuous) number of possible valid quantum gates, which implies that we cannot come up with a universal finite set of quantum gates. We end up being able to have an arbitrarily good approximation of any unitary gate using a finite universal set of gates. We will leave the heart of this issue to the physicists and mathematicians, and take it as given that we can build a circuit that will perform whatever unitary operation we desire.

So far we haven't really exploited any quantum weirdness in our manipulation of qubits, except for the fact that we are allowed to use any unitary matrix as a gate. Clearly, if we have a qubit that's in a superposition, the result of applying any of these gates will also be a superposition. So, how do we put a qubit into a superposition in the first place? The answer is another unitary matrix, the *Hadamard gate*:

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

If we apply the Hadamard gate to a qubit that is in a pure $|0\rangle$ state, we get:

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \end{aligned}$$

And, we can do the same thing for a qubit in a pure $|1\rangle$ state:

$$\begin{aligned} H|1\rangle &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{aligned}$$

In both of the above cases, the qubit ends up in a superposition. The reader should recognize the first superposition as the same state in which we had Schrödinger's cat. The second superposition is similar but with a different sign. Then, the reader can verify that a Hadamard gate on a general qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ has the following effect:

$$H|\psi\rangle = \frac{(\alpha+\beta)|0\rangle + (\alpha-\beta)|1\rangle}{\sqrt{2}}$$

Hadamard gates are used ubiquitously in quantum algorithms and are particularly useful for putting qubits into the desired input states for quantum circuits.

We can construct a desired input state by measuring it so that it projects onto one of the pure basis states, and then applying a Hadamard gate to put it into a superposition. If the initial

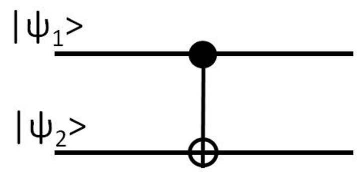

Figure 2.1: The circuit symbol used to represent a CNOT gate, where the upper qubit is the control qubit.

measurement does not give us the basis state we were looking for, we can keep getting new qubits and measuring them until one of them randomly projects onto the desired basis state.

2.3 Two-Qubit Systems

Einstein did not believe in quantum mechanics, and derived what has become known as the EPR (Einstein-Podolsky-Rosen) Paradox. Einstein found that by combining two particles in a certain way they would become *entangled*, meaning that they would be perfectly correlated. Measuring the first particle would apparently instantaneously cause the second particle to project onto the corresponding state, no matter the distance between the particles [11]. This seemed to be a violation of relativity and the axiom that nothing can travel faster than the speed of light. However, the existence of so-called *EPR pairs* has been tested extensively and confirmed. In terms of quantum computing, this implies that if we measure one qubit, we can also know something about a second qubit in the same circuit.

We can construct an entangled pair as follows. Suppose we are dealing with a two-qubit quantum circuit where the first qubit is in state $|\psi_1\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and the second qubit is in state $|\psi_2\rangle = |0\rangle$. First we put the qubits through a controlled-NOT gate (CNOT). A CNOT gate takes in two qubits and flips the value of the second qubit if the first qubit is in the $|1\rangle$ state. If the first qubit is in the $|0\rangle$ state, the second qubit remains unchanged. If either or both of the input qubits are in a superposition, the output of the CNOT gate will also be in a superposition. Figure 2.1 shows the symbol used to represent a CNOT gate in a quantum circuit, and its matrix representation is

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

Notice that the upper left of C is the identity matrix while the lower right is σ_x, which performs negation. For example, consider what happens when we apply the CNOT gate to the state $|01\rangle$, meaning that the first qubit (the control) is in the $|0\rangle$ state and the second qubit is in the $|1\rangle$ state:

$$\begin{aligned} C|01\rangle &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \\ &= |01\rangle \end{aligned}$$

So the state remains unchanged. However, if we send the two-qubit state $|10\rangle$ through a CNOT gate, we get

$$\begin{aligned} C|10\rangle &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\ &= |11\rangle \end{aligned}$$

As expected, the control qubit stays the same and the value of the second qubit flips.

If we apply a CNOT gate to the two qubits we are entangling, using the first qubit as the control, the overall state of the system is

$$|\psi\rangle = C\frac{(|0\rangle + |1\rangle) \otimes |0\rangle}{\sqrt{2}} = C\frac{|00\rangle + |10\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{2.1}$$

The $\otimes$ symbol indicates a tensor product, which we will not treat formally here. For now, we will think of tensor products as being similar to concatenation with some interesting properties. In particular, tensor products are multilinear, meaning that $a|\psi_1\rangle \otimes |\psi_2\rangle = |\psi_1\rangle \otimes a|\psi_2\rangle = a(|\psi_1\rangle \otimes |\psi_2\rangle)$. The multilinearity is what lets us ignore global phase factors on either qubit in the system. The first entry in each two-qubit ket corresponds to the first qubit, and the second entry to the second qubit. We see that applying the CNOT gate has no effect on the $|00\rangle$ state, because the first qubit is in the $|0\rangle$ state. However, it takes the $|10\rangle$ state to $|11\rangle$ because the first qubit is in the $|1\rangle$ state.

Now we can read off directly from Equation (2.1) that there is a 50% chance of finding the system in state $|00\rangle$ and an equal chance of finding it in state $|11\rangle$. In other words, because there are only two possible states, if we measure either qubit and get $|0\rangle$, we automatically know that the other qubit must also be in $|0\rangle$, and similarly if our measurement returned $|1\rangle$. Our two qubits are entangled! Measuring either one apparently has an instantaneous effect on the other, and, as we will see later, this can be exploited in clever ways to speed up computation and communication.

However, entanglement has to be treated carefully. We are not in general able to exploit entanglement in an arbitrary two-qubit system. Let's consider a two-qubit quantum circuit, where each qubit is in an even superposition. That is, $|\psi_1\rangle = |\psi_2\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$. In quantum mechanics we have to consider the system as a whole rather than look at individual qubits (except when taking measurements), so we write the overall state of the system:

$$\begin{aligned} |\psi\rangle &= \frac{(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)}{2} \\ &= \frac{|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle}{2} \\ &= \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} \end{aligned}$$

This system exists in an even superposition of all four possible basis states (for a two-qubit system), so clearly if we measure either qubit we still don't have any more information about the state of the other qubit. Imagine measuring the first qubit and getting $|1\rangle$. This implies that the system is either in $|10\rangle$ or $|11\rangle$ with equal probability, so the second qubit is just as likely to be in the $|0\rangle$ state as the $|1\rangle$ state. The qubits are not correlated, and cannot be used in the same way as two entangled qubits.

There are four basis states for a system of two entangled qubits, called the *Bell states*, of which Equation (2.1) is one:

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

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

$$|\beta_{10}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

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

Recall that we were able to produce the first Bell state using a Hadamard gate and a CNOT gate, with the input $|0\rangle \otimes |0\rangle$. The interested reader can deduce the quantum circuits necessary to produce the other three Bell states.

The reader may also notice that in our discussion of two-qubit operations we looked at CNOT and entanglement, but did not mention copying one qubit's state onto another qubit. The *no-cloning theorem* of quantum mechanics states that it is impossible to make a copy of a quantum state, so there will be no copying of qubits allowed. We only get one chance to measure the qubit and destroy its state. This is another reason why measurement is tricky, especially if we need to work with a qubit that has been sent to us and we do not know what previous operations have been performed on it.

2.4 Multiple-Qubit Systems

We treat systems with n qubits in much the same way as we dealt with two-qubit systems. As usual in quantum mechanics, we think about the state of the system as a whole rather than individual qubits. Generalizing from the two-qubit system, the state of the entire system is the tensor product of all the individual n qubits:

$$|\psi\rangle = \bigotimes_{i=1}^{n} |\psi_i\rangle$$

Just as a two-qubit system has four basis states ($|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$), an n-qubit system has 2^n basis states. That is, an n-qubit system can be in a superposition of the values (represented in binary) 0 through $2^n - 1$. This means that it is possible for us to do computation on all of these values simultaneously, but again in the end we only get to measure one random value. In the next chapter we'll see the simplest example of how to manipulate a quantum state so that even though the result of measurement is random we can deduce useful information from it.

2.5 Complexity

Just as in classical computing, we evaluate the complexity of quantum algorithms based on the length of the input. However, the analysis of quantum algorithms is less obvious than many classical algorithms. An important measure of complexity of a quantum algorithm is the minimum number of quantum gates it requires. Finding this minimum is not always easy. But, sometimes using fewer gates necessitates the use of more qubits, which can be a practical problem at the implementation stage. In addition, due to the probabilistic nature of quantum mechanics, most non-trivial quantum algorithms must be repeated until they return the answer with high enough probability. This means that we need to do a careful analysis of these probabilities and factor them into the runtime. We will not be walking through any detailed runtime analysis in the following chapters. The interested reader should consult Cleve [6].

Chapter 3

Introduction to Quantum Algorithms: The Deutsch-Jozsa Algorithm

3.1 Deutsch's Problem

The first quantum algorithm we will examine gives us a solution to Deutsch's problem: Given a "black box" that computes a function on a one-bit input, determine whether the function is constant (gives the same output for both 0 and 1), or balanced (outputs the same number of 0s as 1s). Note that there are only four functions of this kind, so clearly this is a toy problem. However, it gives us a solid introduction to quantum algorithm design. The interesting aspect of Deutsch's problem arises when we ask how many queries we must make to our black box in order to determine whether the function is constant or balanced. Classically, of course, we would always need two queries to the black box, one for each possible input, in order to know with certainty about this global property of the function in question. Using quantum computing, we will find that we only need one query!

3.2 The Procedure

Figure 3.1 shows the quantum circuit for solving Deutsch's problem. The algorithm uses two qubits, two Hadamard gates, and a "black box" unitary operation that computes the function we are investigating. It requires that the qubits be measured at different points in the computation. We are already well equipped to take a close look at each of the states of this two-qubit system.

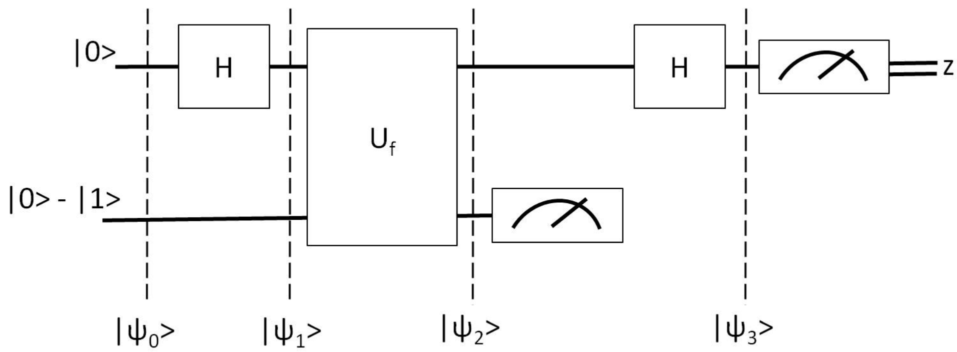

Figure 3.1: The Deutsch-Jozsa Algorithm, with each state in the sequence of computations labelled. The input states of the two qubits are shown, with the second qubit in a superposition. Note that the normalization constants are not shown. The circuit includes two Hadamard gates (H) and a "black box" that computes the function in question (U_f). The meter symbols indicate measurement. Computation flows from left to right as in a classical circuit, but in the actual physical implementation the qubits may not move between different physical locations. Instead, computation can be performed on stationary qubits as time elapses. The output of the circuit is represented as a double line to indicate a classical bit rather than a qubit.

The first overall state is just the tensor product of the two input qubits:[1]

$$|\psi_0\rangle = |0\rangle \otimes (|0\rangle - |1\rangle) = |00\rangle - |01\rangle$$

Then to obtain $|\psi_1\rangle$ we apply a Hadamard gate to the first qubit:

$$\begin{aligned} |\psi_1\rangle &= (H|0\rangle) \otimes (|0\rangle - |1\rangle) \\ &= (|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle) \\ &= |00\rangle - |01\rangle + |10\rangle - |11\rangle \end{aligned}$$

The unitary matrix which computes the mystery function uses the first qubit as the input to the function and writes the output onto the second qubit. The result of applying the black box matrix is that the first qubit remains unchanged and the second qubit gets XORed with the function

[1]Throughout our discussion of this algorithm we will leave off the normalization constants because they will not affect the results of the measurements we make. The reader should be able to insert the appropriate constants based on our definitions of the matrices associated with single-qubit quantum gates.

output [8]:

$$\begin{aligned}
|\psi_2\rangle &= U_f|\psi_1\rangle \\
&= U_f(|00\rangle - |01\rangle + |10\rangle - |11\rangle) \\
&= |0, 0 \oplus f(0)\rangle - |0, 1 \oplus f(0)\rangle + |1, 0 \oplus f(1)\rangle - |1, 1 \oplus f(1)\rangle \\
&= |0, f(0)\rangle - |0, \overline{f(0)}\rangle + |1, f(1)\rangle - |1, \overline{f(1)}\rangle \\
&= \sum_{x=0,1} (|x, f(x)\rangle - |x, \overline{f(x)}\rangle) \\
&= \sum_{x=0,1} (-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle)
\end{aligned}$$

where x takes on only the discrete values 0 and 1 within the summations. Before we apply the second Hadamard gate to the first qubit, we measure the second qubit. Note that part of the contribution to $|\psi_2\rangle$ from the second qubit is just the constant $(|0\rangle - |1\rangle)$. Measuring the second qubit will remove this constant from the system, leaving only the terms in the summation that depend on x. Strangely, the $(-1)^{f(x)}$ factor, which we would intuitively associate with the second qubit, stays behind and apparently becomes "attached" to the first qubit after the measurement. This is part of the weirdness of quantum mechanics, but it is crucial to the behavior of the Deutsch-Jozsa Algorithm. After measuring away the second qubit, we apply the second Hadamard gate to the first qubit:

$$\begin{aligned}
|\psi_3\rangle &= H \sum_{x=0,1} (-1)^{f(x)}|x\rangle \\
&= H\left((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\right) \\
&= (-1)^{f(0)}(|0\rangle + |1\rangle) + (-1)^{f(1)}(|0\rangle - |1\rangle) \\
&= \left((-1)^{f(0)} + (-1)^{f(1)}\right)|0\rangle + \left((-1)^{f(0)} - (-1)^{f(1)}\right)|1\rangle
\end{aligned}$$

Consider what this tells us if we measure the qubit and get $|0\rangle$. It would tell us that the coefficient of $|0\rangle$ was non-zero, and therefore $f(0) = f(1)$. If the result of the measurement is $|1\rangle$, we know the coefficient of $|1\rangle$ was non-zero, so $f(0) \neq f(1)$. This means that if we measure the qubit to be in the $|0\rangle$ state, the function is constant, and if the qubit is in the $|1\rangle$ state, the function is balanced. In this way, we only need to make one query to the “black box” in order to determine whether the function is constant or balanced, a clear improvement over the two queries we would need using classical computation!

Chapter 4

Shor's Algorithm

There are two main paradigms for quantum algorithms. The first, used in Shor's Algorithm, is to run the qubits through some sequence of gates, and then to perform a Fourier transform on the resulting state. We will encounter the second paradigm, an iterative approach, in the next chapter when we discuss Grover's Algorithm.

Classically, the factoring problem is believed not to be in P, although it has not been proven. In addition, factoring is not known to be NP-hard.[1] In fact, the well-known RSA cryptosystem relies on the assumption that factoring large numbers is hard. Using quantum effects, Shor's Algorithm can factor numbers in time polynomial in the length of the input, an exponential speedup over what is currently possible classically. In other words, if we can build a large enough quantum computer, we can break RSA.

4.1 Reducing Factoring to Order-Finding

The factoring question is quite simple. Given $N = pq$, with p and q primes, we ask to find p and q. Shor's algorithm works by reducing this problem to order-finding as follows.

First, we can use Euclid's Algorithm to check if the greatest common divisor (gcd) of two numbers is 1. Euclid's Algorithm is a classical algorithm that divides N by x, then divides x by the

[1]Recall that a problem is in the class P if it can be solved in polynomial time. A problem is in the class NP if there exists a polynomial-time verifier for it. A problem is NP-complete if it is in NP and all other problems in NP reduce to it (in other words, the problem is NP-hard). For a review of these topics, see Sipser [27].

remainder. It continues by dividing each remainder by the next, and the resulting final remainder is the gcd of N and x.

If we pick x randomly, with $x < N$, and Euclid's Algorithm tells us that the gcd of x and N is not 1, then we know that the gcd is either p or q, so we can simply divide N by the gcd and we are done. However, if the gcd is 1, we find the smallest r such that $x^r \equiv 1 \pmod{N}$. r is called the *order* of x modulo N. It turns out that r is not useful to us if it's odd, so we continue randomly picking x values until we obtain an even r (or until we stumble upon an x for which the gcd is not 1). When r is even, we have:

$$(x^{r/2} - 1)(x^{r/2} + 1) \equiv 0 \pmod{N}$$

Then, taking the gcd of N and $(x^{r/2} - 1)$ and the gcd of N and $(x^{r/2} + 1)$ returns the two prime factors of N with high probability [25]. So, we just need to be able to find r efficiently.

4.2 The Quantum Fourier Transform

Because we are dealing with the mod operator when we do our order-finding, we are working with a periodic function. The periodicity of the function suggests that using a Fourier transform might be useful.[2] The classical discrete Fourier transform is

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi ijk/N}$$

where x_j, $y_k \in \mathbb{C}$ and N is the length of the vector containing x_j. However, we need to use the quantum version of the Fourier transform, which acts on a set of basis states rather than a vector of complex numbers:

$$|j\rangle \longrightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle$$

[2] Recall that in general the result of applying a Fourier transform is a representation of the same system with respect to a different basis. This is often used in signal analysis to analyze frequencies. For a review of the classical Fourier transform and fast Fourier transform (FFT), see Cormen, Leiserson, Rivest and Stein [7].

where $|j\rangle$ is a basis state. So the quantum Fourier transform (QFT) expresses the same quantum state of the system in terms of a different basis set. We can rewrite this [21]:

$$\begin{aligned}
\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{2\pi ijk/N}|k\rangle &= \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1} e^{2\pi ijk/2^n}|k\rangle \\
&= \frac{1}{2^{n/2}}\sum_{k_1=0}^{1}\dots\sum_{k_n=0}^{1} e^{2\pi ij\left(\sum_{l=1}^{n} k_l 2^{-l}\right)}|k_1\dots k_n\rangle \\
&= \frac{1}{2^{n/2}}\sum_{k_1=0}^{1}\dots\sum_{k_n=0}^{1}\bigotimes_{l=1}^{n} e^{2\pi ijk_l 2^{-l}}|k_l\rangle \\
&= \frac{1}{2^{n/2}}\bigotimes_{l=1}^{n}\left(\sum_{k_l=0}^{1} e^{2\pi ijk_l 2^{-l}}|k_l\rangle\right) \\
&= \frac{1}{2^{n/2}}\bigotimes_{l=1}^{n}\left(|0\rangle + e^{2\pi ij2^{-l}}|1\rangle\right)
\end{aligned}$$

where $N = 2^n$ because the number of basis states for a qubit system is always a power of 2. Writing out the whole tensor product gives:

$$|j\rangle \longrightarrow \frac{\left(|0\rangle + e^{2\pi i0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i0.j_{n-1}j_n}|1\rangle\right)\dots\left(|0\rangle + e^{2\pi i0.j_1j_2\dots j_n}|1\rangle\right)}{2^{n/2}}$$

Now it looks like we've arrived at an expression we could obtain by applying a sequence of quantum gates to an N-qubit system. We construct the unitary transformation

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

and create a QFT circuit using repeated applications of Hadamard gates and R_k gates. It is possible to construct the QFT circuit such that it runs in $\Theta(n^2)$ time, compared to $\Theta(n2^n)$ for the classical fast Fourier transform [21].

4.3 Applying the QFT to Factoring

The periodic nature of order-finding lets us use the QFT to factor numbers in polynomial time. Figure 4.1 shows the quantum circuit for Shor's Algorithm. The algorithm uses two $O(n)$-qubit registers, where n is the number of bits required to represent N. We let q be the power of two such

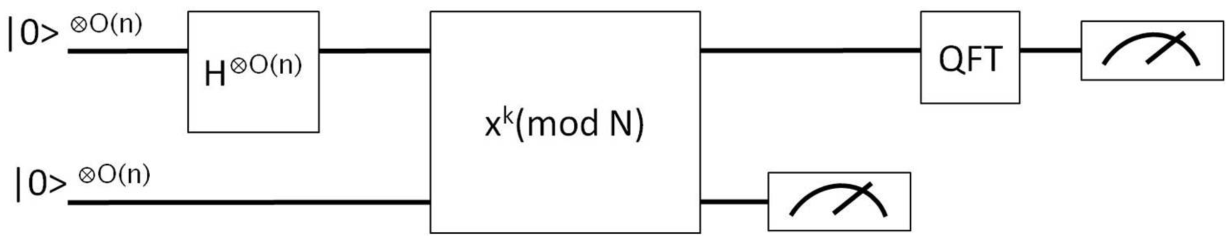

Figure 4.1: The circuit for Shor's Algorithm, an example of the first quantum algorithm paradigm, in which we manipulate qubits and then apply the QFT.

that $N^2 \leq q < 2N^2$, and apply an $O(n)$-qubit Hadamard gate to the first register, giving us the state

$$|\psi_0\rangle = \frac{1}{\sqrt{q}} \sum_{k=0}^{q-1} |k\rangle|0\rangle$$

because all the qubits in the second register are in state $|0\rangle$. Now we have the first register in an even superposition. We can create a unitary matrix that computes $x^k \pmod{N}$ and apply it to the second register, giving us the second state:

$$|\psi_1\rangle = \frac{1}{\sqrt{q}} \sum_{k=0}^{q-1} |k\rangle|x^k \pmod{N}\rangle$$

Next we apply the Quantum Fourier Transform to the first register, which leaves the second register unchanged and gives us the overall state

$$|\psi_2\rangle = \frac{1}{q} \sum_{k=0}^{q-1} \sum_{j=0}^{q-1} e^{2\pi kj/q}|j\rangle|x^k \pmod{N}\rangle$$

Now we're ready to measure the state of the system. Using some number theory that is beyond the scope of this paper, it turns out that the probability of finding the system in a particular state $|j, x^k \pmod{N}\rangle$ is at least $\frac{1}{3}r^2$ if there exists a such that[3]

$$\frac{-r}{2} \leq rj - aq \leq \frac{r}{2}$$

We divide by rq and rewrite this as

$$\left|\frac{j}{q} - \frac{a}{r}\right| \leq \frac{1}{2q}$$

[3]For a full derivation, see Shor [25] and Hardy [15].

where we know j and q. There is a unique value of a/r that satisfies the above equation and our requirement that $r < N$. It is possible to find this fraction in lowest terms in polynomial time using a continued fraction expansion. Then, if a is relatively prime to r, we are done. It turns out that this will be the case with probability at least $\frac{d}{\log\log r}$, for some constant d, so if we repeat Shor's Algorithm $O(\log\log r)$ times, we have a high probability of succeeding [5]. The entire algorithm runs in $O\left((\log N)^3\right)$ time, which is polynomial in the length of the number we wanted to factor! No known classical algorithm is capable of this.

Chapter 5

Grover's Algorithm

5.1 The Quantum Search Problem

Grover's Algorithm is a search algorithm which uses an iterative approach. By repeated applications of the algorithm, we can "rotate" the state of our quantum system closer and closer to a superposition of solutions to the search problem. Classically, we would have to visit each element in an unsorted list for our search to be successful, but Grover's Algorithm is able to search in $O(\sqrt{N})$ time.

The general search problem is given an unordered list of N elements, find an element that satisfies the given search criteria. We will assume that N is a power of 2. That is, $N = 2^n$, and we will deal primarily with the indices of the elements, which range from 0 to $N - 1$. For the sake of notation, we will say that the search problem has M solutions. That is, exactly M of the elements satisfy our search criteria. Grover's Algorithm makes use of an oracle that is a unitary transformation capable of recognizing solutions to the search problem.[1] In other words, the oracle is a gate O which takes a state $|x\rangle$ to the state $(-1)^{f(x)}|x\rangle$, where $f(x) = 0$ if x is not a solution to the search problem and $f(x) = 1$ if x is a solution. In this way, the phase of the quantum state will be shifted based on whether or not it is a solution to our search problem. How many times do we have to use this oracle in order to solve the search problem?

[1] Recall that a *recognizer* is able to answer in the affirmative if the input is a solution. This is different from being able to solve the problem in question.

5.2 The Iterative Routine

Grover's Algorithm requires one n-qubit register, with all n qubits initially in the $|0\rangle$ state. First, we put the register into an even superposition using some Hadamard gates:

$$|\psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

so now our system is in an even superposition of all possible values.

Next we send the system through a series of four steps, which we will combine and call the *Grover operator*. This will seem strange at first, but we will see that repeated applications of the Grover operator will help us solve the search problem. The first step in the Grover operator is to apply the oracle to our system. Second, we send the system through another set of Hadamard gates. Next, we perform a phase shift on every state except for the $|0\rangle$ state, such that

$$|x\rangle \longrightarrow -(-1)^{\delta_{x0}}|x\rangle$$

where δ_{ij} is the Dirac delta, defined $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ otherwise. The reader may object to this due to our earlier discussion of our ability to neglect overall phase factors. However, because we do not shift the $|0\rangle$ state, and our system clearly includes the $|0\rangle$ state in addition to many other states, this will never affect all the states in our superposition, and so is a relative phase factor, making it measurable and worthy of our attention.

This phase shift is accomplished by applying the unitary transformation $(2|0\rangle\langle 0| - \mathbb{I})$ to the system, where the $\langle|$ is called a *bra* and denotes the complex conjugate of a ket [21]. For example, the two-qubit form of this matrix is

$$\begin{aligned} 2|00\rangle\langle 00| - \mathbb{I} &= 2\begin{pmatrix}1\\0\\0\\0\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix} - \begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{pmatrix} \\ &= \begin{pmatrix}1 & 0 & 0 & 0\\0 & -1 & 0 & 0\\0 & 0 & -1 & 0\\0 & 0 & 0 & -1\end{pmatrix}\end{aligned}$$

Then, if the two-qubit system is in the $|0\rangle$ state, we get

$$\begin{aligned}
(2|00\rangle\langle 00| - \mathbb{I})|00\rangle &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\
&= |00\rangle
\end{aligned}$$

The $|00\rangle$ state remains unchanged, while any other state would pick up a relative phase factor of -1. This generalizes to an n-qubit system by simply using appropriately-sized $|0\rangle$ and $\mathbb{I}$ matrices.

The fourth, and final, step in the Grover operator is to apply yet another set of Hadamard gates. Because each of these Hadamard operations is used on n qubits, we write the gate as $H^{\otimes n}$. This means that we can write the Grover operator, G as

$$G = H^{\otimes n}(2|0\rangle\langle 0| - \mathbb{I})H^{\otimes n}O$$

Notice that the rightmost matrix, the oracle, gets applied first. In the circuit diagram, shown in Figure 5.1, the leftmost gate is applied first. At this point, we're done with Grover's Algorithm, which runs in $O(\sqrt{N})$ time [26]! Repeated applications of the Grover operator will, with high probability, put our system into a superposition of all solutions to the search problem so that when we take a measurement the system will project onto a solution state. In the next section we take a closer look at how this works.

5.3 Rotating the Quantum State

To see how Grover's Algorithm efficiently solves the search problem, it will be helpful to picture the state of the overall system as a vector in the space spanned by the two normalized basis states

$$\begin{aligned}
|\alpha\rangle &= \frac{1}{\sqrt{N-M}} \sum_x'' |x\rangle \\
|\beta\rangle &= \frac{1}{\sqrt{M}} \sum_x' |x\rangle
\end{aligned}$$

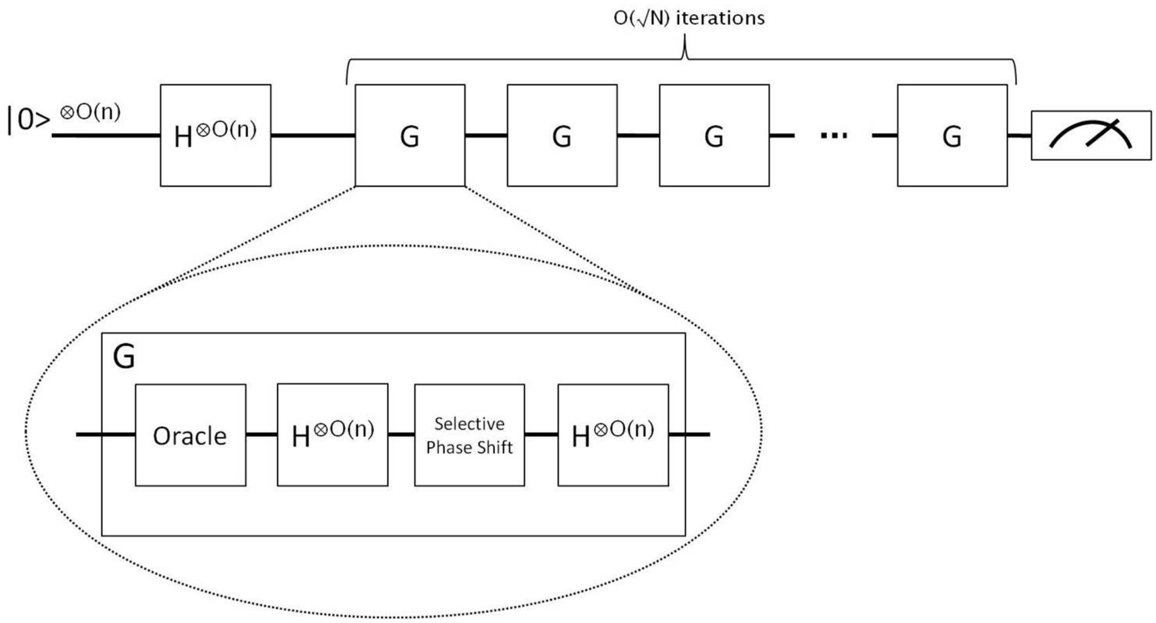

Figure 5.1: The circuit for Grover's Algorithm, an example of the second quantum algorithm paradigm, in which we apply the same operations multiple times.

where $\sum'_x$ indicates a sum over all x such that x is a solution to the search problem, and $\sum''_x$ is a sum over all x that are not solutions [21]. We re-write the state of the system (which is in an even superposition) in terms of this new basis:

$$\begin{aligned}
|\psi_1\rangle &= \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \\
&= \frac{1}{\sqrt{N}} \left(\sum_x{}'' |x\rangle + \sum_x{}' |x\rangle \right) \\
&= \frac{1}{\sqrt{N}} \left(\sqrt{N-M}|\alpha\rangle + \sqrt{M}|\beta\rangle \right) \\
&= \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle
\end{aligned}$$

$|\psi_1\rangle$ is the state we send into the first iteration of the Grover operator. Now let's consider what the Grover operator looks like in this basis. Applying the oracle is equivalent to a reflection across $|\alpha\rangle$, because the oracle will mark solutions to the search problem by negating the coefficient of $|\beta\rangle$.

The Hadamard, selective phase shift, Hadamard combination is equivalent to a reflection across $|\psi_1\rangle$ [21]. We know that applying two reflections gives us a rotation, so no matter how many times we apply the Grover operator, G, the state of our system remains within the space spanned by $|\alpha\rangle$ and $|\beta\rangle$. If we choose θ such that

$$\cos\left(\frac{\theta}{2}\right) = \sqrt{\frac{N-M}{N}}$$

we can rewrite $|\psi_1\rangle$ in our new basis as

$$|\psi_1\rangle = \cos\left(\frac{\theta}{2}\right)|\alpha\rangle + \sin\left(\frac{\theta}{2}\right)|\beta\rangle$$

Notice that we can do this re-write without any calculation, because we know that the squares of the coefficients must sum to 1. It can be shown that, in terms of θ and the new basis, G has the following effect on $|\psi_1\rangle$ [21]:

$$G|\psi_1\rangle = \cos\left(\frac{3\theta}{2}\right)|\alpha\rangle + \sin\left(\frac{3\theta}{2}\right)|\beta\rangle$$

This means that G rotates our $|\psi_1\rangle$ by the angle θ, because $|\psi_1\rangle$ starts out at an angle $\frac{\theta}{2}$ from $|\alpha\rangle$, and ends up at an angle $\frac{3\theta}{2}$ from $|\alpha\rangle$. The difference between these two angles is θ, so this is the resulting angle through which $|\psi_1\rangle$ is rotated by the Grover operator.

We have defined our new basis such that $|\beta\rangle$ only includes solutions to the search problem, so our goal is to manipulate the state of our system so that it ends up equivalent to $|\beta\rangle$. Then, when we take a measurement we will project onto a solution. Notice that after one application of the Grover operator we have moved the state of the system farther away from $|\alpha\rangle$ and closer to $|\beta\rangle$. What happens if we send the system through a second Grover operator? We get

$$\begin{aligned} G^2|\psi_1\rangle &= GG|\psi_1\rangle = G\left(\cos\left(\frac{3\theta}{2}\right)|\alpha\rangle + \sin\left(\frac{3\theta}{2}\right)|\beta\rangle\right) \\ &= \cos\left(\frac{5\theta}{2}\right)|\alpha\rangle + \sin\left(\frac{5\theta}{2}\right)|\beta\rangle \end{aligned}$$

Now the overall state of the system is even father away from $|\alpha\rangle$ and moving closer to $|\beta\rangle$! In general, if we send our system through k Grover operators, we end up with [21]

$$G^k|\psi_1\rangle = \cos\left(\frac{2k+1}{2}\theta\right)|\alpha\rangle + \sin\left(\frac{2k+1}{2}\theta\right)|\beta\rangle$$

So as we apply more Grover operators the state of the system continues to move closer to $|\beta\rangle$. We don't have to worry about overshooting our goal, because we already know that the coefficient of $|\beta\rangle$ in our starting state $|\psi_1\rangle$ is $\sqrt{M/N}$, which means that our system state starts out $\phi = \arccos\left(\sqrt{M/N}\right)$ radians away from $|\beta\rangle$. All we need to do is apply enough Grover operators so that we rotate the state of the system through ϕ radians. The number of repetitions required to do this is the closest integer to

$$R = \frac{\arccos\left(\sqrt{M/N}\right)}{\theta}$$

in order to have a measurement return a solution to the search problem with approximately 50% probability [14].

In most instances of the search problem, we have the case where $M \ll N$, which lets us use a small angle approximation ($\theta \approx \sin\theta$) to arrive at $R \in O\left(\sqrt{N/M}\right)$. This is a quadratic, but not exponential, improvement over the $O(N/M)$ queries to the oracle that would be required classically.

Chapter 6

Quantum Teleportation

6.1 The Communication Question

Quantum effects also give us new methods of communication. Suppose Alice has a qubit in some unknown quantum state that she wants to send to Bob. We already know that it is impossible to copy a qubit, so Alice can't just clone her qubit and send the clone to Bob. She also can't measure the qubit and try to deduce how to recreate it because she won't be able to figure out if it was in a superposition. She could simply send the physical qubit to Bob, but that would be boring, and she might want to use the qubit for something else later. Using an entangled pair of qubits and two classical bits in addition to the qubit she wants to send, Alice will be able to communicate an entire continuous quantum state to Bob while only sending him extra two bits of classical information [23].

6.2 The Teleportation Protocol

The teleportation circuit is shown in Figure 6.1. First, Alice and Bob each start with half of an entangled pair of qubits, in the β_{00} Bell state, so the overall state of the system is

$$\begin{aligned}
|\psi_0\rangle &= |\psi\rangle|\beta_{00}\rangle \\
&= (\alpha|0\rangle + \beta|1\rangle)\left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right) \\
&= \frac{\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle)}{\sqrt{2}}
\end{aligned}$$

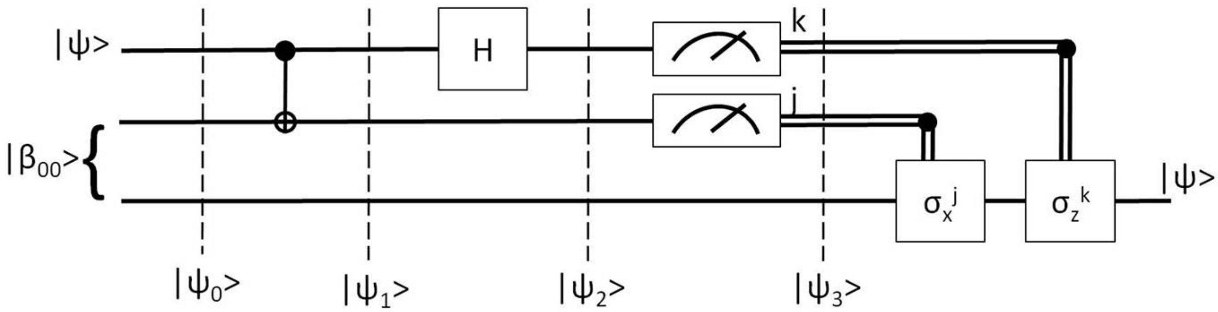

Figure 6.1: The quantum teleportation circuit, with each state of the system labelled. The top two qubits belong to Alice, and the bottom qubit belongs to Bob. The first operation is a CNOT gate, with the top qubit as the control. Which gates Bob applies to his qubit are dependent on the outcome of Alice's measurements (which are classical bits). The circuit successfully transmits the original unknown qubit state to Bob, without physically sending him any qubits.

where the unknown state Alice wants to send is $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, and the first two qubits belong to Alice while the third qubit belongs to Bob. Then, Alice applies a CNOT gate to her qubits, using the unknown state as the control qubit to get

$$|\psi_1\rangle = \frac{\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle)}{\sqrt{2}} \tag{6.1}$$

The final trick is to have Alice send the unknown state through a Hadamard gate, and do some clever re-grouping of terms, so that we have

$$\begin{aligned}
|\psi_2\rangle &= [\alpha(|0\rangle + |1\rangle) \otimes (|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle) \otimes (|10\rangle + |01\rangle)]/2 \\
&= [\alpha(|0\rangle + |1\rangle)|00\rangle + \alpha(|0\rangle + |1\rangle)|11\rangle + \beta(|0\rangle - |1\rangle)|10\rangle + \beta(|0\rangle - |1\rangle)|01\rangle]/2 \\
&= [\alpha|000\rangle + \alpha|100\rangle + \alpha|011\rangle + \alpha|111\rangle + \beta|010\rangle - \beta|110\rangle + \beta|001\rangle - \beta|101\rangle]/2 \\
&= [|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\beta|0\rangle + \alpha|1\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(-\beta|0\rangle + \alpha|1\rangle)]/2
\end{aligned}$$

Notice that one of the terms includes the original unknown qubit state, and recall the Pauli spin matrices from our discussion of single-qubit gates. If we rewrite $|\psi_2\rangle$ in terms of the quantum state Alice is trying to send and the Pauli spin matrices, we get

$$|\psi_2\rangle = [|00\rangle|\psi\rangle + |01\rangle\sigma_x|\psi\rangle + |10\rangle\sigma_z|\psi\rangle + |11\rangle\sigma_x\sigma_z|\psi\rangle]/2$$

It's time for Alice to measure both of her qubits. Because Alice and Bob share an entangled pair, if she gets a $|00\rangle$ result, then she knows that Bob's qubit is in the state $|\psi\rangle$. Similarly, if she gets a $|01\rangle$ result, she knows that Bob's qubit is in the state $\sigma_x|\psi\rangle$, and likewise for the other two possible measurement results. Alice only needs two classical bits to store the result of her measurements, and she sends her results to Bob in the normal (non-quantum) way. Note that because Alice must send her measurement results to Bob in order for this routine to work, we still aren't able to communicate information faster than the speed of light.

Bob knows the procedures that Alice is using, so when he receives her measurement results he has all the information he needs in order to obtain the unknown quantum state Alice is trying to send him. If Alice sends him "00," he knows that his qubit is already in the state $|\psi\rangle$, and we're done. However, if Alice sends Bob "01," he knows that his qubit is the negation of $|\psi\rangle$, so he simply applies a σ_x gate to his qubit and retrieves the original unknown state. Similarly, if Bob receives "10," he applies a σ_z gate, and for the "11" case Bob must apply both a σ_z gate and a σ_x gate. In all cases, Bob is able to perform operations on his half of the entangled pair such that he gets back the original state that Alice was trying to communicate!

6.3 Observations

The reader may be inclined to complain that calling this communication routine "teleportation" promises much more than it actually delivers, but consider again what Alice and Bob have achieved here. Quantum states take on continuous values, and require vastly more than two classical bits to completely specify. Yet, using one entangled pair and two classical bits, Alice and Bob are able to send each other entire quantum states without actually physically sending the state between them. This implies that one qubit is, in some sense, equivalent to an entangled pair and two classical bits.

This method of communication using quantum teleportation has been tested experimentally across distances of up to 16km [16], and will theoretically work over any distance.

Chapter 7

Cryptography: The Quantum One-Time Pad

7.1 The Classical Protocol

The most secure classical cryptosystem is the one-time pad. In this protocol, Alice and Bob share a key, a random string of bits at least as long as the plaintext. Alice XORs (adds modulo 2) the plaintext with the key, and sends the result to Bob. Because the key is random, no information can be deduced by Eve, the eavesdropper, as long as the key is only used once (hence the name "one-time pad"). When Bob receives the ciphertext from Alice, he XORs it with his copy of the key and gets back the plaintext. However, if Alice and Bob use the same key more than once, Eve will be able to gain information about the messages they send and eventually break the system. The difficulty lies in generating a different key each time Alice and Bob want to communicate, and in sending the key itself between Alice and Bob such that they both know the key, but Eve does not. Quantum computing provides a clever method for distributing and generating keys for a one-time pad protocol.

7.2 Two Basis Sets

Up until now we have imagined qubits as being measured in only one basis (the $|0\rangle$, $|1\rangle$ basis), most likely implemented using electron spin. For the purposes of the Bennett-Brassard (BB84) Protocol, which provides a way to generate and distribute keys for a one-time pad, we will measure our qubits using two different bases. The first is along the z-axis, and consists of the $|\uparrow\rangle$ and $|\downarrow\rangle$ states. The second basis, along the x-axis, consists of the states $|\leftarrow\rangle$ and $|\rightarrow\rangle$. Which basis set the system projects onto is determined by our choice of measurement axis. For example, if we prepare a qubit to be in the $|\uparrow\rangle$ state it will be expressed in the x-axis basis as $\frac{|\leftarrow\rangle+|\rightarrow\rangle}{\sqrt{2}}$. There is an equal probability of finding the qubit in the $|\leftarrow\rangle$ or $|\rightarrow\rangle$ state. We get a similar result if we prepare the qubit in the x-axis basis and then measure it along the z-axis [17].

The behavior of a qubit when we use two different bases is a result of the *uncertainty principle*, which states that we cannot simultaneously know the exact result of measurements along both axes. That is, measuring along one axis does not affect the projection we get if we then measure along another axis. In fact, if we were to measure along the z-axis and get $|\uparrow\rangle$, follow it with an x-axis measurement that happened to give us $|\leftarrow\rangle$, and then measure along the z-axis again, there would be no guarantee that we would project onto $|\uparrow\rangle$ again. If we know the result of a measurement along one axis, we do not know anything about what we would get if we measured along the other axis (this has been demonstrated in the well-known Stern-Gerlach Experiment).

7.3 The Bennett-Brassard (BB84) Protocol

In the BB84 protocol [2], Alice sends Bob a sequence of qubits, randomly choosing either the x- or z-axis for each qubit and taking a measurement before sending each one. Bob also randomly chooses an axis for each qubit and takes his measurements. Note that Bob does not know which axes Alice chose when she performed her measurements. Eve also does not know which axes Alice chose, so if Eve has been intercepting and measuring qubits, Alice and Bob will be able to detect her by using the following procedure [24].

Alice uses a normal, non-quantum and possibly not secure, communication channel to send Bob the result of some of the measurements that she made. Because Alice and Bob independently chose a random axis for each qubit, they will have picked the same axis half the time, on average. Bob verifies that his measurements match Alice's results. If his measurements do not match with the expected frequency, Alice and Bob know that Eve has been in the middle measuring their qubits.

There is no way for Eve to disguise her presence. It is possible for Alice and Bob to estimate how much information Eve has intercepted based on how many qubits have been corrupted.

Then, Alice tells Bob which axis she chose for each of her qubits. Bob will have chosen the same axis on approximately half of these qubits, and if there is no eavesdropper present, they can use these matching measurements as a key for a one-time pad. However, if Eve has corrupted some of their qubits, Alice and Bob can determine how many of their qubits Eve knows, and use classical privacy amplification methods to turn their shared string of qubits in to a more secure shorter random string [17]. Then, Alice and Bob can use this shorter string as their key.

Using BB84, Alice and Bob are always able to detect Eve's presence, and they can securely construct multiple random keys for the classical one-time pad protocol.

Chapter 8

Conclusion

8.1 What We Know

Quantum computation pushes the limits of computer science, physics, and engineering. In theory, it can improve runtimes compared to what is currently known classically, especially with regard to factoring and database search problems. Quantum techniques can also improve the amount of information we are able to communicate and its security.

The first proof of concept for Shor's Algorithm was achieved in 2001 when Vandersypen et al. factored 15 into 3 and 5 using a 7-qubit nuclear magnetic resonance quantum computer [29]. A portion of the scientific community questioned the functioning of this machine because no explicit entanglement had been detected. However, Politi et al. successfully factored 15 using a quantum computer built using a photonic chip and conclusively demonstrated the presence of entanglement [22]. The Deutsch-Jozsa Algorithm and Grover's Algorithm have also been successfully implemented, although on a similarly small scale [28],[3]. The current world record for largest implemented quantum computer is 14 qubits [20].

8.2 What We Don't Know

There are various schemes for implementing a quantum computer, including nuclear magnetic resonance, ion traps, and photonic chips. The interested reader with a strong physics background should consult Nielsen and Chuang [21]. Building a quantum computer poses a significant engineering challenge. Quantum states are extremely fragile and eventually fall apart, or *decohere*, even when left alone. The problem gets worse when the quantum states are able to interact with the surrounding environment — decoherence occurs much more quickly. How do you build a computer that is sufficiently isolated from its environment? The answer is not clear.

In addition, decoherence creates major hurdles for the development of quantum memory elements. Long-term storage of qubits is an extremely difficult problem and an active area of research. Some architectures for quantum memories have been proposed [12], but none have been shown to be sufficiently robust or scalable. In fact, none of the physical designs for quantum computers have been proven to be scalable to very powerful sizes (big enough to run Shor's Algorithm on numbers large enough to break RSA, for example). However, there is also no proof that building a scalable quantum computer is impossible.

We also don't know if quantum computing will ever provide us with speedups other than those already discovered. Quantum mechanics is so counterintuitive that it becomes surprisingly hard to design algorithms. Since 1996, there are still only two main approaches to designing a quantum algorithm: the QFT, or iterations. Many questions also remain open in the field of quantum complexity theory, including whether or not quantum computers can solve NP-complete problems. Many researchers believe that not even quantum computers can solve hard problems in polynomial time [1], but it has not been proven. The fact that Shor's Algorithm can factor in polynomial time suggests that quantum computers might be fundamentally more powerful than classical computers, but no one is certain about that either.

Even if we never manage to build quantum computers that use 100 or more qubits, computations on smaller machines are interesting to physicists. Smaller quantum systems provide a wealth of open questions, and strategies from quantum computing are helping to guide the pursuit of some answers. Quantum computers can be used to accurately simulate interesting quantum systems faster than a classical computer could [30], and some researchers even propose that quantum teleportation circuits could be used to simulate and investigate differing theories of time travel [19].

Bibliography

[1] Scott Aaronson. The limits of quantum. *Scientific American*, pages 62 – 69, March 2008.

[2] C.H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. *Proceedings of IEEE International Conference on Computers Systems and Signal Processing*, pages 175 – 179, 1984.

[3] K.-A. Brickman, P.C. Haljan, P.J. Lee, M. Acton, L. Deslauriers, and C. Monroe. Implementation of Grover's quantum search algorithm in a scalable system. *Physical Review A*, 72, 2005. arXiv:quant-ph/0510066v2.

[4] Isaac Chuang. Quantum algorithms and their implementations. Class notes for QuISU: An Introduction for Undergraduates, 2009.

[5] Jill Cirasella. An introduction to quantum computing. Amherst College Computer Science Senior Thesis, 1998.

[6] Richard Cleve. An introduction to quantum complexity theory. To appear in Collected Papers on Quantum Computation and Quantum Information Theory, edited by C. Macchiavello, G.M. Palma, and A. Zeilinger (World Scientific), 1999. arXiv:quant-ph/9906111v1.

[7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, third edition, 2009.

[8] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings: Mathematical and Physical Sciences*, 439(1907):553 – 558, 1992.

[9] Richard Feynman, Robert Leighton, and Matthew Sands. *The Feynman Lectures on Physics, Volume 3*. Addison Wesley, New York, New York, 2006.

[10] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467 – 488, 1982.

[11] Jonathan Friedman. Quantum information, quantum measurement and quantum computing. Transcipts of lectures from Amherst College PHYS-76 class, 2004.

[12] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quantum random access memory. *Physical Review A*, 78, 2008. arXiv:0807.4994v2[quant-ph].

[13] Ian Glendinning. The Bloch sphere. Slides from presentation at the European Centre for Parallel Computing in Vienna, 2005.

[14] Lov K. Grover. A fast quantum mechanical algorithm for database search. *Proceedings, 28th Annual ACM Symposium on the Theory of Computing*, 1996. arXiv:quant-ph/9605043.

[15] G.H. Hardy and E.M. Wright. *An Introduction to the Theory of Numbers*. Oxford Science Publications, New York, New York, fifth edition, 1984.

[16] Xian-Min Jin, Ji-Gang Ren, Bin Yang, Zhen-Huan Yi, Fei Zhou, Xiao-Fan Xu, Shao-Kai Wang, Dong Yang, Yuan-Feng Hu, Shuo Jiang, Tao Yang, Hao Yin, Kai Chen, Cheng-Zhi Peng, and Jian-Wei Pan. Experimental free-space quantum teleportation. *Nature Photonics*, 4:376 – 381, 2010.

[17] Seth Lloyd. Quantum information science. Supplementary reading for QuISU, 2010.

[18] Seth Lloyd. Quantum mechanics of two-level systems. Presented at the QuISU summer program, 2010.

[19] Seth Lloyd, Lorenzo Maccone, Raul Garcia-Patron, Vittorio Giovannetti, Yutaka Shikano, Stefano Pirandola, Lee A. Rozema, Ardavan Darabi, Yasaman Soudagar, Lynden K. Shalm,

and Aephraim M. Steinberg. Closed timelike curves via post-selection: theory and experimental demonstration. *Phys. Rev. Lett.*, 106, 2011. arXiv:1005.2219v1[quant-ph].

[20] Thomas Monz, Philipp Schindler, Julio T. Barreiro, Michael Chwalla, Daniel Nigg, William A. Coish, Maximillian Harlander, Wolfgang Hansel, Markus Hennrich, and Rainer Blatt. 14-qubit entanglement: Creation and coherence. *Phys. Rev. Lett.*, 106, 2011.

[21] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, New York, New York, 2000.

[22] Alberto Politi, Jonathan C.F. Matthews, and Jeremy L. O'Brien. Shor's quantum factoring algorithm on a photonic chip. *Science*, 325:1221, 2009.

[23] Jeffrey H. Shapiro. Entanglement and teleportation. Supplementary reading for QuISU, 2008.

[24] Jeffrey H. Shapiro. Quantum key distribution and other applications. Supplementary reading for QuISU, 2008.

[25] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, 1994. arXiv:quant-ph/9508027v2.

[26] Peter W. Shor. Introduction to quantum algorithms. Notes for talk given at the January 2000 American Math Society Meeting, 2000. arXiv:quant-ph/0005003.

[27] Michael Sipser. *Introduction to the Theory of Computation.* Course Technology, Boston, Massachusetts, second edition, 2006.

[28] Giuseppe Vallone, Gaia Donati, Natalia Bruno, Andrea Chiuri, and Paolo Mataloni. Experimental realization of the Deutsch-Jozsa algorithm with a six-qubit cluster state. *Physical Review A*, 81, 2010. arXiv:1003.4607v1[quant-ph].

[29] Lieven M.K. Vandersypen, Matthia Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, and Isaac L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883 – 887, 2001.

[30] Christof Zalka. Simulating quantum systems on a quantum computer. *Proc. R. Soc. Lond. A*, 454:313 – 322, 1998.