An Introduction to Quantum Computing for Non-Physicists

Eleanor Rieffel
rieffel@pal.xerox.com

Wolfgang Polak
polak@pal.xerox.com

August 14, 1998

1 Introduction

Richard Feynman observed in the early 1980's [1] that certain quantum mechanical effects cannot be simulated efficiently on a classical computer. This observation led to speculation that perhaps computation in general could be done more efficiently if it made use of these quantum effects. But building quantum computers, computational machines that use such quantum mechanical effects, proved tricky, and as no one was sure how to use the quantum effects to speed up computation, the field developed slowly. It wasn't until 1994, when Peter Shor surprised the world by describing a polynomial time quantum algorithm for factoring integers [2, 3], that the field of quantum computing came into its own. This discovery prompted a flurry of activity, both among experimentalists trying to build quantum computers and theoreticians trying to find other quantum algorithms. Additional interest in the subject has been created by the invention of quantum key distribution and, more recently, popular press accounts of experimental successes in quantum teleportation and the demonstration of a two-bit quantum computer.

The aim of this paper is to guide computer scientists and other non-physicists through the conceptual and notational barriers that separate quantum computing from conventional computing and to acquaint them with this new and exciting field. It is important for the computer science community to understand these new developments since they may radically change the way we have to think about computation, programming, and complexity.

The power of quantum computation comes from quantum parallelism. Classically, the time it takes to do certain computations can be decreased by using parallel processors. To achieve an exponential decrease in time requires an exponential increase in the number of processors, and hence an exponential increase in the amount of physical space needed. However, in quantum systems the amount of parallelism increases exponentially with the size of the system.

Thus, an exponential increase in parallelism requires a linear increase in the amount of physical space needed.

There is a catch, and a big catch at that. While a quantum system can perform massive parallel computation, access to the results of the computation is restricted. Accessing the results is equivalent to making a measurement, which disturbs the quantum state. This problem makes the situation, on the face of it, seem even worse than the classical situation; we can only read the result of one parallel thread, and because measurement is probabilistic, we cannot even choose which one we get.

But in the past few years, various people have found clever ways of getting useful information out of quantum parallelism. One technique is to determine a common property of all of the output values such as the symmetry or period of a function. This is the technique used in Shor's factorization algorithm. Another technique is to transform the state to increase the likelihood that the output of interest will be read. Grover's search algorithm makes use of such an amplification technique. This paper describes the details of quantum parallelism, and the techniques currently known for harnessing its power.

The section following this introduction explains some of the basic concepts of quantum mechanics that are important for quantum computation. This section does not attempt to give a comprehensive view of quantum mechanics as this would be beyond the scope of this paper. Our aim is to provide the reader with tools in the form of mathematics and notation with which to work with quantum mechanics. We hope that this paper will equip computer scientists, and other non-physicists, well enough that they can freely explore the theoretical realm of quantum computing.

Section 3 defines the quantum bit, or qubit. Unlike classical bits, a quantum bit can be put in a superposition state that encodes both 0 and 1. There is no good classical explanation of superpositions: a quantum bit representing 0 and 1 can neither be viewed as "between" 0 and 1 nor can it be viewed as a hidden unknown state that represents either 0 or 1 with a certain probability. Even single quantum bits enable interesting applications. We show the use of a single quantum bit for secure key distribution.

But the real power of quantum computation derives from the exponential state spaces of multiple quantum bits: just as a single qubit can be in a superposition of 0 and 1, a register of n qubits can be in a superposition of all 2^n possible values. The famous EPR[1] paradox (see section 3.4) is a result of *entangled* states that form a part of the quantum state space that do not exist in classical systems.

We will discuss the two types of operations a quantum system can undergo: measurement and quantum state transformations. Most quantum algorithms involve a sequence of quantum transformations followed by a measurement. For classical computers there are sets of gates that are universal in the sense that

[1]EPR = Einstein, Podolsky and Rosen

any classical computation can be performed using a sequence of these gates. Similarly, there are sets of primitive quantum transformations, called quantum gates, that are universal for quantum computation. Given enough quantum bits, it is possible to construct a universal quantum Turing machine.

Quantum physics puts restrictions on the types of transformations that can be done. In particular, all quantum transformations, and therefore all quantum gates and all quantum computations, must be reversible. Yet all classical algorithm can be computed on a quantum computer, i.e., in a manner that makes them reversible. Some common quantum gates are defined in section 4.

Two applications combining quantum gates and entangled states are described in section 4.2: teleportation and dense coding. Teleportation is the transfer of quantum state from one place to another through classical channels. That teleportation is possible is surprising since quantum mechanics tells us that it is not possible to clone quantum states or even measure them without disturbing the state. Thus, it is unclear what information could be sent through classical channels that could possibly enable the reconstruction of an unknown quantum state at the other end. Dense coding, a dual to teleportation, uses a single quantum bit to transmit two bits of classical information. Both teleportation and dense coding rely on the entangled states described in the EPR experiment.

It is only in section 5 that we see where an exponential speed up over classical computers might come from. The input to a quantum computation can be put in a superposition state that encodes all possible input values. Performing the computation on this initial state will result in superposition of all corresponding output values. Thus, in the same time it takes to compute the output for a single input state on a classical computer, a quantum computer can compute the values for all input states. This process is known as quantum parallelism. However, measuring the output states will randomly yield only one of the values in the superposition, and at the same time destroy all of the other results of the computation. Section 5 describes this situation in detail.

Section 6 describes the details of Shor's polynomial time factoring algorithm. The fastest known classical factoring algorithm requires exponential time and it is generally believed that there are no classical polynomial time factoring algorithms. Shor's is a beautiful algorithm that takes advantage of quantum parallelism by using a quantum analog of the Fourier transform.

Lov Grover developed a technique for searching an unstructured list of n items in $O(\sqrt{n})$ steps on a quantum computer. Classical computers can do no better than $O(n/2)$, so unstructured search on a quantum computer is provably more efficient than search on a classical computer. However, the speed-up is only polynomial, not exponential, and it has been shown that Grover's algorithm is optimal for quantum computers. It seems likely that search algorithms that could take advantage of some problem structure could do better. Tad Hogg, among others, has explored such possibilities. We describe various quantum search techniques in section 7.

It is as yet unknown whether the power of quantum parallelism can be harnessed for a wide variety of applications. One tantalizing open question is whether quantum computers can solve NP complete problems in polynomial time.

Perhaps the biggest open question is whether useful quantum computers can be built. There are a number of proposals for building quantum computers most of which are based either on ion traps or on nuclear magnetic resonance (NMR) technology.

In an ion trap quantum computer [4, 5] a linear sequence of ions (qubits) are confined by electric fields. Lasers are directed at individual atoms to perform single bit quantum gates. Two-bit operations are realized by using a laser on one qubit to create an impulse that ripples through a chain of ions to the second qubit where another laser pulse stops the rippling and performs the two-bit operation. The ion trap approach requires extreme vacuum and extremely low temperatures.

The NMR approach has the major advantage that it will work at room temperature. The idea is to use macroscopic amounts of matter and encode a quantum bit in the average spin state of a large number of nuclei. The spin states can be manipulated by magnetic fields and the average spin state can be measured with NMR techniques. The main problem with the technique is that it doesn't scale well; the measured signal scales as $1/2^n$ with the number of qubits n. However, a recent proposal ([6]) has been made that may overcome this problem. NMR computers with two qubits have been built successfully [7, 8]. This paper will not discuss further the physical and engineering problems of building quantum computers.

The greatest problem for building quantum computers is decoherence, the distortion of the quantum state due to interaction with the environment. For some time it was feared that quantum computers could not be built because it would be impossible to isolate them sufficiently from the external environment. The breakthrough came from the algorithmic rather than the physical side, through the invention of quantum error correction techniques. It is possible to design error correcting codes that tolerate certain kinds of errors and allow reconstruction of the exact error-free quantum state. Quantum error correction is discussed in section 8.

Appendices provide the necessary background information on tensor products and continued fractions.

Acknowledgements

The authors would like to thank Tad Hogg and Carlos Mochon for many enjoyable conversations about quantum computing, and for their feedback on an earlier draft of this paper. We are also grateful to for the detailed comments we received on an earlier draft from Lee Corbin, David Goldberg, and Norman

Hardy. Finally, we would like to thank FXPAL for enthusiastically supporting this work.

2 Quantum Mechanics

Quantum phenomena are difficult to understand since most of our everyday experiences are not applicable. This paper cannot provide a deep understanding of quantum mechanics (see [9], [10], or [11] for details). Instead, we will give some feeling as to the nature of quantum mechanics and some of the mathematical formalisms needed to work with quantum mechanics to the extent needed for quantum computing.

Quantum mechanics is a theory in the mathematical sense: it is governed by a set axioms. The consequences of the axioms describe the behavior of quantum systems. The axioms lead to several apparent paradoxes: in the Compton effect it appears as if an action precedes its cause; the EPR experiment makes it appear as if action over a distance faster than the speed of light is possible. We will discuss the EPR experiment in detail in section 3.4. Verification of most predictions is indirect, and requires careful experimental design and specialized equipment. We will begin, however, with an experiment that requires only readily available equipment and that will illustrate some of the key aspects of quantum mechanics.

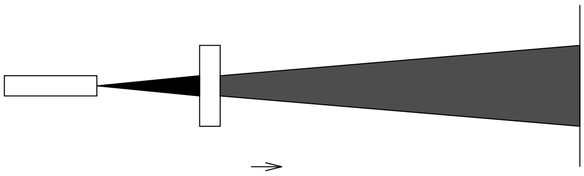
2.1 Photon Polarization

Photons are the only particles that we can directly observe. The following simple experiment can be performed with minimal equipment: a laser pointer (or other strong light source) and three polaroids (polarization filters) that can be picked up at any camera supply store. The experiment demonstrates some of the principles of quantum mechanics through photons and their polarization.

2.1.1 The Experiment

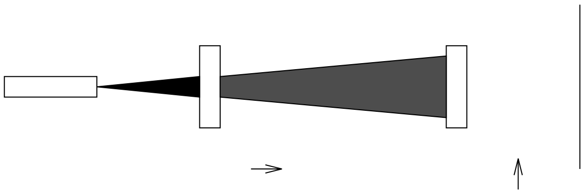
A beam of light shines on a projection screen. Filters A, B, and C are polarized horizontally, at 45^o, and vertically, respectively, and can be placed so as to intersect the beam of light.

First, insert filter A. Assuming the incoming light is randomly polarized, the intensity of the output will have half of the intensity of the incoming light. The outgoing photons are now all horizontally polarized.

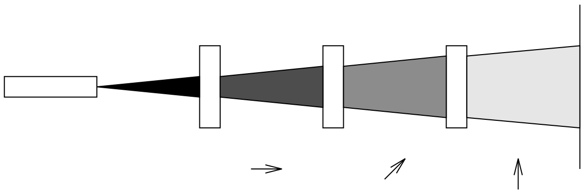


The function of filter A cannot be explained as a "sieve" that only lets those photons pass that happen to be already horizontally polarized. If that were the case, few of the randomly polarized incoming electrons would be horizontally polarized, so we would expect a much larger attenuation of the light as it passes through the filter.

Next, when filter C is inserted the intensity of the output drops to zero. None of the horizontally polarized photons can pass through the vertical filter. A sieve model could explain this behavior.



Finally, after filter B is inserted between A and C, a small amount of light will be visible on the screen, exactly one eighth of the original amount of light.



Here we have another nonintuitive effect. Classical experience suggests that adding a filter should only be able to decrease the amount of light getting through. How can it increase it?

2.1.2 The Explanation

A photon's polarization state can be modelled by a unit vector pointing in the appropriate direction. Any arbitrary polarization can be expressed as a linear combination $a|\uparrow\rangle + b|\rightarrow\rangle$ of the two basis vectors[2] $|\rightarrow\rangle$ (horizontal polarization) and $|\uparrow\rangle$ (vertical polarization).

Since we are only interested in the direction of the polarization (the notion of "magnitude" is not meaningful), the state vector will be a unit vector, i.e., $|a|^2 + |b|^2 = 1$. In general, the polarization of a photon can be expressed as $a|\uparrow\rangle + b|\rightarrow\rangle$ where a and b are complex numbers[3] such that $|a|^2 + |b|^2 = 1$. Note, the choice of orthonormal basis is completely arbitrary: any two orthogonal unit vectors will do (e.g. $\{|\nwarrow\rangle, |\nearrow\rangle\}$).

[2] The notation $|\rightarrow\rangle$ is explained in section 2.2.

[3] Imaginary coefficients correspond to circular polarization.

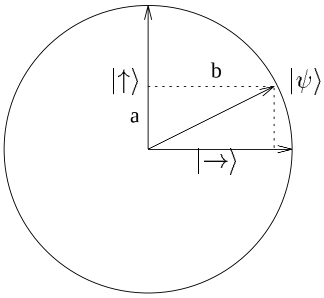


Figure 1: Measurement is a projection onto the basis

The measurement postulate of quantum mechanics states that each measurement has an associated orthonormal basis with respect to which the measurement projects the quantum state. For example, the probability that $\psi = a|\uparrow\rangle + b|\rightarrow\rangle$ is measured as $|\uparrow\rangle$ is $|a|^2$ and the probability that ψ is measured as $|\rightarrow\rangle$ is $|b|^2$ (see Figure 1). As measurements are always made with respect to an orthonormal basis, throughout the rest of this paper all bases will be assumed to be orthonormal. Note that different measuring devices have different associated bases.

Furthermore, measurement of the quantum state will change the state to the result of the measurement. That is, if measurement of $\psi = a|\uparrow\rangle + b|\rightarrow\rangle$ results in $|\uparrow\rangle$, then the state ψ changes to $|\uparrow\rangle$ and if the state is measured again with respect to the same basis will return $|\uparrow\rangle$ with probability 1. Thus, unless the original state happened to be one of the basis vectors, measurement will change that state, and it is not possible to know what the original state was.

Quantum mechanics can explain the polarization experiment as follows. A polaroid measures the quantum state of photons with respect to the basis consisting of the vector corresponding to its polarization and a vector orthogonal to its polarization. polarization. Photons pass through the filter only if the measurement of their state returns the given polarization. The photons which, after being measured by the filter, match the filter's polarization are let through. The others are reflected and now all have a polarization perpendicular to that of the filter. For example, filter A measures the photon polarization with respect to the basis vector $|\rightarrow\rangle$, corresponding to its polarization. The photons that pass through filter A all have polarization $|\rightarrow\rangle$. Those that are reflected by the filter all have polarization $|\uparrow\rangle$.

Assuming that the light source produces photons with random polarization, filter A will measure 50% of a all photons as horizontally polarized. These photons will pass through the filter and their state will be $|\rightarrow\rangle$. Filter C will measure these photons with respect to $|\uparrow\rangle$. But the state $|\rightarrow\rangle = 0|\uparrow\rangle + 1|\rightarrow\rangle$ will be projected onto $|\uparrow\rangle$ with probability 0 and no photons will pass filter C.

Finally, filter B measures the quantum state with respect to the basis

$$\{\frac{1}{\sqrt{2}}(|\uparrow\rangle + |\rightarrow\rangle), \frac{1}{\sqrt{2}}(|\uparrow\rangle - |\rightarrow\rangle)\}$$

which we write as $\{|\nearrow\rangle, |\nwarrow\rangle\}$. Those photons that are measured as $|\nearrow\rangle$ pass through the filter. Photons passing through A with state $|\rightarrow\rangle$ will be measured by B as $|\nearrow\rangle$ with probability $1/2$ and so 50% of the photons passing through A will pass through B and be in state $|\nearrow\rangle$. As before, these photons will be measured by filter C as $|\uparrow\rangle$ with probability $1/2$. Thus only one eighth of the original photons manage to pass through the sequence of filters A, B, and C.

The quantum state of a system, consisting of the positions, momentums, polarizations, spins, etc. of the various particles, evolves over time obeying Schrödinger's equation. The state space of a quantum system is modelled by a Hilbert space of wave functions. For quantum computing we need only deal with finite quantum systems and it suffices to consider the finite dimensional complex vector space with an inner product that is spanned by abstract wave functions such as $|\rightarrow\rangle$. In particular, bases for the state space consisting of mutually orthogonal vectors of unit length can be found.

2.2 State Spaces and Bra/Ket Notation

The quantum state of a system, consisting of the positions, momentums, polarizations, spins, etc. of the various particles, evolves over time obeying Schrödinger's equation. The state space of a quantum system is modelled by a Hilbert space of wave functions. For quantum computing we need only deal with finite quantum systems and it suffices to consider the finite dimensional complex vector space with an inner product that is spanned by abstract wave functions such as $|\rightarrow\rangle$. In particular, bases for the state space consisting of mutually orthogonal vectors of unit length can be found.

Quantum state spaces and the tranformations acting on them can be described in terms of vectors and matrices or in the more compact bra/ket notation. The bra/ket notation was invented by Dirac [12]. Kets like $|x\rangle$ denote column vectors and are typically used to describe quantum states. The matching bra, $\langle x|$, denotes the conjugate transpose of $|x\rangle$. For example, the orthonormal basis$\{(1,0)^T, (0,1)^T\}$ for a two dimensional complex vector space can be expressed as $\{|0\rangle, |1\rangle\}$. Any complex linear combination of $|0\rangle$ and $|1\rangle$, $a|0\rangle + b|1\rangle$, can be written $(a,b)^T$. Note that the choice of the order of the basis vectors is arbitrary. For example, representing $|0\rangle$ as $(0,1)^T$ and $|1\rangle$ as $(1,0)^T$ would be fine as long as this is done consistently.

Combining bra and ket as in $\langle x||y\rangle$, also written as $\langle x|y\rangle$, denotes the inner product of the two vectors. For instance, since $|0\rangle$ is a unit vector we have $\langle 0|0\rangle = 1$ and since $|0\rangle$ and $|1\rangle$ are orthogonal we have $\langle 0|1\rangle = 0$.

The notation $|x\rangle\langle y|$ is the outer product of $|x\rangle$ and $\langle y|$. For example, $|0\rangle\langle 1|$ is the transformation that maps $|1\rangle$ to $|0\rangle$ and $|0\rangle$ to $(0,0)^T$ since

$$\begin{aligned} |0\rangle\langle 1||1\rangle &= |0\rangle\langle 1|1\rangle = |0\rangle \\ |0\rangle\langle 1||0\rangle &= |0\rangle\langle 1|0\rangle = 0|0\rangle = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{aligned}$$

Equivalently, $|0\rangle\langle 1|$ can be written in matrix form where $|0\rangle = (1,0)^T$, $\langle 0| = (1,0)$, $|1\rangle = (0,1)^T$, and $\langle 1| = (0,1)$. Then

$$|0\rangle\langle 1| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} (0,1) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

This notation gives us a convenient way of specifying transformations on quantum states in terms of what happens to the basis vectors (see section 4). For example, the transformation that exchanges $|0\rangle$ and $|1\rangle$ is given by the matrix

$$X = |0\rangle\langle 1| + |1\rangle\langle 0|.$$

In this paper we will prefer the slightly more intuitive notation

$$\begin{array}{rccc} X: & |0\rangle & \rightarrow & |1\rangle \\ & |1\rangle & \rightarrow & |0\rangle \end{array}$$

that explicitly specifies transformations for the basis vectors.

3 Quantum Bits

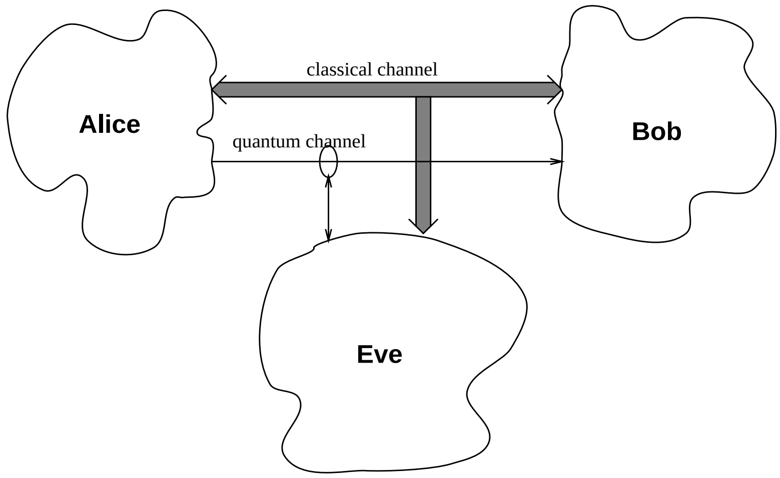
A quantum bit, or qubit, is a unit vector in a two dimensional complex vector space for which a particular basis, denoted by $\{|0\rangle, |1\rangle\}$, has been fixed. The orthonormal basis $|0\rangle$ and $|1\rangle$ may correspond to the $|\uparrow\rangle$ and $|\rightarrow\rangle$ polarizations of a photon respectively, or to the polarizations $|\nearrow\rangle$ and $|\nwarrow\rangle$. Or $|0\rangle$ and $|1\rangle$ could correspond to the spin-up and spin-down states of an electron.

For the purposes of quantum computing, the basis states $|0\rangle$ and $|1\rangle$ are taken to encode the classical bit values 0 and 1 respectively. Unlike classical bits however, qubits can be in a superposition of $|0\rangle$ and $|1\rangle$ such as $a|0\rangle + b|1\rangle$ where a and b are complex numbers such that $|a|^2 + |b|^2 = 1$. Just as in the photon polarization case, if such a superposition is measured with respect to the basis $\{|0\rangle, |1\rangle\}$, the probability that the measured value is $|0\rangle$ is $|a|^2$ and the probability that the measured value is $|1\rangle$ is $|b|^2$. When talking about qubits, and quantum computations in general, a fixed basis with respect to which all statements are made has been chosen in advance. In particular, unless otherwise specified, all measurements are made with respect to the standard basis for quantum computation, $\{|0\rangle, |1\rangle\}$.

Even though quantum bits can be put in a superposition state, it is only possible to encode a single classical bit in each quantum bit. From an information theory point of view, a qubit contains exactly the same amount of information as a classical bit, inspite of its having infinitely many more states. The reason that no more information can be contained in a qubit than in a classical bit is that information can only be extracted by measurement. When a qubit is measured, the measurement changes the state to one of the basis states in the way seen in the photon polarization experiment. As every measurement has an associated basis, and a qubit lives in a two dimensional space, a given measurement can only result in one of the two basis vectors associated with the measurement. So, just as in the classical case, for any measurement of a qubit, there are only two possible results. As measurement changes the state, it is not possible to measure first in one basis and then in another. Furthermore, as we shall see in section 4.1.2, quantum states cannot be cloned, so it is not possible to measure a qubit in two different ways even indirectly by, say, copying the qubit and measuring the copy.

3.1 Quantum Key Distribution

Sequences of single qubits can be used to transmit private keys on insecure channels. Bennett and Brassard in 1984 were the first to describe a quantum key distribution scheme [13], [14]. Consider the situation in which Alice and Bob want to agree on a secret key so that they can communicate privately. They are connected by an ordinary bi-directional open channel and a uni-directional quantum channel both of which can be observed by Eve, who wishes to eavesdrop on their conversation. This situation is illustrated in the figure below. The quantum channel allows Alice to send individual particles (e.g. photons) to Bob who can measure their quantum state. Eve can attempt to measure the state of these particles and can resend the particles to Bob.



To begin the process of establishing a secret key, Alice sends a sequence of bits to Bob by encoding each bit in the quantum state of a photon as follows. For each bit, Alice randomly uses one of the following two bases for encoding each bit:

$$\begin{array}{rcl} 0 & \rightarrow & |\uparrow\rangle \\ 1 & \rightarrow & |\rightarrow\rangle \end{array}$$

or

$$\begin{array}{rcl} 0 & \rightarrow & |\nwarrow\rangle \\ 1 & \rightarrow & |\nearrow\rangle. \end{array}$$

Bob measures the state of the photons he receives by randomly picking either basis. After the bits have been transmitted, Bob and Alice communicate the basis they used for encoding and decoding of each bit over the open channel. With this information both can determine which bits have been transmitted correctly, by identifying those bits for which the sending and receiving bases agree. They will use these bits as the key and discard all others. On average, Alice and Bob will agree on 50% of all bits transmitted.

Suppose that Eve measures the state of the photons transmitted by Alice and resends new photons with the measured state. In this process she will use the wrong basis approximately 50% of the time, in which case she will resend the bit with the wrong basis. So when Bob measures a resent qubit with the correct basis there will be a 25% probability that he measures the wrong value. Thus any eavesdropper on the quantum channel is bound to introduce a high error rate that Alice and Bob can detect by communicating a sufficient number of parity bits of their keys over the open channel. So, not only is it likely that Eve's version of the key is 25% incorrect, but the fact that someone is eavesdropping will be apparent to Alice and Bob.

Other techniques for exploiting quantum effects for key distribution have been proposed. See, for example, Ekert ([15]) and Bennet ([16]).

Quantum key distribution has been realized over a distance of 24 km using standard fiber optical cables [17].

3.2 Multiple Qubits

It is when examining systems of more than one qubit that one first gets a glimpse of where the computational power of quantum computers could come from. As we saw, the state of a qubit is a vector in the two dimensional complex vector space spanned by $|0\rangle$ and $|1\rangle$. In classical physics, the possible states of a system of n particles, whose individual states can be described by a vector in a two dimensional vector space, form a vector space of $2n$ dimensions. However, in a quantum system the resulting state space is much larger; a system of n qubits has a state space of 2^n dimensions.[4] It is this exponential growth of the

[4] Actually, as we shall see, the state space is the set of normalized vectors in this 2^n dimensional space, just as the state $a|0\rangle + b|1\rangle$ of a qubit is normalized so that $|a|^2 + |b|^2 = 1$.

state space with the number of particles that suggests a possible exponential speed-up of computation on quantum computers over classical computers.

Individual state spaces of n particles classically combine through the cartesian product. Quantum states, however, combine through the tensor product. Details on properties of tensor products and their expression in terms of vectors and matrices is given in Appendix A. Let us look briefly at distinctions between the cartesian product and the tensor product that will be crucial to understanding quantum computation.

Let V and W be two 2-dimensional complex vector spaces with bases $\{v_1, v_2\}$ and $\{w_1, w_2\}$ respectively. The cartesian product of these two spaces can take as its basis the union of the bases of its component spaces $\{v_1, v_2, w_1, w_2\}$. Note that the order of the basis was chosen arbitrarily. In particular, the dimension of the state space of multiple classical particles grows linearly with the number of particles, since $\dim(X \times Y) = \ dim(X) + \dim(Y)$. The tensor product of V and W has basis $\{v_1 \otimes w_1, v_1 \otimes w_2, v_2 \otimes w_1, v_2 \otimes w_2\}$. Note that the order of the basis, again, is arbitrary. So the state space for two qubits, each with basis $\{|0\rangle, |1\rangle\}$, has basis $\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$ which can be written more compactly as $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. More generally, we write $|x\rangle$ to mean $|b_n b_{n-1} \dots b_0\rangle$ where b_i are the binary digits of the number x.

A basis for a three qubit system is

$$\{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle\}$$

and in general an n qubit system has 2^n basis vectors. We can now see the exponential growth of the state space with the number of quantum particles. The tensor product $X \otimes Y$ has dimension $dim(X) \times dim(Y)$.

Imagine a macroscopic physical object breaking apart and multiple pieces flying off in different directions. The state of this system can be described completely by describing the state of each of its component pieces separately. A surprising and unintuitive aspect of the state space of an n particle quantum system is that the state of the system cannot always be described in terms of the state of its component pieces. For instance, the state $|00\rangle + |11\rangle$ cannot be decomposed into separate states for each of the two qubits. In other words, we cannot find a_1, a_2, b_1, b_2 such that

$$(a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) = |00\rangle + |11\rangle$$

since

$$(a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) = a_1a_2|00\rangle + a_1b_2|01\rangle + b_1a_2|10\rangle + b_1b_2|11\rangle$$

and $a_1b_2 = 0$ implies that either $a_1a_2 = 0$ or $b_1b_2 = 0$. States which cannot be decomposed in this way are called entangled states. These states represent situations that have no classical counterpart, and for which we have no intuition. These are also the states that provide the exponential growth of quantum state spaces with the number of particles.

Note that it would require vast resources to simulate even a small quantum system on a traditional computer, as such a simulation would require keeping track of exponentially many states. The reason for the potential power of quantum computers is the possibility of exploiting the quantum state evolution as a computational mechanism.

3.3 Measurement

Measurement of one or more particles in a quantum system results in a projection of the state of the system prior to measurement onto the subspace of the state space compatible with the measured values. The amplitude of the projection is then rescaled so that the resulting state vector has length one. The probability that the result of the measurement is a given value is the sum of the squares of the the absolute values of the amplitudes of all components compatible with that value of the measurement.

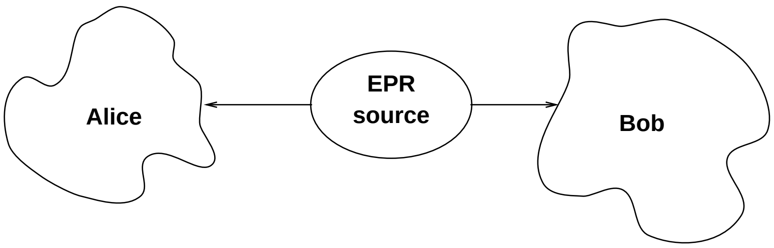
Let us look at an example of measurement in a two qubit system. From now on, unless otherwise specified, all measurements will be assumed to be measurements of individual qubits with respect to the basis $\{|0\rangle, ket1\}$. Any state of a two qubit system can be expressed as $a|00\rangle+b|01\rangle+c|10\rangle+d|11\rangle$, where a, b, c and d are complex numbers such that $|a|^2+|b|^2+|c|^2+|d|^2=1$. When the first qubit is measured with respect to the basis $\{|0\rangle, ket1\}$, the probability that the result is $|0\rangle$ is $|a|^2+|b|^2$. Furthermore, if the measurement gives the first qubit as $|0\rangle$, the state is projected onto the subspace compatible with the measurement, the subspace spanned by $|00\rangle$ and $|01\rangle$. The result of this projection is $a|00\rangle+b|01\rangle$. To get the state of the system after the measurement, we must renormalize so that the total probability is 1:

$$\frac{1}{\sqrt{|a|^2+|b|^2}}(a|00\rangle+b|01\rangle).$$

Measurement gives another way of thinking about entangled particles. Particles are not entangled if the measurement of one has no effect on the other. For instance, the state $\frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$ is entangled since the probability that the first bit is measured to be $|0\rangle$ is $1/2$ if the second bit has not been measured. However, if the second bit had previously been measured, the probability that the first bit is measured as $|0\rangle$ is either 1 or 0, depending on whether the second bit was measured as $|0\rangle$ or $|1\rangle$ respectively. Thus the probable results of measuring the first bit is changed by a measurement of the second bit. On the other hand, the state $\frac{1}{\sqrt{2}}(|00\rangle+|01\rangle)$ is not entangled: since $\frac{1}{\sqrt{2}}(|00\rangle+|01\rangle) = |0\rangle\otimes\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$, any measurement of the first bit will yield $|0\rangle$ regardless of whether the second bit was measured. Similarly, the second bit has a fifty-fifty chance of being measured as $|0\rangle$ regardless of whether the first bit was measured or not. Note that entanglement, in the sense that measurement of one particle has an effect on measurements of another particle, is equivalent to our previous definition of entangled states as states that cannot be written as a tensor product of individual states.

3.4 The EPR Paradox

Einstein, Podolsky and Rosen proposed a gedanken experiment that uses entangled particles in a manner that seemed to violate fundamental principles relativity. Imagine a source that generates two maximally entangled particles $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, called an EPR pair, and sends one each to Alice and Bob.



Alice and Bob can be arbitrarily far apart. Suppose that Alice measures her particle and observes state $|0\rangle$. This means that the combined state will now be $|00\rangle$ and if now Bob measures his particle he will also observe $|0\rangle$. Similarly, if Alice measures $|1\rangle$, so will Bob. Note that the change of the combined quantum state occurs instantaneously even though the two particles may be arbitrarily far apart. It appears that this would enable Alice and Bob to communicate faster than the speed of light. Further analysis, as we shall see, shows that even though there is a coupling between the two particles, there is no way for Alice or Bob to use this mechanism to communicate.

There are two standard ways that people use to describe entangled states and their measurement. Both have their positive aspects, but both are incorrect and can lead to misunderstandings. Let us examine both in turn.

Einstein, Podolsky and Rosen proposed that each particle has some internal state that completely determines what the result of any given measurement will be. This state is, for the moment, hidden from us, and therefore the best we can currently do is to give probabilistic predictions. Such a theory is known as a local hidden variable theory. The simplest hidden variable theory for an EPR pair is that the particles are either both in state $|0\rangle$ or both in state $|1\rangle$, we just don't happen to know which. In such a theory no communication between possibly distant particles is necessary to explain the correlated measurements. However, this point of view cannot explain the results of measurements with respect to a different basis. In fact, Bell showed that any local hidden variable theory predicts that certain measurements will satisfy an inequality, known as Bell's inequality. However, the result of actual experiments performing these measurements show that Bell's inequality is violated. Thus quantum mechanics cannot be explained by any local hidden variable theory. See [10] for a highly readable account of Bell's theorem and related experiments.

The second standard description is in terms of cause and effect. For example, we said earlier that a measurement performed by Alice affects a measurement performed by Bob. However, this view is incorrect also, and results, as Einstein, Podolsky and Rosen recognized, in deep inconsistencies when combined with relativity theory. It is possible to set up the EPR scenario so that one observer sees Alice measure first, then Bob, while another observer sees Bob measure first, then Alice. According to relativity, physics must equally well explain the observations of the first observer as the second. While our terminology of cause and effect cannot be compatible with both observers, the actual experimental values are invariant under change of observer. The values can be explained equally well by Bob's measuring first and causing a change in the state of Alice's particle, as the other way around. This symmetry shows that Alice and Bob cannot, in fact, use their EPR pair to communicate faster than the speed of light, and thus resolves the apparent paradox. All that can be said is that Alice and Bob will observe the same random behavior.

As we will see in the section on dense coding and teleportation, EPR pairs can be used to aid communication, albeit communication slower than the speed of light.

4 Quantum Gates

So far we have looked at static quantum systems which changed only when measured. The dynamics of a quantum system, when not being measured, are governed by Schrödinger's equation; the dynamics must take states to states in a way that preserves orthogonality. For a complex vector space, linear transformations that preserve orthogonality are unitary transformations, defined as follows. Any linear transformation on a complex vector space can be described by a matrix. Let M^* denote the conjugate transpose of the matrix M. A matrix M is unitary (describes a unitary transformation) if $MM^* = I$. Any unitary transformation of a quantum state space is a legitimate quantum transformation, and vice versa. One can think of unitary transformations as being rotations of a complex vector space.

One important consequence of the fact that quantum transformations are unitary is that they are reversible. Thus quantum gates must be reversible. Bennett, Fredkin, and Toffoli had already looked at reversible versions of standard computing models. See Feynman's *Lectures on Computation* [18] for an account of some of the central ideas in this work.

4.1 Simple Quantum Gates

The following are some examples of useful single-qubit quantum state transformations. Because of linearity, the transformations are fully specified by their effect on the basis vectors. The associated matrix is also shown.

$$
\begin{array}{llll}
I: & |0\rangle \to |0\rangle & & \multirow{2}{*}{} \\
 & |1\rangle \to |1\rangle & & \left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right) \\
X: & |0\rangle \to |1\rangle & & \\
 & |1\rangle \to |0\rangle & & \left(\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array}\right) \\
Y: & |0\rangle \to |1\rangle & & \\
 & |1\rangle \to -|0\rangle & & \left(\begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array}\right) \\
Z: & |0\rangle \to |0\rangle & & \\
 & |1\rangle \to -|1\rangle & & \left(\begin{array}{cc} 1 & 0 \\ 0 & -1 \end{array}\right)
\end{array}
$$

The names of these transformations are conventional. I is the identity transformation, X is negation, Z is a phase shift operation, and $Y = ZX$ is a combination of both. The X transformation was discussed previously in section 2.2. It can be readily verified that these gates are unitary. For example

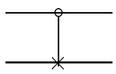
$$YY^* = \left(\begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array}\right)\left(\begin{array}{cc} 0 & 1 \\ -1 & 0 \end{array}\right) = I.$$

The controlled-NOT gate, C_{not}, operates on two qubits as follows: it changes the second bit if the first bit is 1 and leaves this bit unchanged otherwise.

$$
C_{not}: \begin{array}{lll} |00\rangle & \to & |00\rangle \\ |01\rangle & \to & |01\rangle \\ |10\rangle & \to & |11\rangle \\ |11\rangle & \to & |10\rangle \end{array} \qquad \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right)
$$

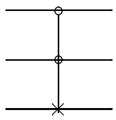
The transformation C_{not} is unitary since $C_{not}^* = C_{not}$ and $C_{not}C_{not} = I$. The C_{not} gate cannot be decomposed into a tensor product of two single-bit transformations.

It is useful to have graphical representations of quantum state transformations, especially when several transformations are combined. The controlled-NOT gate C_{not} is typically represented by a circuit of the form

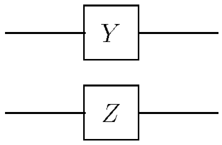


The open circle indicates the control bit, and the $\times$ indicates the conditional negation of the subject bit. In general there can be multiple control bits. Some authors use a solid circle to indicate negative control, in which the subject bit is toggled when the control bit is 0.

Similarly, the controlled-controlled-NOT, which negates the last bit of three if and only if the first two are both 1, has the following graphical representation.



Single bit operations are graphically represented by appropriately labelled boxes as shown.



4.1.1 The Walsh-Hadamard Transformation

Another important single-bit transformation is the Hadamard Transformation defined by

$$\begin{array}{rcl} H: \; |0\rangle & \to & \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle & \to & \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{array}$$

The transformation H has a number of important applications. When applied to $|0\rangle$, H creates a superposition state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Applied to n bits individually, H generates a superposition of all 2^n possible states, which can be viewed as the binary representations for the numbers from 0 to $2^n - 1$.

$$\begin{array}{rl} & (H \otimes H \otimes \ldots \otimes H)|00\ldots0\rangle \\ = & \frac{1}{\sqrt{2^n}}\left((|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes \ldots \otimes (|0\rangle + |1\rangle)\right) \\ = & \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle. \end{array}$$

The transformation that applies H to n bits is called the Walsh, or Walsh-Hadamard, transformation W. It can be defined as a recursive decomposition of the form

$$W_1 = H, W_{n+1} = H \otimes W_n.$$

4.1.2 No Cloning

The unitary property implies that quantum state cannot be cloned. The no cloning proof given here, originally due to Wootters and Zurek ([19]), is a simple application of the linearity of unitary transformations.

Assume that U is a unitary transformation that clones, in that $U(|a0\rangle) = |aa\rangle$ for all quantum states $|a\rangle$. Let $|a\rangle$ and $|b\rangle$ be two orthogonal quantum states. Say $U(|a0\rangle) = |aa\rangle$ and $U(|b0\rangle) = |bb\rangle$. Consider $|c\rangle = (1/\sqrt{2})(|a\rangle + |b\rangle)$. By linearity,

$$\begin{aligned} U(|c0\rangle) &= \tfrac{1}{\sqrt{2}}(U(|a0\rangle) + U(|b0\rangle)) \\ &= \tfrac{1}{\sqrt{2}}(|aa\rangle + |bb\rangle). \end{aligned}$$

But if U is a cloning transformation then

$$U(|c0\rangle) = |cc\rangle = 1/2(|aa\rangle + |ab\rangle + |ba\rangle + |bb\rangle),$$

which is not equal to $(1/\sqrt{2})(|aa\rangle + |bb\rangle)$. Thus there is no unitary operation that can reliably clone unknown quantum states.

It is important to understand what sort of cloning is and isn't allowed. It is possible to clone a known quantum state. What the no cloning principle tells us is that it is impossible to reliably clone an unknown quantum state. Also, it is possible to obtain n particles in an entangled state $a|00\ldots 0\rangle + b|11\ldots 1\rangle$ from an unknown state $a|0\rangle + b|1\rangle$. Each of these particles will behave in exactly the same way when measured with respect to the standard basis for quantum computation, $\{|0\ldots 00\rangle, |0\ldots 01\rangle, \ldots, |1\ldots 11\rangle\}$, but not when measured with respect to other bases. It is not possible to create the n particle state $(a|0\rangle + b|1\rangle) \otimes \ldots \otimes (a|0\rangle + b|1\rangle)$ from an unknown state $a|0\rangle + b|1\rangle$.

4.2 Examples

The use of simple quantum gates can be studied with two simple examples: dense coding and teleportation.

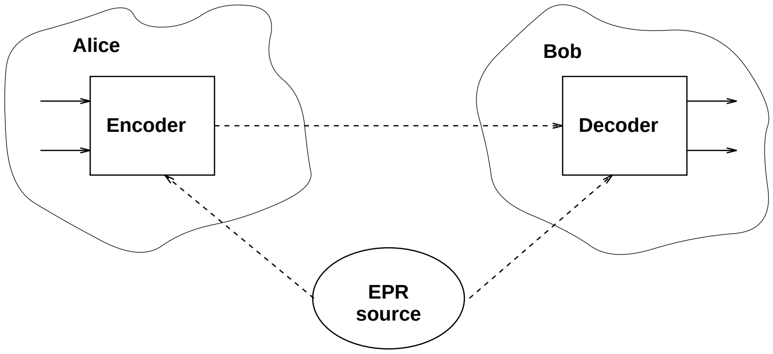
Dense coding uses one quantum bit together with an EPR pair to encode and transmit two classical bits. Since EPR pairs can be distributed ahead of time, only one qubit (particle) needs to be physically transmitted to communicate two bits of information. This result is surprising since, as was discussed in section 3, a qubit only contains one bit's worth of information. Teleportation is the opposite of dense coding, in that it uses two classical bits to transmit a single qubit. Teleportation is surprising in light of the no cloning principle of quantum mechanics, in that it enables the transmission of an unknown quantum state.

The key to both dense coding and teleportation is the use of entangled particles. The initial set up is the same for both processes. Alice and Bob wish to communicate. Each is sent one ot the entangled particles making up an EPR pair,

$$\psi_0 = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Say Alice is sent the first particle, and Bob the second. So until a particle is transmitted, only Alice can perform transformations on her particle, and only Bob can perform transformations on his.

4.2.1 Dense Coding



Alice: Alice receives two classical bits, encoding the numbers 0 through 3. Depending on this number Alice performs one of the transformations $\{I, X, Y, Z\}$ on her qubit of the entangled pair ψ_0. Transforming just one bit of an entangled pair means performing the identity transformation on the other bit. The resulting state is shown in the table.

Value	Transformation	New state
0	$\psi_0 = (I \otimes I)\psi_0$	$\frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$
1	$\psi_1 = (X \otimes I)\psi_0$	$\frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$
2	$\psi_2 = (Y \otimes I)\psi_0$	$\frac{1}{\sqrt{2}}(-\|10\rangle + \|01\rangle)$
3	$\psi_3 = (Z \otimes I)\psi_0$	$\frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$

Alice then sends her qubit to Bob.

Bob: Bob applies a controlled-NOT to the two qubits of the entangled pair.

Initial state	Controlled-NOT	First bit	Second bit
$\psi_0 = \frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|00\rangle + \|10\rangle)$	$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$	$\|0\rangle$
$\psi_1 = \frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$	$\frac{1}{\sqrt{2}}(\|11\rangle + \|01\rangle)$	$\frac{1}{\sqrt{2}}(\|1\rangle + \|0\rangle)$	$\|1\rangle$
$\psi_2 = \frac{1}{\sqrt{2}}(-\|10\rangle + \|01\rangle)$	$\frac{1}{\sqrt{2}}(-\|11\rangle + \|01\rangle)$	$\frac{1}{\sqrt{2}}(-\|1\rangle + \|0\rangle)$	$\|1\rangle$
$\psi_3 = \frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|00\rangle - \|10\rangle)$	$\frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$	$\|0\rangle$

Note that Bob can now measure the second qubit without disturbing the quantum state. If the measurement returns $|0\rangle$ then the encoded value was either 0 or 3, if the measurement returns $|1\rangle$ then the encoded value was either 1 or 2.

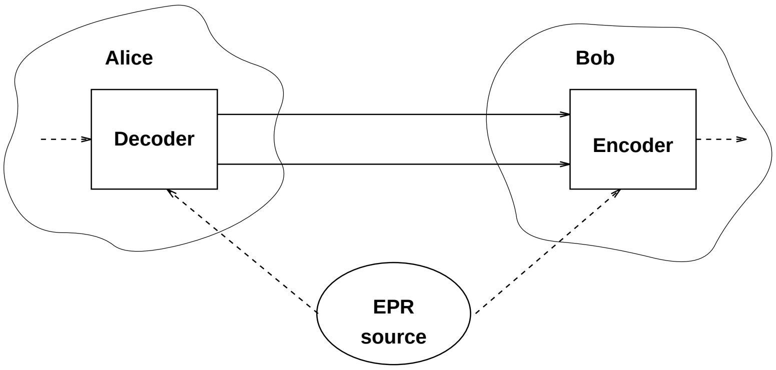
Bob now applies H to the first bit:

Initial state	First bit	H(First bit)
ψ_0	$\frac{1}{\sqrt{2}}(\lvert 0\rangle + \lvert 1\rangle)$	$\frac{1}{\sqrt{2}}\big(\frac{1}{\sqrt{2}}(\lvert 0\rangle + \lvert 1\rangle) + \frac{1}{\sqrt{2}}(\lvert 0\rangle - \lvert 1\rangle)\big) = \lvert 0\rangle$
ψ_1	$\frac{1}{\sqrt{2}}(\lvert 1\rangle + \lvert 0\rangle)$	$\frac{1}{\sqrt{2}}\big(\frac{1}{\sqrt{2}}(\lvert 0\rangle - \lvert 1\rangle) + \frac{1}{\sqrt{2}}(\lvert 0\rangle + \lvert 1\rangle)\big) = \lvert 0\rangle$
ψ_2	$\frac{1}{\sqrt{2}}(-\lvert 1\rangle + \lvert 0\rangle)$	$\frac{1}{\sqrt{2}}\big(-\frac{1}{\sqrt{2}}(\lvert 0\rangle - \lvert 1\rangle) + \frac{1}{\sqrt{2}}(\lvert 0\rangle + \lvert 1\rangle)\big) = \lvert 1\rangle$
ψ_3	$\frac{1}{\sqrt{2}}(\lvert 0\rangle - \lvert 1\rangle)$	$\frac{1}{\sqrt{2}}\big(\frac{1}{\sqrt{2}}(\lvert 0\rangle + \lvert 1\rangle) - \frac{1}{\sqrt{2}}(\lvert 0\rangle - \lvert 1\rangle)\big) = \lvert 1\rangle$

Finally, Bob measures the resulting bit which allows him to distinguish between 0 and 3, and 1 and 2.

4.2.2 Teleportation

The objective is to transmit the quantum state of a particle using classical bits and reconstruct the exact quantum state at the receiver. Since quantum state cannot be cloned, the quantum state of the given particle will necessarily be destroyed. Single bit teleportation has been realized experimentally 1997 ([20]).



Alice: Alice wants to send the state of the qubit

$$\phi = a\lvert 0\rangle + b\lvert 1\rangle$$

to Bob through classical channels. As with dense coding, Alice and Bob each possess one qubit of an entangled pair

$$\psi_0 = \frac{1}{\sqrt{2}}(\lvert 00\rangle + \lvert 11\rangle).$$

Alice applies the decoding step of dense coding to the qubit ϕ to be transmitted and her half of the entangled pair. The starting state is quantum state

$$\begin{aligned}\phi \otimes \psi_0 &= \frac{1}{\sqrt{2}}\big(a\lvert 0\rangle \otimes (\lvert 00\rangle + \lvert 11\rangle) + b\lvert 1\rangle \otimes (\lvert 00\rangle + \lvert 11\rangle)\big) \\ &= \frac{1}{\sqrt{2}}\big(a\lvert 000\rangle + a\lvert 011\rangle + b\lvert 100\rangle + b\lvert 111\rangle\big),\end{aligned}$$

of which Alice controls the first two bits and Bob controls the last one. Alice now applies $C_{not} \otimes I$ and $H \otimes I \otimes I$ to this state:

$$
\begin{aligned}
&(H \otimes I \otimes I)(C_{not} \otimes I)(\phi \otimes \psi_0) \\
&= (H \otimes I \otimes I)(C_{not} \otimes I)\frac{1}{\sqrt{2}}\big(a|000\rangle + a|011\rangle + b|100\rangle + b|111\rangle\big) \\
&= (H \otimes I \otimes I)\frac{1}{\sqrt{2}}\big(a|000\rangle + a|011\rangle + b|110\rangle + b|101\rangle\big) \\
&= \frac{1}{2}\big(a(|000\rangle + |011\rangle + |100\rangle + |111\rangle) + b(|010\rangle + |001\rangle - |110\rangle - |101\rangle)\big) \\
&= \frac{1}{2}\big(|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(a|1\rangle + b|0\rangle) + |10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(a|1\rangle - b|0\rangle)\big)
\end{aligned}
$$

Alice measures the first two qubits to get one of $|00\rangle$, $|01\rangle$, $|10\rangle$, or $|11\rangle$ with equal probability. Depending on the result of the measurement, the quantum state of Bob's qubit is projected to $a|0\rangle+b|1\rangle$, $a|1\rangle+b|0\rangle$, $a|0\rangle-b|1\rangle$, or $a|1\rangle-b|0\rangle$ respectively. Alice sends the result of her measurement as two classical bits to Bob.

Note that when she measured it, Alice irretrievably altered the state of her original qubit ϕ, whose state she is in the process of sending to Bob. This loss of the original state is the reason teleportation does not violate the no cloning principle.

Bob: When Bob receives the two classical bits from Alice he knows how the state of his half of the entangled pair compares to the original state of Alice's qubit.

bits received	state	decoding
00	$a\|0\rangle + b\|1\rangle$	I
01	$a\|1\rangle + b\|0\rangle$	X
10	$a\|0\rangle - b\|1\rangle$	Z
11	$a\|1\rangle - b\|0\rangle$	Y

Bob can reconstruct the original state of Alice's qubit, ϕ, by applying the appropriate decoding transformation to his part of the entangled pair. Note that this is the encoding step of dense coding.

5 Quantum Computer

This section discusses how quantum mechanics can be used to perform computations and how these computations are qualitatively different from those performed by a conventional computer. Recall that all quantum state transformations have to be reversible. While the classical NOT gate is reversible, AND, OR and NAND gates are not. Thus it is not obvious that quantum transformations can carry out all classical computations. The first section describes complete sets of reversible gates that can perform any classical computation on a quantum computer. Furthermore, it describes sets of gates with which all quantum computations can be done. The second subsection discusses quantum parallelism.

5.1 Quantum Gate Arrays

The bra/ket notation is useful in defining complex unitary operations. For two arbitrary unitary transformations U_1 and U_2, the "conditional" transformation $|0\rangle\langle 0| \otimes U_1 + |1\rangle\langle 1| \otimes U_2$ is also unitary. The controlled-NOT gate can defined by

$$C_{not} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X.$$

The three-bit controlled-controlled-NOT gate or Toffoli gate of section 4 is also an instance of this conditional definition:

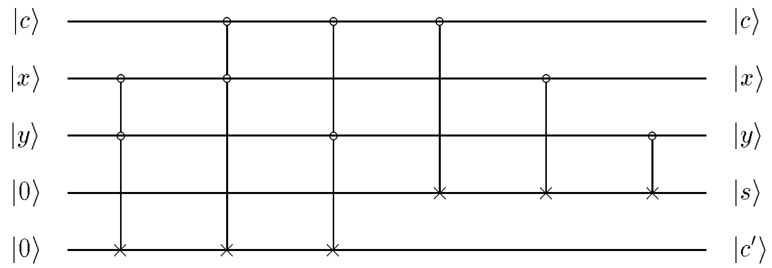
$$T = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes C_{not}.$$

T can be used to construct complete set of boolean connectives in that it can be used to construct the NOT and AND operators in the following way:

$$\begin{array}{rcl} T|1,1,x\rangle & = & |1,1,\neg x\rangle \\ T|x,y,0\rangle & = & |x,y,x \wedge y\rangle \end{array}$$

The T gate is sufficient to construct arbitrary combinatorial circuits.

The following quantum circuit, for example, implements a 1 bit full adder using Toffoli and controlled-NOT gates:



where x and y are the data bits, s is their sum (modulo 2), c is the incoming carry bit, and c' is the new carry bit. Vedral, Barenco and Ekert ([21]) define more complex circuits that include in-place addition and modular addition.

The Fredkin gate is a "controlled swap" and can be defined as

$$F = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes S$$

where S is the swap operation

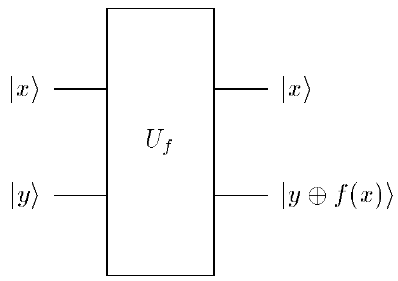
$$S = |00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11|.$$

The following table shows that F, like T, is complete for combinatorial circuits:

$$\begin{array}{rcl} F|x,0,1\rangle & = & |x,x,\neg x\rangle \\ F|x,y,1\rangle & = & |x, y \vee x, y \vee \neg x\rangle \\ F|x,0,y\rangle & = & |x, y \wedge x, y \wedge \neg x\rangle \end{array}$$

Deutsch has shown [22] that it is possible to construct reversible quantum gates for any arbitrary classically computable function. In fact, it is possible to conceive of a universal quantum Turing machine ([23]). In this construction we must assume a sufficient supply of bits that correspond to the tape of a Turing machine.

Knowing that an arbitrary classical function f can be implemented on quantum computer, we assume the existence of a *quantum gatearray* U_f that implements f. The transformation is of the form $U_f|x,y\rangle \to |x, y \oplus f(x)\rangle$ where $\oplus$ does not denote the direct sum of vectors, but rather the bitwise exclusive-OR. U_f defined in this way is unitary for any function f. To compute $f(x)$ we apply U_f to $|x,0\rangle$. Since $f(x) \oplus f(x) = 0$ we have $U_f U_f = I$. Graphically the transformation $U_f : |x,y\rangle \to |x, y \oplus f(x)\rangle$ is depicted as



While the T and F gates are complete for combinatorial circuits, they cannot achieve arbitrary quantum state transformations. In order to realize arbitrary unitary transformations, single bit rotations need to be included. Barenco et. al. show in [24] that C_{not} together with all 1-bit quantum gates is a universal gate set. It suffices to include the following single bit rotations and phase shift transformations

$$\begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}, \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix}, \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$$

for all α together with the C_{not} to obtain a universal set of gates. As we shall see, such non-classical rotations and phase shifts are crucial for exploiting the power of quantum computers.

5.2 Quantum Parallelism

What happens if U_f is applied to input which is in a superposition? The answer is easy but powerful: since U_f is a linear transformation, it is applied to all basis vectors in the superposition simultaneously and will generate a superposition of the results. In this way, it is possible to compute $f(x)$ for n values of x in a single application of U_f. This effect is called quantum parallelism.

The power of quantum algorithms comes from taking advantage of quantum parallelism. So most quantum algorithms begin by computing a function of interest on a superposition of all values as follows. Start with an n-qubit state $|00\ldots0\rangle$. Apply the Walsh-Hadamard transformation W of section 4.1.1 to get a superposition

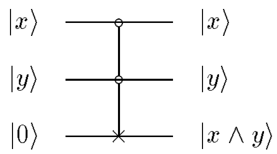
$$\frac{1}{\sqrt{2^n}}(|00\ldots0\rangle + |00\ldots1\rangle + \ldots + |11\ldots1\rangle) = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle$$

which should be viewed as the superposition of all integers $0 \leq x < 2^n$. By linearity

$$\begin{aligned} U_f(\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x,0\rangle) &= \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}U_f(|x,0\rangle) \\ &= \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x,f(x)\rangle \end{aligned}$$

where $f(x)$ is the function of interest. Note that since n qubits enable working simultaneously with 2^n states, quantum parallelism circumvents the time/space trade-off of classical parallelism through its ability to provide an exponential amount of computational space in a linear amount of physical space.

Consider the trivial example of a double controlled-NOT (Toffoli) gate, T, that computes the conjunction of two values:



Now take as input a superposition of all possible bit combinations of x and y together with the necessary 0

$$
\begin{aligned}
H|0\rangle \otimes H|0\rangle \otimes |0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\
&= \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle)
\end{aligned}
$$

Superposition of inputs leads to superposition of results, namely

$$
T(H|0\rangle \otimes H|0\rangle \otimes |0\rangle) = \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |111\rangle)
$$

The resulting superposition can be viewed as a truth table for the conjunction, or more generally as the graph of a function. In the output the values of x, y, and $x \wedge y$ are entangled in such a way that measuring the result will give one line of the truth table, or more generally one point of the function graph. Note that the bits can be measured in any order: measuring the result will project the state to a superposition of the set of all input values for which f produces this result; measuring the input will project the result to the corresponding function value.

The heart of any quantum algorithm is the way in which it manipulates quantum parallelism so that desired results will be measured with high probability. This sort of manipulation has no classical analog, and requires non-traditional programming techniques. We list a couple of the techniques currently known.

- Amplify output values of interest. The general idea is to transform the state in such a way that value of interest have a larger amplitude and have therefore a higher probability of being measured. Examples of this approach will be described in section 7.
- Find common properties of all the values of $f(x)$. This idea is exploited in Shor's algorithm which uses a quantum Fourier transformation to obtain the period of f.

6 Shor's Algorithm

In 1994, inspired by work of Daniel Simon [25], Peter Shor found a bounded probability polynomial time algorithm for factoring n-digit numbers on a quantum computer. Since the 1970's people have searched for efficient algorithms for factoring integers. The most efficient classical algorithm known today is that of Lenstra and Lenstra [26] which is exponential in the size of the input. The input is the list of digits of M, which has size $n \sim \log M$. People were confident enough that no efficient algorithm existed, that the security of cryptography systems, like the widely used RSA algorithm, depend on the difficulty of this problem. Shor's result surprised the community at large, prompting widespread interest in quantum computing.

Most factoring algorithms, including Shor's, use a standard reduction of the factoring problem to the problem of finding the period of a function. Shor uses quantum parallelism in the standard way to obtain a superposition of all the values of the function in one step. He then computes the quantum Fourier transform of the function, which like classical Fourier transforms, puts all the amplitude of the function into multiples of the reciprocal of the period. With high probability, measuring the state yields the period, which in turn is used to factor the integer M.

The above description is something of an oversimplification of the algorithm. The biggest complication is that the quantum Fourier transform is based on the fast Fourier transform and thus gives only approximate results in most cases. Thus extracting the period is trickier than outlined above. Also, there is the trivial complication that the quantum Fourier transform is scaled to output a function with integer domain, so strictly speaking a fraction is not measured.

We will first describe the quantum Fourier transform and then give a detailed outline of Shor's algorithm.

6.1 The Quantum Fourier Transform

The quantum Fourier transform is a variant of the discrete Fourier transform (DFT). The DFT sends a discrete function to another discrete function, conventionally having as its domain equally spaced points $k\frac{2\pi}{N}$ in the interval $[0, 2\pi)$ for some N. By scaling the domain by $\frac{N}{2\pi}$, the quantum Fourier transform (QFT) outputs a function with domain the integers between 0 and $N-1$.

The quantum Fourier transform operates on the amplitude of the quantum state, by sending

$$\sum_x g(x)|x\rangle \to \sum_c G(c)|c\rangle$$

where $G(c)$ is the discrete Fourier transform of $g(x)$, and x and c both range over the binary representations for the integers between 0 and $N-1$. If the state were measured after the Fourier transform was performed, the probability that the result was $|c\rangle$ would be $|G(c)|^2$. Note that the quantum Fourier transform does not output a function the way the U_f transformation does; no output appears in an extra register.

Fourier transforms in general map from the time domain to the frequency domain. So Fourier transforms map functions of period r to functions which have non-zero values only at multiples of the frequency $\frac{1}{r}$. Thus applying the quantum Fourier transform to a periodic function $g(x)$ with period r, we would expect to end up with $\sum_c G(c)|c\rangle$, where $G(c)$ is zero except for multiples of $\frac{N}{r}$. Thus, when the state is measured, the result would be a multiple of $\frac{N}{r}$, say $j\frac{N}{r}$.

The quantum Fourier transform works only approximately as described in the last paragraph. The quantum Fourier transform is a variant of the fast Fourier transform (FFT) which is based on powers of two, and only gives approximate results for periods which are not a power of two. However the larger the power of two used as a base for the transform, the better the approximation. The quantum Fourier transform U_{QFT} with base 2^m is defined by

$$U_{QFT} : |x\rangle \to \frac{1}{\sqrt{2^m}} \sum_{c=0}^{2^m-1} e^{\frac{2\pi i c x}{2^m}} |c\rangle.$$

In order for Shor's algorithm to be a polynomial algorithm, the quantum Fourier transform must be efficiently computable. Shor shows that the quantum Fourier transform with base 2^m can be constructed using only $\frac{m(m+1)}{2}$ gates. The constuction makes use of two types of gates. One is a gate to perform the familiar Hadamard transformation H. We will denote by H_j the Hadamard transformation applied to the jth bit. The other type of gate performs transformations of the form

$$S_{j,k} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta_{k-j}} \end{pmatrix}$$

where $\theta_{k-j} = \pi/2^{k-j}$ which acts on the kth element depending on the value of the jth element. The quantum Fourier transform is given by

$$H_0 S_{0,1} \ldots S_{0,m-1} H_1 \ldots H_{m-3} S_{m-3,m-2} S_{m-3,m-1} H_{m-2} S_{m-2,m-1} H_{m-1}$$

followed by a bit reversal transformation. See [3] for more details.

6.2 A Detailed Outline of Shor's algorithm

Step 1. Quantum parallelism Choose an integer a arbitrarily. If a is not relatively prime to M, we've found a factor of M. Otherwise apply the rest of the algorithm.

Let m be such that $M^2 \leq 2^m < 2M^2$. [This choice is made so that the approximation for non powers of 2 given by the quantum Fourier transform used in Step 3 will be good enough for the rest of the algorithm to work.] Use quantum parallelism as described in 5.2 to compute $f(x) = a^x \bmod M$ for all integers from 0 to $2^m - 1$. The function is thus encoded in the quantum state

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

Step 2. A state whose amplitude has the same period as f The quantum Fourier transform acts on the amplitude function associated with the input state. In order to use the quantum Fourier transform to obtain the period of f, a state is constructed whose amplitude function has the same period as f.

To construct such a state, measure the the qubits of the state obtained in Step 1 that encode $f(x)$. A random value u is obtained. The value u is not of interest in itself; only the effect the measurement has on our set of superpositions is of interest. This measurement projects the state space onto the subspace compatible with the measured value, so the state after measurement is

$$C\sum_x g(x)|x, u\rangle,$$

for some scale factor C where

$$g(x) = \begin{cases} 1 & \text{if } f(x) = u \\ 0 & \text{otherwise} \end{cases}$$

Note that the x's that actually appear in the sum, those with $g(x) \neq 0$, differ from each other by multiples of the period, thus $g(x)$ is the function we are looking for. If we could just measure two successive x's in the sum, we would have the period. Unfortunately the laws of quantum physics permit only one measurement.

Step 3. Applying a quantum Fourier transform The $|u\rangle$ part of the state will not be used, so we will no longer write it. Apply the quantum Fourier transform to the state obtained in Step 2.

$$U_{QFT} : \sum_x g(x)|x\rangle \to \sum_c G(c)|c\rangle$$

Standard Fourier analysis tells us that when the period r of $g(x)$ is a power of two, the result of the quantum Fourier transform is

$$C'\sum_j \rho_j |j\frac{2^m}{r}\rangle$$

where $|\rho_j| = 1$. When the period r does not divide 2^m, the transform approximates the exact case so most of the amplitude is attached to integers close to multiples of $\frac{2^m}{r}$.

Step 4. Extracting the period Measure the state in the standard basis for quantum computation, and call the result v. In the case where the period happens to be a power of 2 so that the quantum Fourier transform gives exactly multiples of the scaled frequency, the period is easy to extract.

In this case, $v = j\frac{2^m}{r}$ for some j. Most of the time j and r will be relatively prime, in which case reducing the fraction $\frac{v}{2^m}$ to it's lowest terms will yield a fraction whose denominator q is the period r. The fact that in general the quantum Fourier transform only gives approximately multiples of the scaled frequency complicates the extraction of the period from the measurement. When the period is not a power of 2, a good guess for the period can be obtained using the continued fraction expansion of $\frac{v}{2^m}$. This technique is described in Appendix B.

Step 5. Finding a factor of M When our guess for the period, q, is even, use the Euclidean algorithm to efficiently check whether either $a^{q/2}+1$ or $a^{q/2}-1$ has a non-trivial common factor with M.

The reason why $a^{q/2}+1$ or $a^{q/2}-1$ is likely to have a non-trivial common factor with M is as follows. If q is indeed the period of $f(x) = a^x \bmod M$, then $a^q = 1 \bmod M$ since $a^q a^x = a^x \bmod M$ for all x. If q is even, we can write

$$(a^{q/2}+1)(a^{q/2}-1) = 0 \bmod M.$$

Thus, so long as neither $a^{q/2}+1$ nor $a^{q/2}-1$ is a multiple of M, either $a^{q/2}+1$ or $a^{q/2}-1$ has a non-trivial common factor with M.

Step 6. Repeating the algorithm, if necessary Various things could have gone wrong so that this process does not yield a factor of M:

1. The value v was not close enough to a multiple of $\frac{2^m}{r}$.
2. The period r and the multiplier j could have had a common factor so that the denominator q was actually a factor of the period not the period itself.
3. Step 5 yields M as M's factor.
4. The period of $f(x) = a^x \bmod M$ is odd.

A few repetitions of this algorithm yields a factor of M with high probability.

6.2.1 A Comment on Step 2 of Shor's Algorithm

Step 2 can be skipped entirely. Apply the quantum Fourier transform tensor the identity, $U_{QFT} \otimes I$, to $C\sum_{x=0}^{2^n-1} |x, f(x)\rangle$ to get

$$C' \sum_{x=0}^{2^n-1} \sum_{c=0}^{2^m-1} e^{\frac{2\pi i x c}{2^m}} |c, f(x)\rangle,$$

which is equal to

$$C' \sum_{u} \sum_{x|f(x)=u} \sum_{c} e^{\frac{2\pi i x c}{2^m}} |c, u\rangle$$

for u in the range of $f(x)$. What results is a superposition of the results of Step 3 for all possible u's. The quantum Fourier transform is being applied to a bunch of separate functions g_u indexed by u where

$$g_u = \begin{cases} 1 & \text{if } f(x) = u \\ 0 & \text{otherwise} \end{cases}$$

The transform $U_{QFT} \otimes I$ as applied above can be written

$$U_{QFT} \otimes I : C \sum_{u \in R} \sum_{x=0}^{2^n-1} g_u(x)|x, f(x)\rangle \to C' \sum_{u \in R} \sum_{x=0}^{2^n-1} \sum_{c=0}^{2^n-1} G_u(c)|c, u\rangle,$$

where $G_u(c)$ is the discrete Fourier transform of $g_u(x)$ and R is the range of $f(x)$.

Measure c and run Steps 4 and 5 as before.

7 Search Problems

A large class of problems can be specified as search problems of the form "find some x such that $P(x)$ is true" for some predicate P. Such problems range from sorting to graph coloring to database search. For example:

- Given an n element vector A, find a permutation π on $[1..n]$ such that $\forall 1 \leq i < n : A_{\pi(i)} < A_{\pi(i+1)}$.
- Given a graph (V, E) with n vertices V and e edges $E \subseteq V \times V$ and a set of k colors C, find a mapping c from V to C such that $\forall (v_1, v_2) \in E : c(v_1) \neq c(v_2)$.

For certain types of problems, where there is some problem structure that can be exploited, efficient algorithms are known. Many search problems, like constraint satisfaction problems such as 3-SAT and graph colorability, or searching an alphabeticized list, have structured search spaces in which full solutions can be built from smaller partial solutions. But in the general case with no structure, randomly testing predicates $P(x_i)$ one by one is the best that can be done classically. For a search space of size N, the general unstructured search problem is of complexity $O(N)$, once the time it takes to test the predicate P is factored out. On a quantum computer, however, Grover showed that the unstructured search problem can be solved with bounded probability within $O(\sqrt{N})$ time. Thus Grover's search algorithm ([27]) is provably more efficient than any algorithm that could run on a classical computer.

Grover's search algorithm searches a completely unstructured solution space. While Grover's algorithm is optimal [28] [29] [30], for completely unstructured searches, most search problems involve searching a structured solution space. One would expect that this structure would enable more efficient searching strategies. For example, constraint satisfaction problems, such as SAT problems and graph colorability, have structured search spaces in which full solutions can be built from smaller partial solutions. One would expect that for such problems there would be more efficient search methods than Grover's that would take advantage of this problem structure.

Tad Hogg has developed quantum algorithms that use the problem structure in a similar way to classical heuristic search algorithms. One problem with this approach is that the introduction of problem structure makes the algorithms complicated enough that it is hard to determine the probability that a single iteration of the algorithm will give a correct answer. Therefore it is unknown how efficient Hogg's algorithms are. Classically the efficiency of heuristic algorithms is estimated by empirically testing the algorithm. But as there is an exponential slow down when simulating a quantum computer on a classical one, empirical testing of quantum algorithms is currently infeasible except in small cases. Small cases indicate that Hogg's algorithms are more efficient than Grover's algorithm applied to structured search problems, but that the speed up is likely to be only polynomial. Until sufficiently large quantum computers are built, or better techniques for analyzing such algorithms are found, the efficiency cannot be determined for sure.

7.1 Grover's Search Algorithm

Grover's algorithm searches an unstructured list of size N. Let n be such that $2^n \geq N$. Assume that predicate P on n-bit values x is implemented by a quantum gate U_P:

$$U_P : |x, 0\rangle \to |x, P(x)\rangle$$

where true is encoded as 1. The first step is the standard one for quantum computing described in section 5.2. Compute P for all possible inputs x_i by applying U_P to a register containing the superposition $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x\rangle$ of all 2^n possible inputs x together with a register set to 0:

$$U_P : \frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x, 0\rangle \to \frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x, P(x)\rangle.$$

The difficult step is to obtain a useful result from this superposition.

For any x_0 such that $P(x_0)$ is true, $|x_0, 1\rangle$ will be part of the superposition $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x, P(x)\rangle$, but since its amplitude is $\frac{1}{\sqrt{2^n}}$, the probability that a measurement of the superposition produces x_0 is only 2^{-n}. The trick is to change the quantum state $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x, P(x)\rangle$ so as to greatly increase the amplitude of vectors $|x_0, 1\rangle$, for which the predicate is true, and decrease the amplitude of vectors $|x, 0\rangle$, for which the predicate is false.

Once such a transformation of the quantum state has been performed, one can simply measure the last qubit of the quantum state, which represents

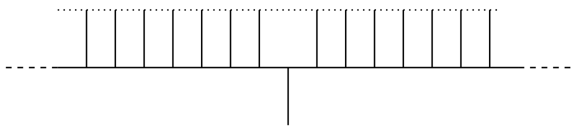
$P(x)$. Because of the amplitude change, there is a high probability that the result will be 1. If this is the case, the measurement has projected the state $\frac{1}{\sqrt{2^n}}\sum_{x=0}^{n-1}|x, P(x)\rangle$ onto the subspace $\frac{1}{\sqrt{2^k}}\sum_{i=1}^{k}|x_i, 1\rangle$ where k is the number of solutions. Further measurement of the remaining bits will provide one of these solutions. If the measurement of qubit $P(x)$ yields 0, then the whole process is started over and the superposition $\frac{1}{\sqrt{2^n}}\sum_{x=0}^{n-1}|x, P(x)\rangle$ must be computed again.

Grover's algorithm then consists of the following steps:

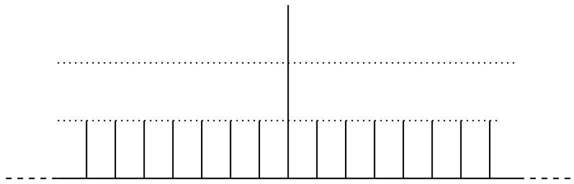
1. Prepare a register containing a superposition of all of the possible values $x_i \in [0 \ldots 2^n - 1]$.

2. Compute $P(x_i)$ on this register.

3. Change amplitude a_j to $-a_j$ for x_j such that $P(x_j) = 1$. An efficient algorithm for changing selected signs is described in section 7.1.2. A plot of the amplitudes after this step is shown here.



4. Apply inversion about the average to increase amplitude of x_j with $P(x_j) = 1$. The quantum algorithm to efficiently perform inversion about the average is given in section 7.1.1. The resulting amplitudes look as shown, where the amplitude of all the x_i's with $P(x_i) = 0$ have been diminished imperceptibly.



5. Repeat steps 2 through 4 $\frac{\pi}{4}\sqrt{2^n}$ times.

6. Read the result.

Boyer et.al. ([29]) provide a detailed analysis of the performance of Grover's algorithm. They prove that Grover's algorithm is optimal up to a constant factor; no quantum algorithm can perform an unstructured search faster. They also show that if there is only a single x_0 such that $P(x_0)$ is true, then after $\frac{\pi}{8}\sqrt{2^n}$ iterations of steps 2 through 4 the failure rate, is 0.5. After iterating

$\frac{\pi}{4}\sqrt{2^n}$ times the failure rate drops to 2^{-n}. Interestingly, additional iterations will increase the failure rate. For example, after $\frac{\pi}{2}\sqrt{2^n}$ iterations the failure rate is close to 1.

There are many classical algorithms in which a procedure is repeated over and over again for ever better results. Repeating quantum procedures may improve results for a while, but after a sufficient number of repetitions the results will get worse again. Quantum procedures are unitary transformations, which are rotations of complex space, and thus while a repeated applications of a quantum transform may rotate the state closer and closer to the desired state for a while, eventually it will rotate past the desired state to get farther and farther from the desired state. Thus to obtain useful results from a repeated application of a quantum transformation, it must be known when to stop.

7.1.1 Inversion about the Average

To perform inversion about the average on a quantum computer the inversion must be a unitary transformation. Furthermore, in order for the algorithm as a whole to solve the problem in $O(\sqrt{N})$ time, the inversion must be able to be performed efficiently. As will be shown shortly, the inversion can be accomplished with $O(n) = O(\log(N))$ quantum gates.

It is easy to see that the transformation

$$\sum_{i=0}^{N-1} a_i|x_i\rangle \to \sum_{i=0}^{N-1}(2A - a_i)|x_i\rangle$$

is performed by the $N \times N$ matrix

$$D = \begin{pmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \dots & \frac{2}{N} \\ \dots & \dots & \dots & \dots \\ \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} - 1 \end{pmatrix}.$$

Since $DD^* = I$, D is unitary and is therefore a possible quantum state transformation.

We now turn to the question of how efficiently the transformation can be performed, and show that it can be decomposed into $O(n) = O(\log(N))$ elementary quantum gates. Following Grover, D can be defined as $D = WRW$ where W is the Walsh-Hadamard transform defined in section 4 and

$$R = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots \\ 0 & \dots & \dots & 0 \\ 0 & \dots & 0 & -1 \end{pmatrix}.$$

To see that $D = WRW$, consider $R = R' - I$ where I is the identity and

$$R' = \begin{pmatrix} 2 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots \\ 0 & \dots & \dots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}.$$

Now $WRW = W(R' - I)W = WR'W - I$. It is easily verified that

$$WR'W = \begin{pmatrix} \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} & \frac{2}{N} & \dots \\ \frac{2}{N} & \dots & \dots & \frac{2}{N} \\ \frac{2}{N} & \dots & \frac{2}{N} & \frac{2}{N} \end{pmatrix}$$

and thus $WR'W - I = D$.

7.1.2 Changing the Sign

We still have to explain how to invert the amplitude of the desired result. We show, more generally, a simple and surprising way to invert the amplitude of exactly those states with $P(x) = 1$ for a general P.

Let U_P be the gate array that performs the computation $U_P : |x, b\rangle \to |x, b \oplus P(x)\rangle$. Apply U_P to the superposition $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x\rangle$ and choose $b = \frac{1}{\sqrt{2}}|0\rangle - |1\rangle$ to end up in a state where the sign of all x with $P(x) = 1$ has been changed, and b is unchanged.

To see this, let $X_0 = \{x | P(x) = 0\}$ and $X_1 = \{x | P(x) = 1\}$ and consider the application of U_P.

$$\begin{aligned}
&U_P(|\psi, b\rangle) \\
&= \frac{1}{\sqrt{2^{n+1}}} U_P\Big(\sum_{x \in X_0} |x, 0\rangle + \sum_{x \in X_1} |x, 0\rangle - \sum_{x \in X_0} |x, 1\rangle - \sum_{x \in X_1} |x, 1\rangle\Big) \\
&= \frac{1}{\sqrt{2^{n+1}}} \Big(\sum_{x \in X_0} |x, 0 \oplus 0\rangle + \sum_{x \in X_1} |x, 0 \oplus 1\rangle - \sum_{x \in X_0} |x, 1 \oplus 0\rangle - \sum_{x \in X_1} |x, 1 \oplus 1\rangle\Big) \\
&= \frac{1}{\sqrt{2^{n+1}}} \Big(\sum_{x \in X_0} |x, 0\rangle + \sum_{x \in X_1} |x, 1\rangle - \sum_{x \in X_0} |x, 1\rangle - \sum_{x \in X_1} |x, 0\rangle\Big) \\
&= \frac{1}{\sqrt{2^n}} \Big(\sum_{x \in X_0} |x\rangle - \sum_{x \in X_1} |x\rangle\Big) \otimes b
\end{aligned}$$

Thus the amplitude of the states in X_1 have been inverted as desired.

7.2 Structured Search

7.2.1 A Note on the Walsh-Hadamard Transform

There is another representation for the Walsh-Hadamard transformation of section 4.1.1 that is useful for understanding how to use the Walsh-Hadamard transformation in constructing quantum algorithms. The n bit Walsh-Hadamard transformation is a $2^n \times 2^n$ matrix W with entries W_{rs} where both r and s range from 0 to $2^n - 1$. We will show that

$$W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$$

where $r \cdot s$ is the number of common 1 bits in the the binary representations of r and s.

To see this equality, note that

$$W(|r\rangle) = \sum_s W_{rs}|s\rangle.$$

Let $r_{n-1} \ldots r_0$ be the binary representation of r, and $s_{n-1} \ldots s_0$ be the binary representation of s.

$$\begin{aligned}
W(|r\rangle) &= (H \otimes \ldots \otimes H)(|r_n - 1\rangle \otimes \ldots \otimes |r_0\rangle) \\
&= \frac{1}{\sqrt{2^n}}(|0\rangle + (-1)^{r_{n-1}}|1\rangle) \otimes \ldots \otimes (|0\rangle + (-1)^{r_0}|1\rangle) \\
&= \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{s_{n-1}r_{n-1}}|s_{n-1}\rangle \otimes \ldots \otimes (-1)^{s_0 r_0}|s_0\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{s \cdot r}|s\rangle.
\end{aligned}$$

7.2.2 Overview of Hogg's algorithms

A constraint satisfaction problem (CSP) has n variables $V = \{v_1, \ldots, v_n\}$ which can take m different values $X = \{x_1, \ldots, x_m\}$ subject to certain constraints $C_1, \ldots, C_l$. Solutions to a constraint satisfaction problem lie in the space of assignments of x_i's to v_j's, $V \times X$. There is a natural lattice structure on this space given by set containment. Figure 2 shows the assignment space and its lattice structure for $n = 2$, $m = 2$, $x_1 = 0$, and $x_2 = 1$. Note that the lattice includes both incomplete and inconsistent assignments.

Using the standard correspondence between sets of enumerated elements and binary sequences, in which a 1 in the nth place corresponds to inclusion of the nth element and a 0 corresponds to exclusion, standard basis vectors for a quantum state space can be put in one to one correspondence with the sets. For example, Figure 3 shows the lattice of Figure 2 rewritten in ket notation where

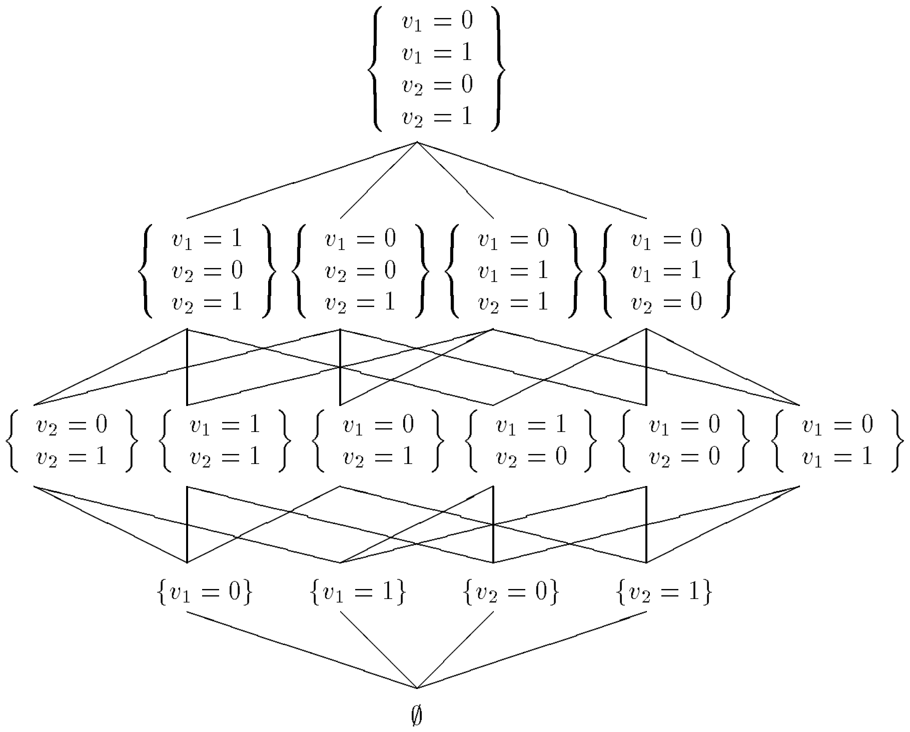


Figure 2: Lattive of variable assignments in a CSP

the elements $v_1 = 0$, $v_1 = 1$, $v_2 = 0$ and $v_2 = 1$ have been enumerated in that order.

If a state violates a constraint, then so do all states above it in the lattice. The approach Hogg takes in designing quantum algorithms for constraint satisfaction problems is to begin with all the amplitude concentrated in the $|0\ldots0\rangle$ state and to iteratively move amplitude up the lattice from sets to supersets and away from sets that violate the constraints. Note that this algorithm begins differently than Shor's algorithm and Grover's algorithm, which both begin by computing a function on a superposition of all the input values at once.

Hogg gives two ways [31, 32] of constructing a unitary matrix for moving amplitude up the lattice. We will describe both methods, and then describe how he moves amplitude away from bad sets.

Moving amplitude up: Method 1. There is an obvious matrix that moves amplitude from sets to supersets. Any amplitude associated to the empty set is evenly distributed between all sets with a single element. Any amplitude associated to a set with a single element is evenly distributed between all two element sets which contain that element and so on. For the lattice of a three

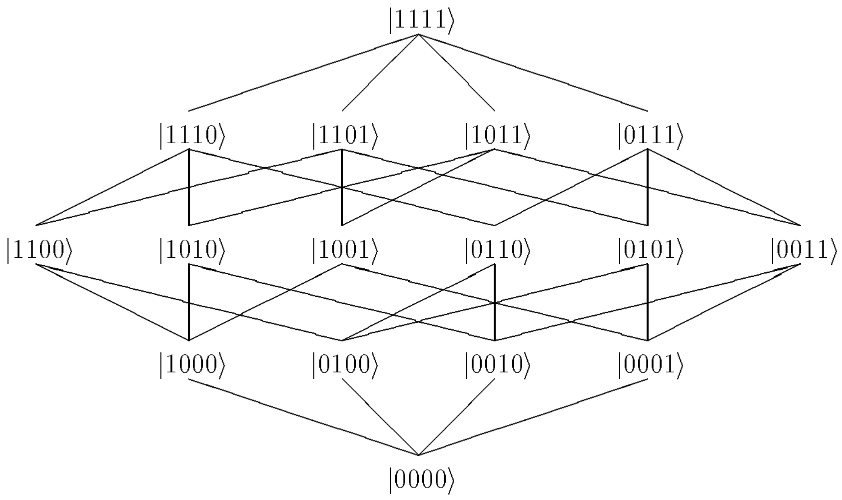
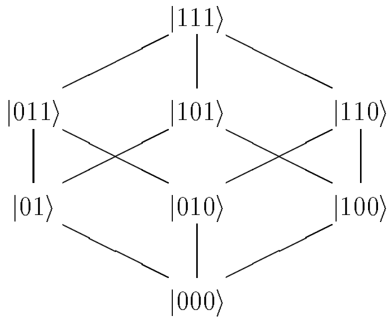


Figure 3: Lattice of variable assignments in ket form

element set



the matrix looks like

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Unfortunately this matrix is not unitary. It turns out [31] that the closest (in a suitable metric) unitary matrix U_M to an arbitrary matrix M can be found using M's singular value decomposition $M = UDV^T$ where D is a diagonal matrix, and U and V are unitary matrices. The product $U_M = UV^T$ gives the closest unitary matrix to M. Provided that U_M is sufficiently close to M, U_M will behave in a similar way to M and will therefore do a reasonably job of moving amplitude from sets to their supersets.

Moving amplitude up: Method 2. The second approach [32] uses the Walsh-Hadamard transformation. Hogg assumes that the desired matrix has form WDW where W is the Walsh-Hadamard transformation and D is a diagonal matrix whose entries depend only on the size of the sets. Hogg calculates the entries for D which maximize the movement of amplitude from a set to its supersets. This calculation exploits the property

$$W_{rs} = \frac{1}{\sqrt{N}}(-1)^{|r \cdot s|} = \frac{1}{\sqrt{N}}(-1)^{|r \cap s|}$$

shown in section 7.2.1.

Moving amplitude away from bad sets. To effect moving amplitude away from sets that violate the constraints, Hogg suggests adjusting the phases of the sets, depending on the extent to which they violate the constraints, in such a way that amplitude distributed to sets that have bad subsets cancels, where as the amplitude distributed to sets from all good subsets adds. There seems to be a variety of choices here, that will work more or less effectively depending on the particular problem. One choice he suggests is inverting the phase of all bad sets which will result in some cancelation in the amplitude of supersets between the amplitude coming from good subsets and bad subsets. This phase inversion can be done as in Grover's algorithm (7.1.2) using a predicate P that tests if a given state violates any of the constraints. Another suggestion is to give random phases to the bad sets so that on average the contribution to the amplitude of a superset from bad subsets is zero. Other choices are possible.

Because the canceling resulting from the phase changes varies from problem to problem, the probability of obtaining a solution is difficult to analyse. A few small experiments have been done and the guess is that the cost of the search still grows exponentially, but considerably more slowly than in the unstructured case. But until sufficiently large quantum computers are built, or better techniques for analyzing such algorithms are found, the efficiency cannot be determined for sure.

8 Quantum Error Correction

One fundamental problem in building quantum computers is the need to isolate the quantum state. An interaction of particles representing qubits with the external environment disturbs the quantum state, and causes it to decohere, or transform in a non-unitary fashion.

Steane [33] estimates that the decoherence of any system likely to be built is 10^7 times too large to be able to run Shor's algorithm as it stands on a 130 digit number. However, adding error correction algorithms to Shor's algorithm mitigates the effect of decoherence, making it again look possible that a system could be built on which Shor's algorithm could be run for large numbers.

On the surface quantum error correction is similar to classical error correcting codes in that redundant bits are used to detect and correct errors. But the situation for quantum error correction is somewhat more complicated than in the classical case since we are not dealing with binary data but with quantum states.

Quantum error correction must reconstruct the exact encoded quantum state. Given the impossibility of cloning or copying the quantum state, this reconstruction appears harder than in the classical case. However, it turns out that classical techniques can be modified to work for quantum systems.

8.1 Characterization of Errors

In the following it is assumed that all errors are the result of quantum interaction between the a set of qubits and the environment. The possible errors for each single qubit considered are linear combinations of no errors (I), bit flip errors (X), phase errors (Z), and bit flip phase errors (Y). A general single bit error is thus a transformation $e_1 I + e_2 X + e_3 Y + e_4 Z$. Interaction with the environment transforms single qubits according to

$$|\psi\rangle \to (e_1 I + e_2 X + e_3 Y + e_4 Z)|\psi\rangle = \sum_i e_i E_i |\psi\rangle.$$

For the general case of quantum registers, possible errors are expressed as linear combinations of unitary error operators E_i. These could be combinations of single bit errors, like tensor products of the single bit error transformations $\{I, X, Y, Z\}$, or more general multi-bit transformations. In any case, an error can be written as $\sum_i e_i E_i$ for some error operators E_i and coefficients e_i.

8.2 Recovery of Quantum State

An error correcting code for a set of errors E_i consists of a mapping C that embeds n data bits in $n + k$ code bits together with a syndrome extraction operators S_C that maps $n + k$ code bits to the set of indices of correctable errors E_i such that $i = S_C(E_i(C(x)))$. If $y = E_j(C(x))$ for some unknown but correctable error, then error $S_C(y)$ can be used to recover a properly encoded value $C(x)$, i.e. $E^{-1}_{S_C(y)}(y) = C(x)$.

Now consider the case of a quantum register. First, the state of the register can be in a superposition of basis vectors. Furthermore, the error can be a combination of correctable error operators E_i. It turns out that it is still possible to recover the encoded quantum state.

Given an error correcting code C with syndrome extraction operator S_C, an n-bit quantum state $|\psi\rangle$ is encoded in a $n+k$ bit quantum state $|\phi\rangle = C|\psi\rangle$. Assume that decoherence leads to an error state $\sum_i e_i E_i|\phi\rangle$ for some combination of correctable errors E_i. The original encoded state $|\phi\rangle$ can be recovered as follows:

1. Apply the syndrome extraction operator S_C to the quantum state padded with sufficient $|0\rangle$ bits:

$$S_C(\sum_i e_i E_i|\phi\rangle) \otimes |0\rangle = \sum_i e_i(E_i|\phi\rangle \otimes |i\rangle).$$

   Quantum parallelism gives a superposition of different errors each associated with their respective error index i.

2. Measure the $|i\rangle$ component of the result. This yields some (random) value i_0 and projects the state to

$$E_{i_0}|\phi, i_0\rangle$$

3. Apply the inverse error transformation $E_{i_0}^{-1}$ to the first $n+k$ qubits of $E_{i_0}|\phi, i_0\rangle$ to get the corrected state $|\phi\rangle$.

Note that step 2 projects a superposition of multiple error transformations into a single error. Consequently, only one inverse error transformation is required in step 3.

8.3 Error Correction Example

Consider the trivial error correcting code C that maps $|0\rangle \to |000\rangle$ and $|1\rangle \to |111\rangle$. C can correct single bit flip errors

$$E = \{I \otimes I \otimes I, X \otimes I \otimes I, I \otimes X \otimes I, I \otimes I \otimes X\}.$$

The syndrome extraction operator is

$$S : |x_0, x_1, x_2, 0, 0, 0\rangle \to |x_0, x_1, x_2, x_0 \,\mathrm{xor}\, x_1, x_0 \,\mathrm{xor}\, x_2, x_1 \,\mathrm{xor}\, x_2\rangle,$$

with the corresponding error correction operators shown in the table. Note that $E_i = E_i^{-1}$ for this example.

Bit flipped	Syndrome	Error correction
none	$\|000\rangle$	none
0	$\|110\rangle$	$X \otimes I \otimes I$
1	$\|101\rangle$	$I \otimes X \otimes I$
2	$\|011\rangle$	$I \otimes I \otimes X$

Consider the quantum bit $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ that is encoded as

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

and the error

$$E = \frac{4}{5}X \otimes I \otimes I + \frac{3}{5}I \otimes X \otimes I.$$

The resulting error state is

$$\begin{aligned}
E|\phi\rangle &= (\frac{4}{5}X \otimes I \otimes I + \frac{3}{5}I \otimes X \otimes I)(\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)) \\
&= \frac{4}{5}X \otimes I \otimes I(\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)) + \frac{3}{5}I \otimes X \otimes I(\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)) \\
&= \frac{4}{5\sqrt{2}}X \otimes I \otimes I(|000\rangle - |111\rangle) + \frac{3}{5\sqrt{2}}I \otimes X \otimes I(|000\rangle - |111\rangle) \\
&= \frac{4}{5\sqrt{2}}(|100\rangle - |011\rangle) + \frac{3}{5\sqrt{2}}(|010\rangle - |101\rangle)
\end{aligned}$$

Next apply the syndrome extraction to $(E|\phi\rangle) \otimes |000\rangle$ as follows:

$$\begin{aligned}
&S_C((E|\phi\rangle) \otimes |000\rangle) \\
&= S_C(\frac{4}{5\sqrt{2}}(|100000\rangle - |011000\rangle) + \frac{3}{5\sqrt{2}}(|010000\rangle - |101000\rangle)) \\
&= \frac{4}{5\sqrt{2}}(|100110\rangle - |011110\rangle) + \frac{3}{5\sqrt{2}}(|010101\rangle - |101101\rangle) \\
&= \frac{4}{5\sqrt{2}}(|100\rangle - |011\rangle) \otimes |110\rangle + \frac{3}{5\sqrt{2}}(|010\rangle - |101\rangle) \otimes |101\rangle
\end{aligned}$$

Measuring the last three bits of this state yields either $|110\rangle$ or $|101\rangle$. Assume the measurement produces the former, then the state becomes

$$\frac{1}{\sqrt{2}}(|100\rangle - |011\rangle) \otimes |110\rangle.$$

The measurement has the almost magical effect of causing all but one summand of the error to disappear. The remaining part of the error can be removed by applying the inverse error operator $X \otimes I \otimes I$, corresponding to the measured value $|110\rangle$, to the first three bits, to produce

$$\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) = C|\psi\rangle = |\phi\rangle.$$

9 Conclusions

Quantum computing is a new, emerging field that has the potential to dramatically change the way we think about computation, programming and complexity. The challenge for computer scientists and others is to develop new programming techniques appropriate for quantum computers. Quantum entanglement and phase cancellation introduce a new dimension to computation. Programming no longer consists of merely formulating step-by-step algorithms but requires new techniques of adjusting phases, and mixing and diffusing amplitudes to extract useful output.

We have tried to give an accurate account of the state-of-the-art of quantum computing for computer scientists and other non-physicists. We have described some of the quantum mechanical effects, like the exponential state space, the entangled states, and the linearity of quantum state transformations, that make quantum parallelism possible. Even though quantum computations must be linear and reversible, any classical algorithm can be implemented on a quantum computer. But the real power of these new machines, the exponential parallelism, can only be exploited using new, innovative programming techniques. People have only recently begun to research such techniques.

We have described Shor's polynomial-time factorization algorithm that has stimulated the field of quantum computing. Given a practical quantum computer, Shor's algorithm would make many present cryptographic methods obsolete. Grover's search algorithm, while only providing a polynomial speedup, proves that quantum computers are strictly more powerful than classical ones. Even though Grover's algorithm has been shown to be optimal, there is hope that faster algorithms can be found by exploiting properties of the problem structure. We have described one such approach taken by Hogg.

There are a few other known quantum algorithms that we did not discuss. Jones and Mosca ([34]) describe the implementation on a 2-bit quantum computer of a constant time algorithm [35] that can distinguish whether a function is balanced or constant. Grover ([36]) describes an efficient algorithm for estimating the median of a set of values and Terhal and Smolin ([37]) can solve the coin weighing problem in a single step.

Beyond these algorithms not much more is known about what could be done with a practical quantum computer. It is an open question whether or not $P = NP$ on quantum computers. There is some speculation among physicists that quantum transformations might be slightly non-linear. So far all experiments that have been done are consistent with the standard linear quantum mechanics, but a slight non-linearity is still possible. Abrams and Lloyd ([38]) show that even a very slight non-linearity could be exploited to solve all NP hard problem on a quantum computer in polynomial time. This result further highlights the fact that computation is fundamentally a physical process, and that what can be computed may be dependent on subtle issues in physics.

Of course, there are daunting physical problems that must be overcome if anyone is ever to build a useful quantum computer. Decoherence, the distortion of the quantum state due to interaction with the environment, is a key problem. A big breakthrough for dealing with decoherence came from the algorithmic, rather than the physical, side of the field with the development of quantum error correction techniques. We have described some of the principles involved. Further advances in quantum error correction and the development of robust algorithms will be as important as advances in the hardware side for quantum computers to become useful.

9.1 Further Reading

Andrew Steane's survey article "Quantum computing" [33] is aimed at physicists. We found it long on the classical theory of computation, and short on quantum mechanics. We hope that after having read the present paper, readers will find Steane's article an easy read. We recommend reading his paper for his viewpoint on this subject, particularly for his description of connections between information theory and quantum computing and for his discussion of error correction, of which he was one of the main developers. He also has an overview of the physics involved in actually building quantum computers, and a survey of what had been done up to July 1997. His article contains a more detailed history of the ideas related to quantum computing than the present paper, and has more references as well.

Richard Feynman's *Lectures on Computation* [18] contains a reprint of the lecture "Quantum Mechanical Computers" [39] which began the whole field. It also discusses the thermodynamics of computations which is closely tied with reversible computing and information theory.

Colin Williams and Scott Clearwater's book *Explorations in Quantum Computing* [40] comes with software in the form of Mathematica notebooks that simulate some quantum algorithms like Shor's algorithm.

The second half of the October 1997 issue of the SIAM Journal of Computing contains six seminal articles on quantum computing [28] [23] [3] [25].

Most of the articles referenced in this paper, and many more, can be found at the Los Alamos preprint server: `http://xxx.lanl.gov/archive/quant-ph`. Lots of other interesting information about quantum computing can be found on the web. One good place to start would be the Stanford-Berkeley-MIT-IBM Quantum Computation Research Project's web pages at `http://feynman.stanford.edu/qcomp/` which, in addition to having a fair amount of information about quantum computing, have a lot of links to other sites of interest.

References

[1] R. Feynman, International Journal of Theoretical Physics 21 (1982) 467.

[2] P.W. Shor, Proceedings of the 35th Annual Symposium on Foundations of Computer Science, pp. 124–134, Institute of Electrical and Electronic Engineers Computer Society Press, 1994, `ftp://netlib.att.com/netlib/att/math/shor/quantum.algorithms.ps.Z`.

[3] P.W. Shor, Society for Industrial and Applied Mathematics Journal on Computing 26 (1997) 1484, Expanded version of [2].

[4] J.I. Cirac and P. Zoller, Physical Review Letters 74 (1995) 4091.

[5] A. Steane, The ion trap quantum information processor, 1996, quant-ph/9608011.

[6] L.J. Schulman and U. Vazirani, Scalable NMR quantum computation, 1998, quant-ph/9804060.

[7] N.A. Gershenfeld and I.L. Chuang, Science 275 (1997) 350.

[8] R. Laflamme et al., NMR GHZ, 1997, quant-ph/9709025.

[9] R. Feynman, The Feynman Lectures on Physics, Vol. III (Addison-Wesley, Reading, Mass, 1965).

[10] G. Greenstein and A.G. Zajonc, The Quantum Challenge (Jones and Bartlett Publishers, Sudbury, Mass, 1997).

[11] R.L. Liboff, Introductory Quantum Mechanics (3rd edition) (Addison-Wesley, Reading, Mass, 1997).

[12] P. Dirac, The Principles of Quantum Mechanics, 4th ed. (Oxford University Press, 1958).

[13] C.H. Bennett and G. Brassard, SIGACTN: SIGACT News (ACM Special Interest Group on Automata and Computability Theory) 18 (1987).

[14] C.H. Bennett, G. Brassard and A.K. Ekert, Scientific American 267 (1992) 50.

[15] A.K. Ekert et al., Physical Review Letters 69 (1992).

[16] C.H. Bennett, Physical Review Letters 68 (1992).

[17] R.J. Hughes et al., Photonic Quantum Computing, edited by S.P. Hotaling and A.R. Pirich Vol. 3076, pp. 2–11, 1997.

[18] R. Feynman, Feynman lectures on computation, 1996.

[19] W.K. Wootters and W.H. Zurek, Nature 299 (1982) 802.

[20] D. Bouwmeester et al., Nature 390 (1997) 575.

[21] V. Vedral, A. Barenco and A.K. Ekert, Quantum networks for elementary arithmetic operations, Physical Review A, 1996, quant-ph/9511018.

[22] D. Deutsch, Proceedings of the Royal Society of London Ser. A A400 (1985) 97.

[23] E. Bernstein and U.V. Vazirani, Society for Industrial and Applied Mathematics Journal on Computing 26 (1997) 1411, A preliminary version of this paper appeared in the Proceedings of the 25th Association for Computing Machinery Symposium on the Theory of Computing.

[24] A. Barenco et al., Physical Review A 52 (1995) 3457, quant-ph/9503016,

[25] D.R. Simon, Society for Industrial and Applied Mathematics Journal on Computing 26 (1997) 1474, A preliminary version of this paper appeared in the Proceedings of the 35th Annual Symposium on Foundations of Computer Science.

[26] A. Lenstra and H. Lenstra, editors, The Development of the Number Field Sieve, Lecture Notes in Mathematics Vol. 1554 (Springer Verlag, 1993).

[27] L.K. Grover, Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, pp. 212–219, Philadelphia, Pennsylvania, 1996.

[28] C.H. Bennett et al., Society for Industrial and Applied Mathematics Journal on Computing 26 (1997) 1510, quant-ph/9701001.

[29] M. Boyer et al., Proceedings of the Workshop on Physics of Computation: PhysComp '96, Los Alamitos, CA, 1996, Institute of Electrical and Electronic Engineers Computer Society Press, quant-ph/9605034.

[30] C. Zalka, Grover's quantum searching algorithm is optimal, 1997, quant-ph/9711070.

[31] T. Hogg, Journal of Artificial Intelligence Research 4 (1996) 91, quant-ph/9508012.

[32] T. Hogg, Physical Review Letters 80 (1998) 2473, quant-ph/9508012.

[33] A. Steane, Reports on Progress in Physics 61 (1998) 117, quant-ph/9708022.

[34] J.A. Jones and M. Mosca, Journal of Chemical Physics 109 (1998) 1648, quant-ph/9801027.

[35] D. Deutsch and R. Jozsa, Proceedings of the Royal Society of London Ser. A A439 (1992) 553.

[36] L.K. Grover, Proceedings of the 30th annual ACM symposium on the theory of computing (1998) 53, quant-ph/9711043.

[37] B.M. Terhal and J.A. Smolin, Single quantum querying of a database, 1997, quant-ph/9705041.

[38] D.S. Abrams and S. Lloyd, Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and #p problems, 1998, quant-ph/9801041.

[39] R. Feynman, Optics News 11 (1985), Also in *Foundations of Physics*, 16(6):507–531, 1986.

[40] C.P. Williams and S.H. Clearwater, Explorations in Quantum Computing (Telos, Springer-Verlag, 1998).

[41] T.A. Hungerford, Algebra (Springer Verlag, New York, Heidelberg, Berlin, 1974).

[42] G.H. Hardy and E.M. Wright, An Introduction to the Theory of Numbers (Oxford University Press, 1979).

A Tensor Products

The tensor product ($\otimes$) of a n dimensional and a k dimensional vector is a nk dimensional vector. Similarly, if A and B are transformations on n dimensional and a k dimensional vectors respectively, then $A \otimes B$[5] is a transformation on nk dimensional vectors.

The exact mathematical details of tensor products are beyond the scope of this paper (see [41] for a comprehensive treatment). For our purposes the following algebraic rules are sufficient to calculate with tensor products. For matrices A,B,C,D, U, vectors u, x, y, and scalars a, b the following hold:

$$\begin{aligned}
(A \otimes B)(C \otimes D) &= AC \otimes BD \\
(A \otimes B)(x \otimes y) &= Ax \otimes By \\
(x + y) \otimes u &= x \otimes u + y \otimes u \\
u \otimes (x + y) &= u \otimes x + u \otimes y \\
ax \otimes by &= ab(x \otimes y)
\end{aligned}$$

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \otimes U = \begin{pmatrix} A \otimes U & B \otimes U \\ C \otimes U & D \otimes U \end{pmatrix},$$

[5] Technically, this is a right Kronecker product.

which specialized for scalars a, b, c, d to

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \otimes U = \begin{pmatrix} aU & bU \\ cU & dU \end{pmatrix}.$$

The conjugate transpose distributes over tensor products, i.e.

$$(A \otimes B)^* = A^* \otimes B^*.$$

A matrix U is *unitary* if its conjugate transpose its inverse: $U^*U = I$.

The tensor product of several matrices is unitary if and only if each one of the matrices is unitary up to a constant. Let $U = A_1 \otimes A_2 \otimes \ldots \otimes A_n$. Then U is unitary if $A_i^* A_i = k_i I$ and $\Pi_i k_i = 1$.

$$\begin{aligned} U^*U &= (A_1^* \otimes A_2^* \otimes \ldots \otimes A_n^*)(A_1 \otimes A_2 \otimes \ldots \otimes A_n) \\ &= A_1^*A_1 \otimes A_2^*A_2 \otimes \ldots \otimes A_n^*A_n \\ &= k_1 I \otimes \ldots k_n I \\ &= I \end{aligned}$$

For example, the distributive law allows computations of the form:

$$\begin{aligned} & (a_0|0\rangle + b_0|1\rangle) \otimes (a_1|0\rangle + b_1|1\rangle) \\ = \; & (a_0|0\rangle \otimes a_1|0\rangle) + (b_0|1\rangle \otimes a_1|0\rangle) + (a_0|0\rangle \otimes b_1|1\rangle) + (b_0|1\rangle \otimes b_1|1\rangle) \\ = \; & a_0a_1((|0\rangle \otimes |0\rangle) + b_0a_1(|1\rangle \otimes |0\rangle) + a_0b_1(|0\rangle \otimes |1\rangle) + b_0b_1(|1\rangle \otimes |1\rangle) \\ = \; & a_0a_1(|00\rangle + b_0a_1|10\rangle + a_0b_1|01\rangle + b_0b_1|11\rangle \end{aligned}$$

B Continued fractions and extracting the period from the measurement in Shor's algorithm

In the general case where the period r does not divide 2^m, the value v measured in step 4 of Shor's algorithm will be, with high probability, close to some multiple of $\frac{2^m}{r}$, say $j\frac{2^m}{r}$.

The aim is to extract the period r from the measured value v. Shor shows that, with high probability, v is within $\frac{1}{2}$ of some $j\frac{2^m}{r}$. Thus

$$\left| v - j\frac{2^m}{r} \right| < \frac{1}{2}$$

for some j, which implies that

$$\left| \frac{v}{2^m} - \frac{j}{r} \right| < \frac{1}{2 \cdot 2^m} < \frac{1}{2M^2}.$$

The difference between two distinct fractions $\frac{p}{q}$ and $\frac{p'}{q'}$ with denominators less than M is bounded

$$\left|\frac{p}{q} - \frac{p'}{q'}\right| = \left|\frac{pq' - p'q}{qq'}\right| > \frac{1}{M^2}.$$

Thus there is at most one fraction $\frac{p}{q}$ with denominator $q < M$ such that $\left|\frac{v}{2^m} - \frac{p}{q}\right| < \frac{1}{M^2}$. In the high probability case that v is within $\frac{1}{2}$ of $j\frac{2^m}{r}$, this fraction will be $\frac{j}{r}$.

The unique fraction with denominator less than M that is within $\frac{1}{M^2}$ of $\frac{v}{2^m}$ can be obtained efficiently from the continued fraction expansion of $\frac{v}{2^m}$ as follows. Using the sequences

$$\begin{aligned}
a_0 &= \left[\frac{v}{2^m}\right] \\
\epsilon_0 &= \frac{v}{2^m} - a_0 \\
a_n &= \left[\frac{1}{\epsilon_{n-1}}\right] \\
\epsilon_n &= \frac{1}{\epsilon_{n-1}} - a_n \\
p_0 &= a_0 \\
p_1 &= a_1 a_0 + 1 \\
p_n &= a_n p_{n-1} + p_{n-2} \\
q_0 &= 1 \\
q_1 &= a_1 \\
q_n &= a_n q_{n-1} + q_{n-2}
\end{aligned}$$

compute the first fraction $\frac{p_n}{q_n}$ such that $q_n < M \leq q_{n+1}$. See any standard number theory text, like Hardy and Wright [42], for why this procedure works.

In the high probability case when $\frac{v}{2^m}$ is within $\frac{1}{M^2}$ of a multiple $\frac{j}{r}$ of $\frac{1}{r}$, the fraction obtained from the above procedure is $\frac{j}{r}$ as it has denominator less than M. We take the denominator q of the obtained fraction as our guess for the period, which will work when j and r are relatively prime.