An Introduction to Quantum Computing for Non-Physicists

ELEANOR RIEFFEL
FX Palo Alto Laboratory
AND
WOLFGANG POLAK

Richard Feynman's observation that certain quantum mechanical effects cannot be simulated efficiently on a computer led to speculation that computation in general could be done more efficiently if it used these quantum effects. This speculation proved justified when Peter Shor described a polynomial time quantum algorithm for factoring integers.

In quantum systems, the computational space increases exponentially with the size of the system, which enables exponential parallelism. This parallelism could lead to exponentially faster quantum algorithms than possible classically. The catch is that accessing the results, which requires measurement, proves tricky and requires new nontraditional programming techniques.

The aim of this paper is to guide computer scientists through the barriers that separate quantum computing from conventional computing. We introduce basic principles of quantum mechanics to explain where the power of quantum computers comes from and why it is difficult to harness. We describe quantum cryptography, teleportation, and dense coding. Various approaches to exploiting the power of quantum parallelism are explained. We conclude with a discussion of quantum error correction.

Categories and Subject Descriptors: A.1 [**Introductory and Survey**]

General Terms: Algorithms, Security, Theory

Additional Key Words and Phrases: Quantum computing, complexity, parallelism

1. INTRODUCTION

Richard Feynman observed in the early 1980s [Feynman 1982] 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 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 [Shor 1994; 1997], that the field of quantum computing came into its own. This discovery prompted a flurry of activity among experimentalists trying to build quantum computers and theoreticians trying to find other quantum algorithms.

Authors' address: E. Rieffel, FX Palo Alto Laboratory, 3400 Hillview Av., Palo Alto, CA 94304; W. Polak, Consultant.

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 3-bit quantum computer.

The aim of this paper is to guide computer scientists and other nonphysicists 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.

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 only a linear increase in the amount of physical space needed. This effect is called quantum parallelism [Deutsch and Jozsa 1992].

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 finessing the measurement problem to exploit the power of quantum parallelism. This sort of manipulation has no classical analog and requires nontraditional programming techniques. One technique manipulates the quantum state so that a common property of all of the output values such as the symmetry or period of a function can be read off. This technique is used in Shor's factorization algorithm. Another technique transforms the quantum state to increase the likelihood that output of interest will be read. Grover's search algorithm makes use of such an amplification technique. This paper describes quantum parallelism in detail, and the techniques currently known for harnessing its power.

Section 2, following this introduction, explains of the basic concepts of quantum mechanics that are important for quantum computation. This section cannot give a comprehensive view of quantum mechanics. Our aim is to provide the reader with tools in the form of mathematics and notation with which to work with the quantum mechanics involved in quantum computation. We hope that this paper will equip readers 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 describe 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 "extra" states that have no classical analog and lead to the exponential size of the quantum state space are the entangled states, like the state leading to the famous EPR[1] paradox (see Section 3.4).

We discuss the two types of operations a quantum system can undergo:

[1] EPR = Einstein, Podolsky, and Rosen

measurement and quantum state transformations. Most quantum algorithms involve a sequence of quantum state transformations followed by a measurement. For classical computers there are sets of gates that are universal in the sense that any classical computation can be performed using a sequence of these gates. Similarly, there are sets of primitive quantum state 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 state transformations, and therefore all quantum gates and all quantum computations, must be reversible. Yet all classical algorithms can be made reversible and can be computed on a quantum computer in comparable time. 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 a 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 not obvious 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 of the 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. Sections 6 and 7 describe techniques for taking advantage of quantum parallelism inspite of the severe constraints imposed by quantum mechanics on what can be measured.

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 is no classical polynomial time factoring algorithm. 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 $O(n)$, 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 using ion traps, nuclear magnetic resonance (NMR), and optical and solid-state techniques. All of the current proposals have

scaling problems, so a breakthrough will be needed to go beyond tens of qubits to hundreds of qubits. While both optical and solid-state techniques show promise, NMR and ion trap technologies are the most advanced so far.

In an ion trap quantum computer [Circ and Zoller 1995; Steane 1996] a linear sequence of ions representing the qubits are confined by electric fields. Lasers are directed at individual ions 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 2-bit operation. The approach requires that the ions be kept in extreme vacuum and at extremely low temperatures.

The NMR approach has the advantage that it will work at room temperature and that NMR technology in general is already fairly advanced. 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 [Schulman and Vazirani 1998] has been made that may overcome this problem. NMR computers with three qubits have been built successfully [Cory et al. 1998; Gershenfeld and Chuang 1997; Laflamme et al. 1997; Vandersypen et al. 1999]. 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. Initially people thought quantum error correction might be impossible because of the impossibility of reliably copying unknown quantum states, but it turns out that it is possible to design quantum error correcting codes that detect certain kinds of errors and enable the reconstruction of the exact error-free quantum state. Quantum error correction is discussed in Section 8.

Appendices provide background information on tensor products and continued fractions.

2. QUANTUM MECHANICS

Quantum mechanical 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 Feynman et al. [1965], Liboff [1997], and Greenstein and Zajonc [1997] for expositions of quantum mechanics). 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 of 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 needed for quantum computation.

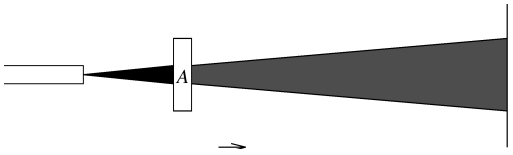
2.1. Photon Polarization

Photons are the only particles that we can observe directly. The following simple

experiment can be performed with minimal equipment: a strong light source, such as a laser pointer, and three polaroids (polarization filters), which can be picked up at any camera supply store. The experiment demonstrates some of the principles of quantum mechanics through photons and their polarization.

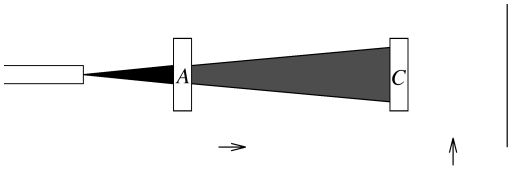
2.1.1 The Experiment. A beam of light shines on a projection screen. Filters A, B, and C are polarized horizontally, at 45°, and vertically, respectively, and can be placed so as to intersect the beam of light.

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.

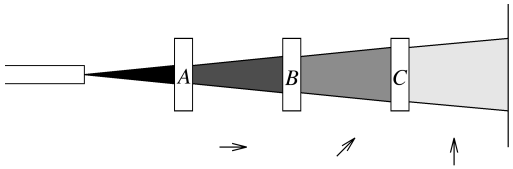


The function of filter A cannot be explained as a "sieve" that only lets those photons pass that happen to be already horizontally polarized. If that were the case, few of the randomly polarized incoming electrons would be horizontally polarized, so we would expect a much larger attenuation of the light 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 a nonintuitive effect. Classical experience suggests that adding a filter should only be able to decrease the number of photons getting through. How can it increase it?

2.1.2 The Explanation. A photon's polarization state can be modeled 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 basis for this representation is completely arbitrary: any two orthogonal unit vectors will do (e.g., $\{|\nwarrow\rangle, |\nearrow\rangle\}$).

The measurement postulate of quantum mechanics states that any device measuring a two-dimensional system has an associated orthonormal basis with respect to which the quantum measurement takes place. Measurement of a state transforms the state into one of the measuring device's associated basis vectors. The probability that the state is measured as basis vector $|u\rangle$ is the square of the norm of the amplitude of the component of the original state in the direction of the basis vector $|u\rangle$. For example, given a device

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

[3] Imaginary coefficients correspond to circular polarization.

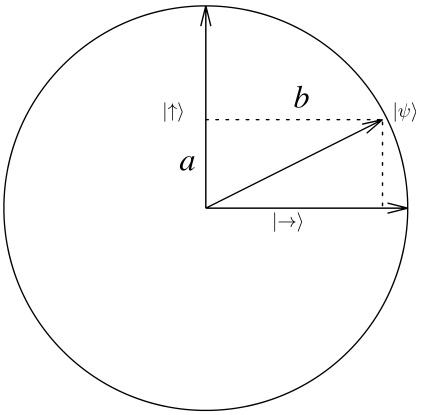


Fig. 1. Measurement is a projection onto the basis.

for measuring the polarization of photons with associated basis $\{|\uparrow\rangle, |\rightarrow\rangle\}$, the state $|\psi\rangle = a|\uparrow\rangle + b|\rightarrow\rangle$ is measured as $|\uparrow\rangle$ with probability $|a|^2$ and as $|\rightarrow\rangle$ with probability $|b|^2$ (see Figure 1). Note that different measuring devices will have different associated bases, and measurements using these devices will have different outcomes. 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.

Furthermore, measurement of the quantum state will change the state to the result of the measurement. That is, if measurement of $|\psi\rangle = a|\uparrow\rangle + b|\rightarrow\rangle$ results in $|\uparrow\rangle$, then the state ψ changes to $|\uparrow\rangle$ and a second measurement 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 determine 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 together with a vector orthogonal to its polarization. The photons that, after being measured by the filter, match the filter's polarization are let through. The others are reflected and now 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 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

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

which we write as $\{|\nearrow\rangle, |\nwarrow\rangle\}$. Note that $|\rightarrow\rangle = \frac{1}{\sqrt{2}}(|\nearrow\rangle - |\nwarrow\rangle)$ and $|\uparrow\rangle = \frac{1}{\sqrt{2}}(|\nearrow\rangle + |\nwarrow\rangle)$. 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.

2.2. State Spaces and Bra/Ket Notation

The state space of a quantum system, consisting of the positions, momentums, polarizations, spins, and so on of the various particles, is modeled by a Hilbert space of wave functions. We will not look at the details of these wave functions. For quantum computing we need only deal with finite quantum systems and it suffices to consider finite dimensional complex vector spaces with an inner product that are spanned by abstract wave functions such as $|\rightarrow\rangle$.

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 invented by Dirac [1958]. 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 $\{|0\rangle, |1\rangle\}$ can be expressed as $\{(1,0)^T, (0,1)^T\}$. 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 $\langle x|$ and $|y\rangle$ 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

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

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 prefer the slightly more intuitive notation

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

which explicitly specifies the result of a transformation on 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. 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 will be made with respect to the standard basis for quantum computation, $\{|0\rangle, |1\rangle\}$.

For the purposes of quantum computation, the basis states $|0\rangle$ and $|1\rangle$ are taken to represent 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 $|b|^2$.

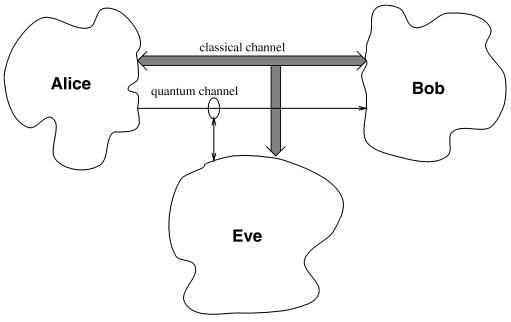
Even though a quantum bit can be put in infinitely many superposition states, it is only possible to extract a single classical bit's worth of information from a single quantum bit. The reason that no more information can be gained from a qubit than in a classical bit is that information can only be obtained 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 can result in only one of two states, one of the basis vectors associated to the given measuring device, so, just as in the classical case, there are only two possible results. As measurement changes the state, one cannot measure the state of a

qubit in two different bases. 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 ways, even indirectly by, say, copying the qubit and measuring the copy in a different basis from the original.

3.1. Quantum Key Distribution

Sequences of single qubits can be used to transmit private keys on insecure channels. In 1984 Bennett and Brassard described the first quantum key distribution scheme [Bennett and Brassard 1987; Bennett et al. 1992]. Classically, public key encryption techniques (e.g., RSA) are used for key distribution.

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 bidirectional open channel and a unidirectional quantum channel, both of which can be observed by Eve, who wishes to eavesdrop on their conversation. This situation is illustrated in the figure that follows. 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:

$$0 \rightarrow |\uparrow\rangle$$
$$1 \rightarrow |\rightarrow\rangle$$

or

$$0 \rightarrow |\nwarrow\rangle$$
$$1 \rightarrow |\nearrow\rangle.$$

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 the 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 re-sends 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 re-send the bit with the wrong basis. So when Bob measures a re-sent 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 et al. [1992], Bennett [1992], and Lo and Chau [1999]. But none of the quantum key distribution techniques are substitutes for public key encryption schemes. Attacks by eavesdroppers other than the one described here are possible. Security against all such schemes is discussed in both Mayers [1998] and Lo and Chau [1999].

Quantum key distribution has been realized over a distance of 24 km using standard fiber optical cables [Hughes et al.

1997] and over 0.5 km through the atmosphere [Hughes et al. 1999].

3.2. Multiple Qubits

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. 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 can be represented by 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 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 combine classically 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 are 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 2 two-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.[5] 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 $|b_n b_{n-1} \ldots b_0\rangle$ where b_i are the binary digits of the number x.

A basis for a 3-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 $dim(X) \times dim(Y)$.

The state $|00\rangle + |11\rangle$ is an example of a quantum state that cannot be described in terms of the state of each of its components (qubits) separately. 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

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

and $a_1 b_2 = 0$ implies that either $a_1 a_2 = 0$ or $b_1 b_2 = 0$. States that 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.

[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 $|a|^2 + |b|^2 = 1$.

[5] It is only when we use matrix notation to describe state transformations that the order of basis vectors becomes relevant.

Note that it would require vast resources to simulate even a small quantum system on traditional computers. The evolution of quantum systems is exponentially faster than their classical simulations. 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

The experiment in Section 2.1.2 illustrates how measurement of a single qubit projects the quantum state on to one of the basis states associated with the measuring device. The result of a measurement is probabilistic and the process of measurement changes the state to that measured.

Let us look at an example of measurement in a two-qubit system. Any two-qubit state 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 $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$. Suppose we wish to measure the first qubit with respect to the standard basis $\{|0\rangle, |1\rangle\}$. For convenience we will rewrite the state as follows:

$$\begin{aligned} &a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle \\ &\quad = |0\rangle \otimes (a|0\rangle + b|1\rangle) + |1\rangle \otimes (c|0\rangle + d|1\rangle) \\ &\quad = u|0\rangle \otimes (a/u|0\rangle + b/u|1\rangle) + v|1\rangle \\ &\qquad \otimes (c/v|0\rangle + d/v|1\rangle). \end{aligned}$$

For $u = \sqrt{|a|^2 + |b|^2}$ and $v = \sqrt{|c|^2 + |d|^2}$ the vectors $a/u|0\rangle + b/u|1\rangle$ and $c/v|0\rangle + d/v|1\rangle$ are of unit length. Once the state has been rewritten as above, as a tensor product of the bit being measured and a second vector of unit length, the probabalistic result of a measurement is easy to read off. Measurement of the first bit will with probability $u^2 = |a|^2 + |b|^2$ return $|0\rangle$, projecting the state to $|0\rangle \otimes (a/u|0\rangle + b/u|1\rangle)$, or with probability $v = |c|^2 + |d|^2$ yield $|1\rangle$, projecting the state to $|1\rangle \otimes (c/v|0\rangle + d/v|1\rangle)$. As $|0\rangle \otimes (a/u|0\rangle + b/u|1\rangle)$ and $|1\rangle \otimes (c/v|0\rangle + d/v|1\rangle)$ are both unit vectors, no scaling is necessary. Measuring the second bit works similarly.

For the purposes of quantum computation, multibit measurement can be treated as a series of single-bit measurements in the standard basis. Other sorts of measurements are possible, such as measuring whether two qubits have the same value without learning the actual value of the two qubits. But such measurements are equivalent to unitary transformations followed by a standard measurement of individual qubits, and so it suffices to look only at standard measurements.

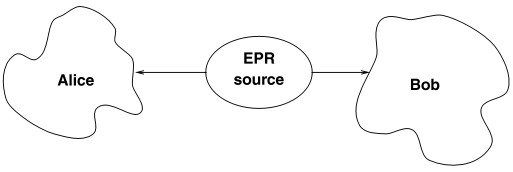
In the two-qubit example, the state space is a cartesian product of the subspace consisting of all states whose first qubit is in the state $|0\rangle$ and the orthogonal subspace of states whose first qubit is in the state $|1\rangle$. Any quantum state can be written as the sum of two vectors, one in each of the subspaces. A measurement of k qubits in the standard basis has 2^k possible outcomes m_i. Any device measuring k qubits of an n-qubit system splits of the 2^n-dimensional state space $\mathcal{H}$ into a cartesian product of orthogonal subspaces $S_1, \ldots, S_{2^k}$ with $\mathcal{H} = S_1 \times \cdots \times S_{2^k}$, such that the value of the k qubits being measured is m_i and the state after measurement is in space the space S_i for some i. The device randomly chooses one of the S_i's, with probability the square of the amplitude of the component of ψ in S_i, and projects the state into that component, scaling to give length 1. Equivalently, 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 basis vectors compatible with that value of the measurement.

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 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 result 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 of 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 Greenstein and Zajonc [1997] 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 experimental results 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 change 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 showing that all classical computations can be done reversibly. See Feynman's *Lectures on Computation* [Feynman 1996] for an account of reversible computation and its relation to the energy of computation and information.

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, with $\{|0\rangle, |1\rangle\}$ as the preferred ordered basis, is also shown.

$$\begin{aligned}
I &: |0\rangle \rightarrow |0\rangle \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
&\quad |1\rangle \rightarrow |1\rangle \\
X &: |0\rangle \rightarrow |1\rangle \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
&\quad |1\rangle \rightarrow |0\rangle \\
Y &: |0\rangle \rightarrow -|1\rangle \quad \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \\
&\quad |1\rangle \rightarrow |0\rangle \\
Z &: |0\rangle \rightarrow |0\rangle \quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\
&\quad |1\rangle \rightarrow -|1\rangle
\end{aligned}$$

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

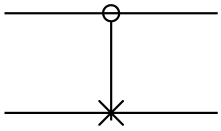
$$YY^* = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = I.$$

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. The vectors $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ form an orthonormal basis for the state space of a two-qubit system, a four-dimensional complex vector space. In order to represent transformations of this space in matrix notation we need to choose an isomorphism between this space and the space of complex 4-tuples. There is no reason, other than convention, to pick one isomorphism over another. The one we use here associates $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ to the standard 4-tuple basis $(1, 0, 0, 0)^T$, $(0, 1, 0, 0)^T$, $(0, 0, 1, 0)^T$, and $(0, 0, 0, 1)^T$, in that order. The C_{not} transformation has representations

$$C_{not}: \begin{array}{ccc} |00\rangle & \rightarrow & |00\rangle \\ |01\rangle & \rightarrow & |01\rangle \\ |10\rangle & \rightarrow & |11\rangle \\ |11\rangle & \rightarrow & |10\rangle \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

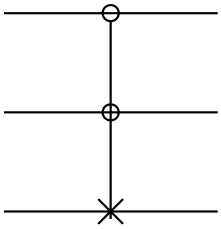
The transformation C_{not} is unitary since $C_{not}^* = C_{not}$ and $C_{not}C_{not} = I$. The C_{not} gate cannot be decomposed into a tensor product of two single-bit transformations.

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

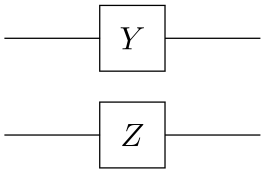


The open circle indicates the control bit, and the $\times$ indicates the conditional negation of the subject bit. In general there can be multiple control bits. Some authors use a solid circle to indicate negative control, in which the subject bit is toggled when the control bit is 0.

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 labeled boxes as shown.



4.1.1 The Walsh–Hadamard Transformation. Another important single-bit transformation is the Hadamard transformation, defined by

$$\begin{aligned} H: |0\rangle &\rightarrow \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle &\rightarrow \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned}$$

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 representation of the numbers from 0 to $2^n - 1$.

$$\begin{aligned} &(H \otimes H \otimes \cdots \otimes H)|00\ldots0\rangle \\ &\quad = \frac{1}{\sqrt{2^n}}((|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \\ &\qquad \otimes \cdots \otimes (|0\rangle + |1\rangle)) \\ &\quad = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle. \end{aligned}$$

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 states cannot be copied or cloned. The no cloning proof given here, originally due to Wootters and Zurek [1982], 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,

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

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 clear that cloning is not possible by using measurement, since measurement is both probabalistic and destructive of states not in the measuring device's associated subspaces.

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 $a|00\ldots0\rangle + b|11\ldots1\rangle$ 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 $\{|00\ldots0\rangle, |00\ldots01\rangle, \ldots, |11\ldots1\rangle\}$, 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.

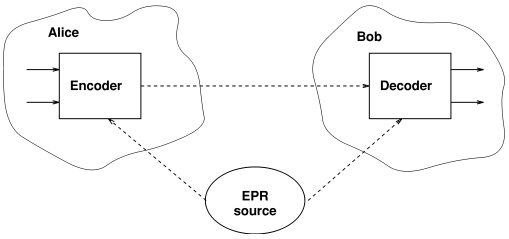
Dense coding uses one quantum bit together with an EPR pair to encode and transmit two classical bits. Since EPR pairs can be distributed ahead of time, only one qubit (particle) needs to be physically transmitted to communicate two bits of information. This result is surprising since, as was discussed in Section 3, only one classical bit's worth of information can be extracted from a qubit. 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 of 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. 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
$\psi_0 = \frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|00\rangle + \|10\rangle)$	$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$	$\|0\rangle$
$\psi_1 = \frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$	$\frac{1}{\sqrt{2}}(\|11\rangle + \|01\rangle)$	$\frac{1}{\sqrt{2}}(\|1\rangle + \|0\rangle)$	$\|1\rangle$
$\psi_2 = \frac{1}{\sqrt{2}}(-\|10\rangle + \|01\rangle)$	$\frac{1}{\sqrt{2}}(-\|11\rangle + \|01\rangle)$	$\frac{1}{\sqrt{2}}(-\|1\rangle + \|0\rangle)$	$\|1\rangle$
$\psi_3 = \frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	$\frac{1}{\sqrt{2}}(\|00\rangle - \|10\rangle)$	$\frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$	$\|0\rangle$

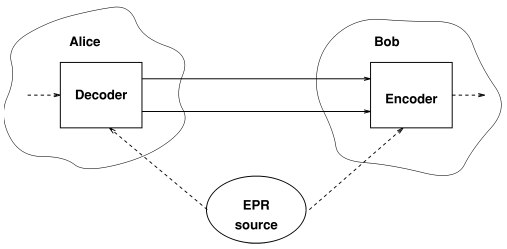
Note that Bob can now measure the second qubit without disturbing the quantum state. If the measurement returns $|0\rangle$ then the encoded value was either 0 or 3, if the measurement returns $|1\rangle$ then the encoded value was either 1 or 2.

Bob now applies H to the first bit:

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

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 copied, the quantum state of the given particle will necessarily be destroyed. Single-bit teleportation has been realized experimentally [Boschi et al. 1998; Bouwmeester et al. 1997; Nielsen et al. 1998].



Alice. Alice has a qubit whose state she doesn't know. She wants to send the state of ths qubit

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

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

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

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

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

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:

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

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 COMPUTERS

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 from Section 4 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 subsection 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 be 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:

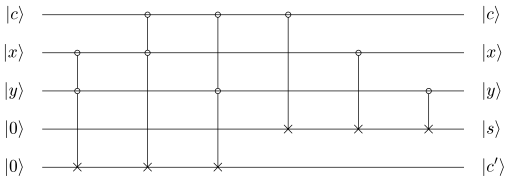
$$T = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes C_{not}.$$

The Toffoli gate T can be used to construct complete set of boolean connectives, as can be seen from the fact that it can be used to construct the AND and NOT operators in the following way:

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

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 [1996] define more complex circuits that include in-place addition and modular addition.

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

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

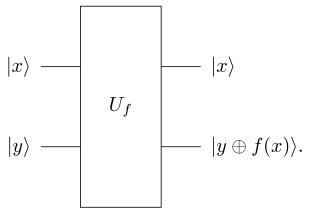
where S is the swap operation

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

The reader can verify that F, like T, is complete for combinatorial circuits.

Deutsch has shown [1985] that it is possible to construct reversible quantum gates for any classically computable function. In fact, it is possible to conceive of a universal quantum Turing machine [Bernstein and Vazirani 1997]. 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 with m input and k output bits can be implemented on quantum computer, we assume the existence of a *quantum gate array* U_f that implements f. U_f is a $m+k$–bit transformation of the form $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$, where $\oplus$ denotes the bitwise exclusive-OR.[6] Quantum gate arrays U_f, defined in this way, are unitary for any function f. To compute $f(x)$ we apply U_f to $|x\rangle$ tensored with k zeros $|x, 0\rangle$. Since $f(x) \oplus f(x) = 0$ we have $U_f U_f = I$. Graphically the transformation $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$ 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,[7] single-bit rotations need to be included. Barenco et al. [1995] show that C_{not} together with all 1-bit quantum gates is a universal gate set. It suffices to include the following 1-bit transformations

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

for all $0 \leq \alpha \leq 2\pi$ together with the C_{not} to obtain a universal set of gates. As we shall see, such nonclassical transformations are crucial for exploiting the power of quantum computers.

5.2. Quantum Parallelism

What happens if U_f is applied to input that 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 and entanglement. 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

[6] $\oplus$ is not the direct sum of vectors.

[7] More precisely, we mean arbitrary unitary transformations up to a constant phase factor. A constant phase shift of the state has no physical, and therefore no computational, significance.

Walsh–Hadamard transformation W of Section 4.1.1 to get a superposition

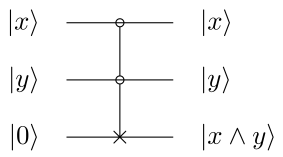
$$\frac{1}{\sqrt{2^n}}(|00\ldots0\rangle + |00\ldots1\rangle + \cdots + |11\ldots1\rangle) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

which should be viewed as the superposition of all integers $0 \le x < 2^n$. Add a k-bit register $|0\rangle$ then by linearity

$$U_f\left(\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, 0\rangle\right) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} U_f(|x, 0\rangle) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, f(x)\rangle$$

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 tradeoff 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 controlled-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:

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

Apply T to the superposition of inputs to get a superposition of the 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 graph of the function. 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, and measuring the input will project the result to the corresponding function value.

Measuring at this point gives no advantage over classical parallelism because only one result is obtained, and worse still one cannot even choose which result one gets. 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 nontraditional 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 values of interest have a larger amplitude and therefore have 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 (later published in Simon [1997]), Peter Shor found a bounded probability polynomial time algorithm for factoring n-digit numbers on a quantum computer. Since the 1970s people have searched for efficient algorithms for factoring integers. The most efficient classical

algorithm known today is that of Lenstra and Lenstra [1993], 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 cryptographic 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.

This description captures the essence of the quantum algorithm but is something of an oversimplification. 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 here, but the techniques for extracting the period are classical.

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

6.1. The Quantum Fourier Transform

Fourier transforms in general map from the time domain to the frequency domain. So Fourier transforms map functions of period r to functions that have nonzero values only at multiples of the frequency $\frac{2\pi}{r}$. The discrete Fourier transform (DFT) operates on N equally spaced samples in the interval $[0, 2\pi)$ for some N and outputs a function whose domain is the integers between 0 and $N-1$. The discrete Fourier transform of a (sampled) function of period r is a function concentrated near multiples of $\frac{N}{r}$. If the period r divides N evenly, the result is a function that has nonzero values only at multiples of $\frac{N}{r}$. Otherwise, the result will approximate this behavior, and there will be nonzero terms at integers close to multiples of $\frac{N}{r}$.

The Fast Fourier transform (FFT) is a version of DFT where N is a power of 2. The quantum Fourier transform (QFT) is a variant of the discrete Fourier transform, which, like FFT, uses powers of 2. The quantum Fourier transform operates on the amplitude of the quantum state, by sending

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

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.

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 at multiples of $\frac{N}{r}$. Thus, when the state is measured, the result would be a multiple of $\frac{N}{r}$, say $j\frac{N}{r}$. But as described above, the quantum Fourier transform only gives approximate results for periods that are not a power of two (i.e., do not divide N). 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 $N = 2^m$ is defined by

$$U_{QFT} : |x\rangle \rightarrow \frac{1}{\sqrt{2^m}} \sum_{c=0}^{2^m-1} \exp\left(\frac{2\pi icx}{2^m}\right)|c\rangle.$$

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 construction 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 2-bit 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 $\theta_{k-j} = \pi/2^{k-j}$. This transformation acts on the kth and jth bits of a larger register. 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. If FFT is followed by measurement, as in Shor's algorithm, the bit reversal can be performed classically. See Shor [1997] for more details.

6.2. A Detailed Outline of Shor's algorithm

The detailed steps of *Shor's* algorithm are illustrated with a running example where we factor $M = 21$.

Step 1. Quantum parallelism. Choose an integer a arbitrarily. If a is not relatively prime to M, we have 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 used in Step 3 for functions whose period is not a power of 2 will be good enough for the rest of the algorithm to work.] Use quantum parallelism as described in Section 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

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

Example. Suppose $a = 11$ were randomly chosen. Since $M^2 = 441 \leq 2^9 < 882 = 2M^2$, we find $m = 9$. Thus, a total of 14 quantum bits, 9 for x and 5 for $f(x)$, are required to compute the superposition of equation 1.

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 last $\lceil \log_2 M \rceil$ qubits of the state of Eq. 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 measure two successive x's in the sum, we would have the period. Unfortunately the laws of quantum physics permit only one measurement.

Example. Suppose that random measurement of the superposition of Eq. 1 produces 8. The state after this measurement[8] (Figure 2) clearly shows the periodicity of f.

Step 3. Applying a quantum Fourier transform. The $|u\rangle$ part of the state will not be

[8] Only the 9 bits of x are shown in Figure 2; the bits of $f(x)$ are known from the measurement.

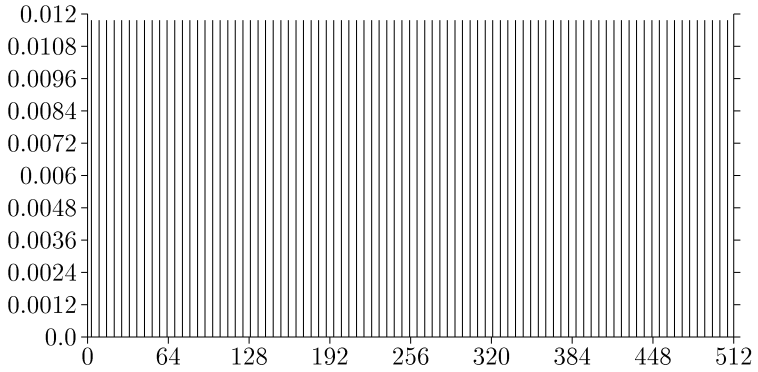


Fig. 2. Probabilities for measuring x when measuring the state $C\Sigma_{x\in X}|x, 8\rangle$ obtained in Step 2, where $X = \{x|211^x \bmod 21 = 8\}\}$.

used, so we will no longer write it. Apply the quantum Fourier transform to the state obtained in Step 2.

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

Standard Fourier analysis tells us that when the period r of the function $g(x)$ defined in Step 2 is a power of 2, the result of the quantum Fourier transform is

$$\sum_j c_j \left| j\frac{2^m}{r} \right\rangle,$$

where the amplitude is 0 except at multiples of $2^m/r$. 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 $\frac{2^m}{r}$.

Example. Figure 3 shows the result of applying the quantum Fourier transform to the state obtained in Step 2. Note that Figure 3 is the graph of the fast Fourier transform of the function shown in Figure 2. In this particular example the period of f does not divide 2^m.

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 $2^m/r$, 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 $\frac{v}{2^m}(= \frac{j}{r})$ to its lowest terms will yield a fraction whose denominator q is the period r. The fact that in general the quantum Fourier transform only approximately gives 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 classical technique is described in Appendix B.

Example. Say that measurement of the state returns $v = 427$. Since v and 2^m are relatively prime, the period r will most likely not divide 2^m and the continued fraction expansion described in Appendix B needs to be applied. The following is a trace of the algorithm described in Appendix B:

i	a_i	p_i	q_i	ϵ_i
0	0	0	1	0.8339844
1	1	1	1	0.1990632
2	5	5	6	0.02352941
3	42	211	253	0.5

which terminates with $6 = q_2 < M \leq q_3$. Thus, $q = 6$ is likely to be the period of f.

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 nontrivial common factor with M.

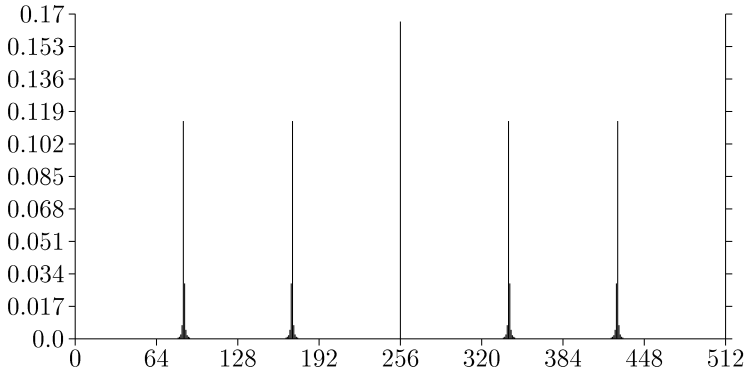


Fig. 3. Probability distribution of the quantum state after Fourier transformation.

The reason why $a^{q/2}+1$ or $a^{q/2}-1$ is likely to have a nontrivial 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, as 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 nontrivial common factor with M.

Example. Since 6 is even either $a^{6/2}-1 = 11^3 - 1 = 1330$ or $a^{6/2}+1 = 11^3+1 = 1332$ will have a common factor with M. In this particular example we find two factors $\gcd(21, 1330) = 7$ and $\gcd(21, 1332) = 3$.

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.

Shor shows that 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. The measurement in Step 2 can be skipped entirely. More generally Bernstein and Vazirani [1997] show that measurements in the middle of an algorithm can always be avoided. If the measurement in Step 2 is omitted, the state consists of a superpositions of several periodic functions all of which have the same period. By the linearity of quantum algorithms, applying the quantum Fourier transformation leads to a superposition of the Fourier transforms of these functions, each of which is entangled with the corresponding u and therefore do not interfere with each other. Measurement gives a value from one of these Fourier transforms. Seeing how this argument can be formalized illustrates some of the subtleties of working with quantum superpostions. Apply the quantum Fourier transform tensored with the identity, $U_{QFT} \otimes I$, to $C\Sigma_{x=0}^{2^n-1}|x, f(x)\rangle$ to get

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

which is equal to

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

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 family of separate functions g_u indexed by u where

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

all with the same period. Note that the amplitudes in states with different u's never interfere (add or cancel) with each other. The transform $U_{QFT} \otimes I$ as applied above can be written

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

$$\rightarrow C' \sum_{u \in R} \sum_{x=0}^{2^n-1} \sum_{c=0}^{2^n-1} G_u(c)|c, u\rangle,$$

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 in a set of possible solutions such that statement $P(x)$ is true." Such problems range from database search to sorting to graph coloring. For example, the graph coloring problem can be viewed as a search for an assignment of colors to vertices so that the statement "all adjacent vertices have different colors" is true. Similarly, a sorting problem can be viewed as a search for a permutation for which the statement "the permutation x takes the initial state to the desired sorted state" is true.

An *unstructured* search problem is one where nothing is know (or no assumption are used) about the structure of the solution space and the statement P. For example, determining $P(x_0)$ provides no information about the possible value of $P(x_1)$ for $x_0 \neq x_1$. A *structured* search problem is one where information about the search space and statement P can be exploited.

For instance, searching an alphabetized list is a structured search problem and the structure can be exploited to construct efficient algorithms. In other cases, like constraint satisfaction problems such as 3-SAT or graph colorability, the problem structure can be exploited for heuristic algorithms that yield efficient solution for some problem instances. But in the general case of an unstructured problem, randomly testing the truth of statements $P(x_i)$ one by one is the best that can be done classically. For a search space of size N, the general unstructured search problem requires $O(N)$ evaluations of P. On a quantum computer, however, Grover showed that the unstructured search problem can be solved with bounded probability within $O(\sqrt{N})$ evaluations of P. Thus Grover's search algorithm [Grover 1996] is provably more efficient than any algorithm that could run on a classical computer.

While Grover's algorithm is optimal [Bennett et al. 1997; Boyer et al. 1996; Zalka 1997] for completely unstructured searches, most search problems involve searching a structured solution space. Just as there are classical heuristic algorithms that exploit problem structure, one would expect that there are more efficient quantum algorithms for certain structured problem instances. Cerf et al. [1998] use Grover's search algorithm in place of classical searches within a heuristic algorithm to show that a quadratic speed-up is possible over a particularly simple classical heuristic for solving NP-hard problems. Brassard et al. [1998], using the techniques of Grover's search algorithm in a less obvious way, show that general heuristic searches have quantum analogs with quadratic speed-up.

There is hope that for certain structured problems a speed-up greater than quadratic is possible. Such algorithms will likely require new approaches that are not merely quantum implementations of classical algorithms. Shor's algorithm, when viewed as a search for factors, is an example of an algorithm that achieves exponential speed-up by using problem structure (number theory) in new ways unique to quantum computation.

Tad Hogg has developed heuristic quantum search algorithms that exploit problem structure. His approach is distincly nonclassical and uses unique properties of quantum computation. One problem with this approach is that, like most heuristic algorithms, the use of problem structure is complicated enough that it is hard to determine the probability that a single execution of an 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 slowdown 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. While less interesting theoretically, even a small polynomial speed-up on average for these computationally difficult problems is of significant practical interest. 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 for an x that makes a statement true. Let n be such that $2^n \geq N$, and let U_p be the quantum gate that implements the classical function $P(x)$ that tests the truth of the statement, where true is encoded as 1.

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

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 $\frac{1}{\sqrt{2^n}} \Sigma_{x=0}^{n-1} |x\rangle$ of all 2^n possible inputs x together with a register set to 0, leading to the superposition

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

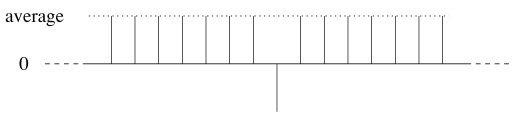
The difficult step is to obtain a useful result from this superposition.

For any x_0 such that $P(x_0)$ is true, $|x_0, 1\rangle$ will be part of the superposition of Eq. 2. Since the amplitude of such a state is $\frac{1}{\sqrt{2^n}}$, the probability that a random measurement of the superposition produces x_0 is only 2^{-n}. The trick is to change the quantum state in Eq. 2 so as to greatly increase the amplitude of vectors $|x_0, 1\rangle$ for which P is true and decrease the amplitude of vectors $|x, 0\rangle$ for which P 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 of Eq. 2 onto the subspace $\frac{1}{\sqrt{k}} \Sigma_{i=1}^{k} |x_i, 1\rangle$ 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 of Eq. 2 must be computed again.

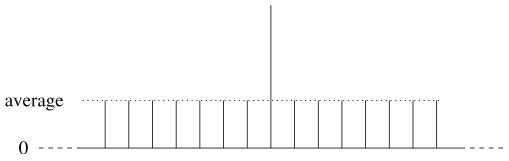
Grover's algorithm then consists of the following steps:

1. Prepare a register containing a superposition of all 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 are 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. [1996] 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, one must know when to stop. Brassard et al. [1998] describe an extension of Grover's algorithm that uses Fourier transforms to determine the number of solutions and the optimal number of iterations. The extension does not increase the overall complexity of the algorithm.

Grover has extended his algorithm to achieve quadratic speed-up for other nonsearch problems such as computing the mean and median of a function [Grover 1998]. Using similar techniques Grover has also shown that certain search problems that classically run in $O(\log N)$ can be solved in $O(1)$ on a quantum computer. Grover's search can used as a subroutine in other quantum computations, since Biron et al. [1998] show how the technique can be used with arbitrary initial amplitude distributions while still maintaining $O(\sqrt{N})$ complexity.

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

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

where A denotes the average of the a_i's, is performed by the $N \times N$ matrix

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

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 surprisingly simple 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 $U_P : |x, b\rangle \rightarrow |x, b \oplus P(x)\rangle$. 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.

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

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

7.2. Heuristic Search

7.2.1 A Note on the Walsh–Hadamard Transform. Another representation for the Walsh–Hadamard transformation of Section 4.1.1 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 $r_{n-1} \dots r_0$ be the binary representation of r and $s_{n-1} \dots s_0$ be the binary representation of s.

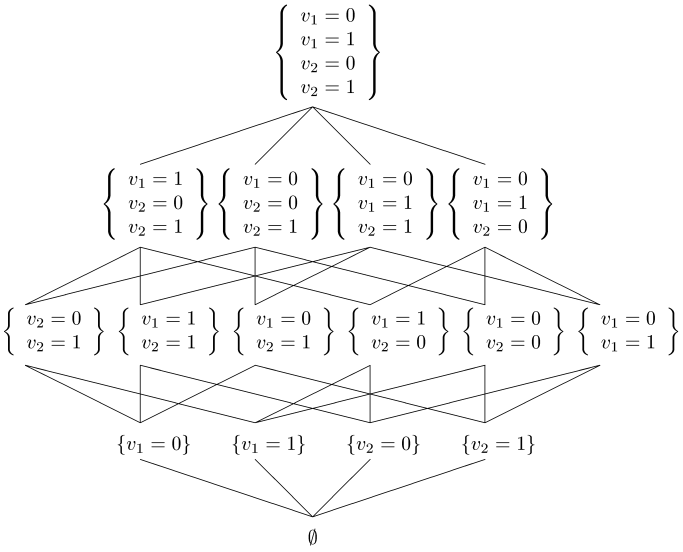


Fig. 4. Lattice of variable assignments in a CSP.

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

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 4 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 5 shows the lattice of Figure 4 rewritten in ket notation, where 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 $|0 \ldots 0\rangle$ 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 [Hogg 1996; 1998] of constructing a unitary matrix for moving amplitude up the lattice. We will describe both methods, and then describe

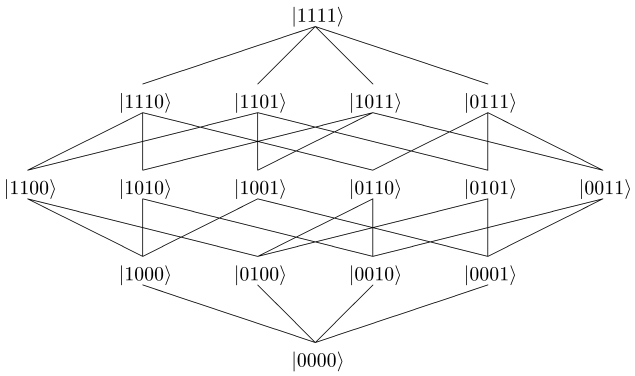
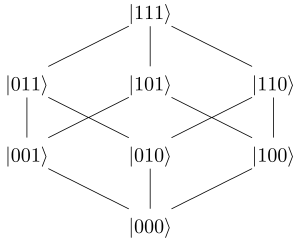


Fig. 5. Lattice of variable assignments in ket form.

how he moves amplitude away from bad sets.

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



We want to transform

$$
\begin{aligned}
|000\rangle &\rightarrow 1/\sqrt{3}(|001\rangle + |010\rangle + |100\rangle \\
|001\rangle &\rightarrow 1/\sqrt{3}(|011\rangle + |110\rangle + |101\rangle \\
&\cdots
\end{aligned}
$$

The complete matrix for this transformation looks like (as usual the basis vectors are ordered according to their binary representation)

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

Unfortunately this transformation is not unitary. Hogg [1996] uses the fact 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 [Hogg 1998] 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 that 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 the amplitude distributed to sets that have bad subsets cancels, whereas the amplitude distributed to sets from all good subsets adds. Different choices here 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 (Section 7.1.2) with a P that tests whether a given state satisfies all of the constraints or not. 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 analyze. 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 an unintended and often nonunitary fashion.

Steane [1998] 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 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

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

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, such as tensor products

of the single-bit error transformations $\{I, X, Y, Z\}$, or more general multibit transformations. In any case, an error can be written as $\Sigma_i e_i E_i$ 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 operator 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 $\Sigma_i e_i E_i|\phi\rangle$ 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:

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

   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 $E^{-1}_{i_0}$ 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 $|0\rangle \rightarrow |000\rangle$ and $|1\rangle \rightarrow |111\rangle$. 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

$$S : |x_0, x_1, x_2, 0, 0, 0\rangle \rightarrow |x_0, x_1, x_2, x_0 \text{ XOR } x_1, x_0 \text{ XOR } x_2, x_1 \text{XOR } x_2\rangle,$$

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

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

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

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

Measuring the last three bits of this state yields either $|110\rangle$ or $|101\rangle$. Assuming the measurement produces the former, 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 nonphysicists. We have described some of the quantum mechanical effects, such as 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, which 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 speed-up, 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 [1998] describe the implementation on a 2-bit quantum computer of a constant time algorithm [Deutsch and Jozsa 1992] that can distinguish whether a function is balanced or constant. Grover [1998] describes an efficient algorithm for estimating the median of a set of values and both Grover [1998] and Terhal and Smolin [1997], using different methods, 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 we can find quantum algorithms that provide exponential speed-up for problems other than factoring. There is some speculation among physicists that quantum transformations might be slightly nonlinear. So far all experiments that have been done are consistent with the standard linear quantum mechanics, but a slight nonlinearity is still possible. Abrams and Lloyd [1998] show that even a very slight nonlinearity could be exploited to solve all NP-hard problems 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 efficiently may depend on subtle issues in physics.

The unique properties of quantum computers give rise to new kinds of complexity classes. For instance, BQP is the set of all languages accepted by a quantum Turing machine in polynomial time with bounded probability of error. Details of the extensive research done in the field of quantum complexity theory is beyond the scope of this paper. The interested reader may start by consulting Bennett et al. [1997] and Watrous [1998] respectively for analyses of time and space complexity of quantum computation. Williams and Clearwater [1998] contains an introduction to early results in quantum complexity.

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 for the development of practical quantum computers as advances in the hardware side.

9.1. Further Reading

Andrew Steane's survey article "Quantum Computing" [Steane 1998] is aimed at physicists. 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. Another shorter and very readable tutorial can be found in Berthiaume [1997].

Richard Feynman's *Lectures on Computation* [Feynman 1996] contains a reprint of the lecture "Quantum Mechanical Computers" [Feynman 1985], 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* [Williams and Clearwater 1998] comes with software, in the form of Mathematica

notebooks, that simulates some quantum algorithms such as Shor's algorithm.

The second half of the October 1997 issue of the SIAM Journal of Computing contains six seminal articles on quantum computing, including four we have already cited [Bennett et al. 1997; Bernstein and Vazirani 1997; Shor 1997; Simon 1997].

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`. Links to research projects and other information about quantum computing can be found on our web site `http://www.fxpal.xerox.com/Quantum Computing`.

APPENDIX

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 k-dimensional vectors respectively, then $A \otimes B$[9] is a transformation on nk-dimensional vectors.

The exact mathematical details of tensor products are beyond the scope of this paper (see Hungerford [1974] 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:

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

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

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

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

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

The conjugate transpose distributes over tensor products; that is,

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

A matrix U is *unitary* if its conjugate transpose is 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 $U = A_1 \otimes A_2 \otimes \cdots \otimes A_n$. Then U is unitary if $A_i^* A_i = k_i I$ and $\Pi_i k_i = 1$.

$$\begin{aligned}U^*U &= \left(A_1^* \otimes A_2^* \otimes \cdots \otimes A_n^*\right)\\ &\quad \times (A_1 \otimes A_2 \otimes \cdots \otimes A_n)\\ &= A_1^*A_1 \otimes A_2^*A_2 \otimes \cdots \otimes A_n^*A_n\\ &= k_1 I \otimes \cdots \otimes k_n I\\ &= I\end{aligned}$$

where each I refers to the identity matrix of appropriate dimension.

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

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

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 $|\frac{v}{2^m} - \frac{p}{q}| < \frac{1}{M^2}$. 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

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

compute the first fraction $\frac{p_n}{q_n}$ such that $q_n < M \leq q_{n+1}$. See any standard number theory text, such as Hardy and Wright [1979], 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 $\frac{j}{r}$, because 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.

ACKNOWLEDGMENTS

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 Lee Corbin, David Goldberg, Lov Grover, Norman Hardy, Vaughan Pratt, Marc Rieffel, and the anonymous referees for detailed comments on earlier drafts of this paper. Finally, we would like to thank FXPAL for enthusiastically supporting this work.

REFERENCES

ABRAMS, D. S. AND LLOYD, S. 1998. Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and #p problems. Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9801041`.

BARENCO, A., BENNETT, C. H., CLEVE, R., DIVINCENZO, D. P., MARGOLUS, N. H., SHOR, P. W., SLEATOR, T., SMOLIN, J. A., AND WEINFURTER, H. 1995. Elementary gates for quantum computation. *Physical Review A 52*, 5, 3457–3467. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9503016` and at `http://vesta.physics.ucla.edu/cgi-bin/uncompress_ps_cgi?torgats1.ps`.

BENNETT, C. H. 1992. Quantum cryptography using any two nonorthogonal states. *Physical Review Letters 68*, 3121–3124.

BENNETT, C. H., BERNSTEIN, E., BRASSARD, G., AND VAZIRANI, U. V. 1997. Strengths and weaknesses of quantum computing. *Society for Industrial and Applied Mathematics Journal on Computing 26*, 5, 1510–1523. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9701001`.

BENNETT, C. H. AND BRASSARD, G. 1987. Quantum public key distribution reinvented. *SIGACT News (ACM Special Interest Group on Automata and Computability Theory) 18*, 51–53.

BENNETT, C. H., BRASSARD, G., AND EKERT, A. K. 1992. Quantum cryptography. *Scientific American 267*, 4 (Oct.), 50.

BERNSTEIN, E. AND VAZIRANI, U. V. 1997. Quantum complexity theory. *Society for Industrial and Applied Mathematics Journal on Computing 26*, 5, 1411–1473. A preliminary version of this paper appeared in the *Proceedings of the 25th Association for Computing Machinery Symposium on the Theory of Computing.*

BERTHIAUME. 1997. Quantum computation. In ALAN L. SELMAN, ED., *Complexity Theory Retrospective, In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday*, (July 5, 1988), Vol. 2, 23–51.

BIRON, D., BIHAM, O., BIHAM, E., GRASSEL, M., AND LIDAR, D. A. 1998. Generalized grover search algorithm for arbitrary initial amplitude distribution. Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9801066`.

BOSCHI, D., BRANCA, S., MARTINI, F. D., HARDY, L., AND POPESCU, S. 1998. Experimental realization of teleporting an unknown pure quantum state via dual classical and einstein-podolski-rosen channels. *Physical Review Letters 80*, 1121–1125.

BOUWMEESTER, D., PAN, J.-W., MATTLE, K., EIBL, M., WEINFURTER, H., AND ZEILINGER, A. 1997. Experimental quantum teleportation. *Nature 390*, 575.

BOYER, M., BRASSARD, G., HØYER, P., AND TAPP, A. 1996. Tight bounds on quantum search. In *Proceedings of the Workshop on Physics of Computation: PhysComp '96* (Los Alamitos, CA, 1996), 36–43. Institute of Electrical and Electronic Engineers Computer Society Press. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9605034`.

BRASSARD, G., HØYER, P., AND TAPP, A. 1998. Quantum counting. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9805082`.

CERF, N. J., GROVER, L. K., AND WILLIAMS, C. P. 1998. Nested quantum search and np-complete problems. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9806078`.

CIRAC, J. I. AND ZOLLER, P. 1995. Quantum computations with cold trapped ions. *Physical Review Letters 74*, 4091–4094.

CORY, D. G., MASS, W., PRICE, M., KNILL, E., LAFLAMME, R., ZUREK, W. H., HAVEL, T. F., AND SOMAROO, S. S. 1998. Experimental quantum error correction. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9802018`.

DEUTSCH, D. 1985. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London Series A A400*, 97–117.

DEUTSCH, D. AND JOZSA, R. 1992. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London Series A A439*, 553–558.

DIRAC, P. 1958. *The Principles of Quantum Mechanics* (4th ed.). Oxford University Press.

EKERT, A. K., RARITY, J., TAPSTER, P., AND PALMA, G. 1992. Practical quantum cryptography based on two-photon interferometry. *Physical Review Letters 69*, 1293–1295.

FEYNMAN, R. 1982. Simulating physics with computers. *International Journal of Theoretical Physics 21*, 6&7, 467–488.

FEYNMAN, R. 1985. Quantum mechanical computers. *Optics News 11*, 11–20. Also in *Foundations of Physics*, 16(6), 507–531, 1986.

FEYNMAN, R. 1996. In *Feynman Lectures on Computation*. A. J. HEY AND R. W. ALLEN EDS., Addison-Wesley.

FEYNMAN, R. P., LEIGHTON, R. B., AND SANDS, M. 1965. *Lectures on Physics, Vol. III*. Addison-Wesley.

GERSHENFELD, N. A. AND CHUANG, I. L. 1997. Bulk spin resonance quantum computing. *Science 275*, 350–356.

GREENSTEIN, G. AND ZAJONC, A. G. 1997. *The Quantum Challenge.* Jones and Bartlett Publishers, Sudbury, Mass.

GROVER, L. K. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing* (Philadelphia, Pennsylvania, 22–24 May 1996), pp. 212–219.

GROVER, L. K. 1998. A framework for fast quantum mechanical algorithms. *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, 53–62. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9711043`.

HARDY, G. H. AND WRIGHT, E. M. 1979. *An Introduction to the Theory of Numbers*. Oxford University Press.

HOGG, T. 1996. Quantum computing and phase transitions in combinatorial search. *Journal of Artificial Intelligence Research 4*, 91–128. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9508012`.

HOGG, T. 1998. Highly structured searches with quantum computers. *Physical Review Letters 80*, 2473–2473.

HUGHES, R. J., BUTTLER, W. T., KWIAT, P. G., LAMOREAUX, S. K., MORGAN, G. L., NORDHOLT, J. E., AND PETERSON, C. G. 1999. Practical Quantum Cryptography for Secure Free-Space Communications. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9905009`.

HUGHES, R. J., BUTTLER, W. T., KWIAT, P. G., LUTHER, G. G., MORGAN, G. L., NORDHOLT, J. E., PETERSON, C. G., AND SIMMONS, C. M. 1997. Secure communications using quantum cryptography. In S. P. HOTALING AND A. R. PIRICH EDS., *Photonic Quantum Computing*, Vol. 3076, 2–11.

HUNGERFORD, T. A. 1974. *Algebra*. Springer Verlag, New York, Heidelberg, Berlin.

JONES, J. A. AND MOSCA, M. 1998. Implementation of a quantum algorithm on a nuclear magnetic resonance quantum computer. *Journal of Chemical Physics 109*, 5, 1648–1653. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9801027`.

LAFLAMME, R., KNILL, E., ZUREK, W., CATASTI, P., AND MARIAPPAN, S. 1997. NMR GHZ. Los Alamos

Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9709025.

LENSTRA, A. AND LENSTRA, H. EDS. 1993. *The Development of the Number Field Sieve*, Vol. 1554 of *Lecture Notes in Mathematics*. Springer Verlag.

LIBOFF, R. L. 1997. *Introductory Quantum Mechanics* (3rd ed.). Addison-Wesley, Reading, Mass.

LO, H.-K. AND CHAU, H. F. 1999. Unconditional security of quantum key distribution over arbitrarily long distances. *Science 283*, 2050–2056.

MAYERS, D. 1998. Unconditional Security in Quantum Cryptography. Preprint at Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9802025.

NIELSEN, M. A., KNILL, E., AND LAFLAMME, R. 1998. Complete Quantum Teleportation using Nuclear Magnetic Resonance. Preprint at Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9811020.

SCHULMAN, L. J. AND VAZIRANI, U. 1998. Scalable NMR Quantum Computation. Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9804060.

SHOR, P. W. 1994. Algorithms for quantum computation: Discrete log and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (Nov.), 124–134. Institute of Electrical and Electronic Engineers Computer Society Press. ftp://netlib.att.com/netlib/att/math/shor/quantum.algorithms.ps.Z.

SHOR, P. W. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *Society for Industrial and Applied Mathematics Journal on Computing 26*, 5, 1484–1509. Expanded version of Shor [1994].

SIMON, D. R. 1997. On the power of quantum computation. *Society for Industrial and Applied Mathematics Journal on Computing 26*, 5, 1474–1483. A preliminary version of this paper appeared in the *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*.

STEANE, A. 1996. The Ion Trap Quantum Information Processor. Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9608011.

STEANE, A. 1998. Quantum computing. *Reports on Progress in Physics 61*, 2, 117–173. Preprint at Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9708022.

TERHAL, B. M. AND SMOLIN, J. A. 1997. Single Quantum Querying of a Database. Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9705041.

VANDERSYPEN, L. M. K., YANNONI, C. Y., SHERWOOD, M. H., AND CHUANG, I. L. 1999. Realization of Effective Pure States for Bulk Quantum Computation. Preprint at Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9905041.

VEDRAL, V., BARENCO, A., AND EKERT, A. K. 1996. Quantum Networks for Elementary Arithmetic Operations. Physical Review A. Preprint at Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9511018.

WATROUS, J. 1998. Relationships between quantum and classical space-bounded complexity classes. In *13th Annual IEEE Conference on Computational Complexity* (June 1998), 210–227.

WILLIAMS, C. P. AND CLEARWATER, S. H. 1998. *Explorations in Quantum Computing*. Telos, Springer-Verlag.

WOOTTERS, W. K. AND ZUREK, W. H. 1982. A single quantum cannot be cloned. *Nature 299*, 802.

ZALKA, C. 1997. Grover's Quantum Searching Algorithm is Optimal. Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9711070.

Accepted September 1999