Michael Spillane
Final Project 414

Introduction

Factoring of integers has been a problem of interest in number theory for years. Using classical computers the current (in 2010) best algorithm for factoring has order O(exp((logN)^(1/3)*log(log (N))^(2/3))). Shor's algorithm uses a quantum computer to reduce the problem to $O((\log N)^3)$, which is exponentially faster. Shor's algorithm is a special case of a period finding algorithm. While quantum computers would be much faster than classical computers for factoring, the implementation of quantum computing is still in its infancy, at least compared with classical computing. Nonetheless, in 2001 a group working at IBM successfully used Shor's algorithm to factor 15 using NMR (Nuclear Magnetic Resonance) qubits. Since then it has also been successfully implemented using photonic quantum computers in 2007. Shor's algorithm uses a quantum Fourier Transform algorithm ($O(\log N)^2$)), as well as period calculating algorithm (O(logN)). I will give a brief introduction to the necessary elements of quantum mechanics/computing and then describe the algorithms needed. The calculations involved were taken from lecture notes for Physics 427 summer 2009 as well as the book by Le Bellac.

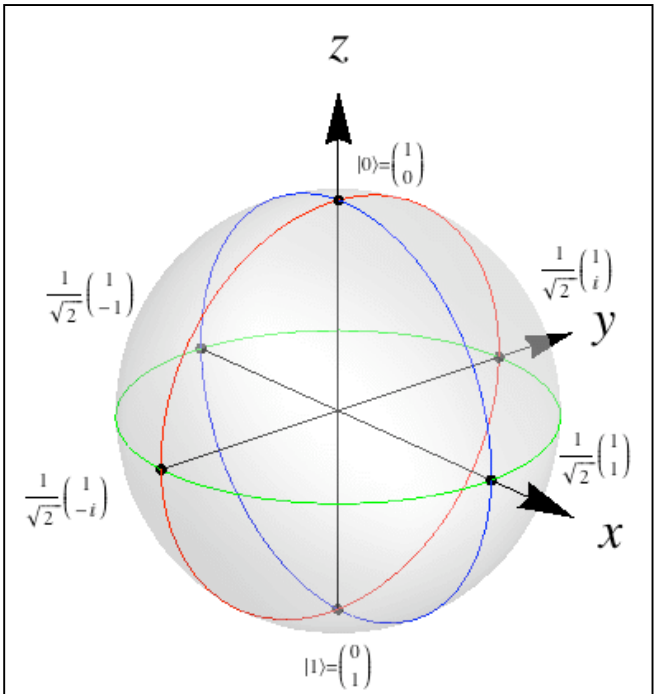

Figure 1. This is a picture of the Bloch Sphere. The north and south poles represent the base states. A qubit can take any value on the sphere. These are linear superpositions of the |0> and |1> states whose inner product with themselves is 1. (http://www.bradrubin.com/Site/Blog/Entries/2008/12/29_Entry_1.html)

Basics of Quantum Computing

In quantum computing a two state system is usually known as a qubit. A two state system is some quantum mechanical system which only has two states. Analogous with classical computing, the two states are represented as |0> (the ground state) and |1> (the excited state). However, unlike classical computing, in quantum computing we can have superpositions of these two states. This is a quantum mechanical effect analogous to superpositions of two waves on a string. These states are represented as points on the Bloch Sphere (Figure 1). During the running of a quantum algorithm each of the qubits

will become some superposition of the |0> and |1>. However, when they are measured they snap back to one of the two base states with a probability given by the inner product of their superposition state with the base states.

Because we are working with a quantum mechanical system all actions operating on the qubits must be unitary operations ($U^{\dagger}U=I$). Also it is useful to recall that if we let $A = \{|x\rangle\}$ be a complete orthonormal basis for the Hilbert space of possible states then

$$\sum_A |x\rangle\langle x| = I,$$

where in vector notation $|1\rangle\langle 1| = (0,1)^T*(0,1) = (\{0,0\},\{0,1\})$ this will be referred to as the completeness relationship. (Griffiths)

Quantum computers are able to do many computations more rapidly than classical computers, because they are able "to explore at the same time all the branches of a nondeterministic algorithm" (Le Bellac). If one subscribes to the many worlds philosophy (that all possible choices are taken in some universe) a quantum computer is simply using many different universes to do the computation and by interfering the qubits arrive at the answer (in a probabilistic way) when the qubits are measured. Even if you do not subscribe to this philosophy it provides a way to understand what makes a quantum computer so powerful.

Quantum Fourier Transform

One of the components necessary for Shor's algorithm is the quantum Fourier Transform. The quantum Fourier Transform goes as follows. First let any integer x such that $0 \le x \le 2^n-1$ be written as

$$|x\rangle = |x_{n-1}\, x_{n-2} \ldots x_0\rangle$$

where x_j is the j^{th} digit in the binary representation of x and n is the number of qubits. Next let U_{FT} be the unitary operator whose elements are defined by

$$\langle y|U_{FT}|x\rangle = (U_{FT})_{yx} = 2^{-n/2} \exp\hat{}(2\pi i x y/2^n).$$

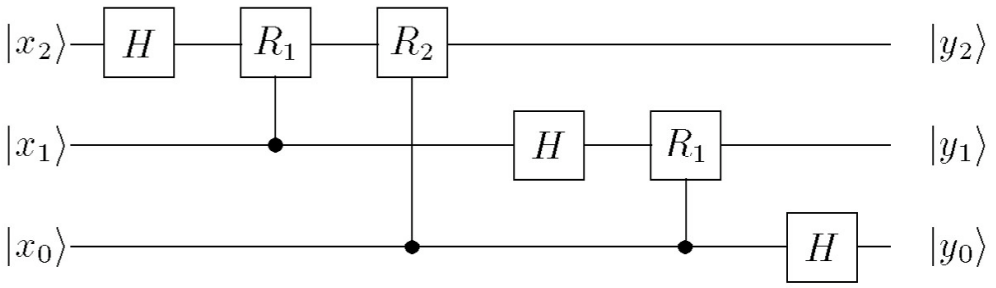

Figure 2. This circuit diagram shows how a quantum Fourier Transform would work for 3 qubits. Here the H stands for a Hadamard gate, which has the following affect on the basis vectors ($H|0\rangle = 2^{-1/2}(|0\rangle+|1\rangle)$, $H|1\rangle=2^{-1/2}(|0\rangle-|1\rangle)$). If one writes out the corresponding matrix it is easy to see that it is unitary. The R_d are control rotation gates which has no affect on the control qubit and if the control qubit is 0 then nothing happens to the other qubit if however the control qubit is 1 then |0> -> |0> and |1> -> exp^(ip/2^d)|1>. If you write out the matrix it is unitary.

Next we define a vector $|\Psi\rangle$ as

$$|\Psi\rangle = \sum_{x\varepsilon B} f(x)|x\rangle \text{ where } B = \{0,1,\ldots,2^n-1\} \quad (1)$$

and

$$\sum_B |f(x)|^2 = 1. \quad (2)$$

The second condition (Eq. 2) just means that the vector is appropriately normalized. Observe that $\langle x|\Psi\rangle = f(x)$. Then

$$\langle y|U_{FT}|\Psi\rangle = \sum_{x\varepsilon B} \langle y| U_{FT}|x\rangle\langle x| \Psi\rangle \quad (3)$$

$$= 2^{-n/2} \sum_{x\varepsilon B} \exp\hat{}(2\pi i xy/2^n)\, f(x) = F(y). \quad (4)$$

where F(y) is the Fourier Transform of y. The first equality (Eq. 3) used the completeness relationship to insert a complete set of states and the second (Eq. 4) was just evaluating the sum and observing that the result was the Fourier Transform.

Quickly we will now show that U_{FT} is in fact unitary.

$$\begin{aligned}((U_{FT}^{\dagger})(U_{FT}))_{y',y} &= \sum_{x\varepsilon B}(U_{FT}^{\dagger})_{y',x}(U_{FT})_{xy} \\ &= \sum_{x\varepsilon B}(U_{FT}^{*})_{xy'}(U_{FT})_{xy} \\ &= 2^{-n/2}\left(\sum_{x\varepsilon B}\exp\hat{}(2\pi i(y-y')x/2^n)\right) \\ &= \delta_{y',y}.\end{aligned}$$

Where $\delta_{y',y}$ is the Kronker delta function. The first equality is just the definition of matrix multiplication. The second is the definition of the Hermitian conjugate, while the third is substituting into the equation. Finally, the last equality uses the fact that the sum over y was a geometric series. So $U^{t}_{FT}U_{FT} = I$ which proves that U_{FT} is a unitary operator.

The quantum Fourier Transform can be implemented using Hadamard gates (see caption in Figure 2 for discussion of such gates) and controlled rotation gates (see caption in Figure 2). The quantum circuit for the quantum Fourier Transform can be seen in Figure 2 for three qubits. Generalizing this to n qubits involves n H-gates and

$$(n-1) + (n-2) + \ldots + 1 = n(n-1)/2$$

controlled rotation gates. So as a whole the quantum Fourier Transform requires $O(n^2)$ gates. One can work out that the circuit shown in Figure 2 does indeed give the desired result and then argue by induction that it would work for n qubits, but the calculation is not particularly enlightening. If the reader would like to see the computation they should look in any introductory quantum computing text. This algorithm for the Fourier transform is significantly faster than the classical algorithm.

Period calculating algorithm

The second algorithm needed is the period calculating algorithm. It uses the Fourier Transform developed in the previous section. Unlike the Fourier Transform the period calculating algorithm is probabilistic. As we will show it is successful roughly 1 in 4 times. Shor's algorithm works by finding the period of $f(x) = b^x \bmod N$ where N is the number to be factored and b is a random integer which does not divide N (if it did we would be done because we had found a factor of N). We start with n qubits where $2^n > N^2$.

We start with the initial state containing n+m qubits

$$|\Phi\rangle = 2^{-n/2}\left(\sum_{x\varepsilon B}|x\rangle\right)\otimes|00\ldots0\rangle,$$

where $B = \{0,1,\ldots,2^n-1\}$. From there apply a unitary transformation which takes $|\Phi\rangle$ to $|\Psi\rangle$ in the following way

$$|\Psi\rangle = 2^{-n/2}\left(\sum_{x\varepsilon B}|x\otimes f(x)\rangle\right.$$

Now if we were to measure the output register (the m qubits) and got f_0 then the first n qubits would be in the state

$$|\Psi_0\rangle = M^{-1}\sum_{x\varepsilon C}|x\rangle,$$

where $C = \{x:f(x) = f_0\}$ and M is a normalization to make the vector ($|\Psi_0\rangle$) unit length. Let m = (#C). Then if we let the period of f(x) = r and x_0 be the smallest x which is an element of C, then

$$|\Psi_0\rangle = m^{-1/2}\sum_{k\varepsilon(\mathbf{Z}/m\mathbf{Z})}|x_0 + kr\rangle.$$

Now we take the quantum Fourier Transform of $|\Psi_0\rangle$ and calculate the inner product of that with $|y\rangle$ to get

$$\langle y|U_{FT}|\Psi\rangle = 2^{-n/2}\, m^{-1/2}\sum_{k\varepsilon(\mathbf{Z}/m\mathbf{Z})}\exp^{\wedge}(2\pi i y(x_0 + kr)/2^n).$$

If we then find the magnitude squared of this value we find the probability that of finding a given $|y\rangle$ as the measured value of the output. So

$$P(y) = |2^{-n/2}\, m^{-1/2}\sum_{k\varepsilon(\mathbf{Z}/m\mathbf{Z})}\exp^{\wedge}(2\pi i y(x_0 + kr)/2^n)|^2$$
$$= (2^n m)^{-1}\,|\sum_{k\varepsilon(\mathbf{Z}/m\mathbf{Z})}\exp^{\wedge}(2\pi i y k r)/2^n)|^2.$$

Now we calculate the sum using the equation for a geometric series. To get

$$\sum_{k\varepsilon(\mathbf{Z}/m\mathbf{Z})}\exp^{\wedge}(2\pi i y k r)/2^n) = (1- \exp^{\wedge}(2\pi i y m r)/2^n))/(1- \exp^{\wedge}(2\pi i y r)/2^n))$$
$$= e^{\wedge}(\pi i(m-1)r)/2^n)\sin(\pi y m r/2^n)/ \sin(\pi y r/2^n).$$

There are now two cases. In the case that $2^n/r$ is an integer then $m = 2^n/r$
$P(y) = (2^n m)^{-1}\sin^2(\pi y)/ \sin^2(\pi y/K) = 1/r$ if $y = j\, m$ for some integer j.
Otherwise,

$$P(y) = 0.$$

Now in Case 1, $j/r = y/2^n$ which tells us what j and r are. In the other case we can write

$$y_j = j\,(2^n/r)+\delta_j.$$

Then,

$$P(y_j) = (2^n m)^{-1}\, \sin(\pi\delta_j m r/2^n)/ \sin(\pi\delta_j r/2^n).$$

This has large values when y is close to $j\,(2^n/r)$ see Figure 3. To calculate just how likely we are to measure the correct value of y consider

$$2\, x/\pi \le \sin(x) \le x \text{ for } 0 \le x \le \pi/2.$$

So if we require $|\pi\delta_j| < \pi/2$ then

$$P(y_j) \ge 4m/(p^2 2^n) \approx 4/(p^2 r).$$

Since r is large and $0 \le j \le r-1$ there is at least a $4/p^2$ (40%) chance of finding one value of y_j near $j2^n/r$. At this point, one can take $y_j/2^n$ (which is known) and expand it out using continued fractions to get j_0/r_0. If we are lucky, then j and r have no common factors and we immediately get $r = r_0$. This will happen roughly $6/\pi^2 \approx 60\%$ (this comes from $\prod_2^{\infty}(1-p^{-2}) = 6/\pi^2$), so $0.40 \times 0.60 \approx 25\%$ of the time one will immediately get the period which can be tested on a classical computer. If that fails trying $2r_0, 3r_0 \ldots$ may also give the period. If this

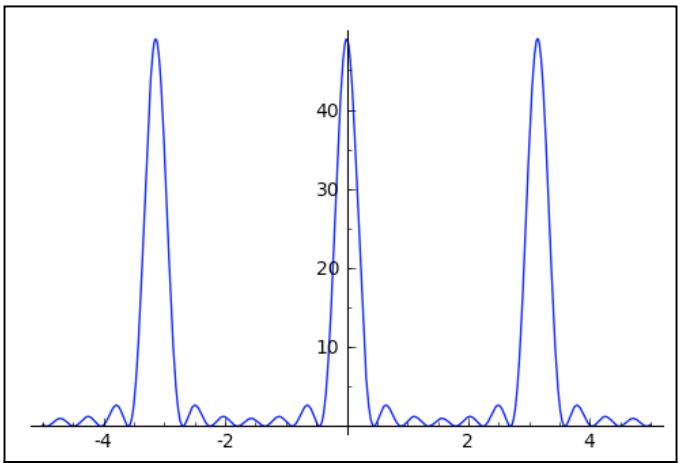

Figure 3. Plot of $\sin^2(7x)/\sin^2(x)$ to show the nature of the function near the origin. It is easy to see that the numbers with the largest probabilities of occurring are those near 0. The peaks to the left and right are farther away than the largest value of x.

fails then likely you did not get the right y_j and the algorithm should be run again. (Le Bellac).

Breaking RSA and Factoring

If we now want to break RSA using the period we calculated. For Eve to break RSA encryption she calculates c where c e = 1 mod r, where e is the public key. Then if b is the message then

$$b^c = a \bmod N$$

where a is the message. Alternatively, if the goal was to factor N then

$$b^r - 1 = 0 \bmod N,$$

so

$$(b^{r/2}-1)(b^{r/2}+1)=0 \bmod N.$$

So as long as

$$b^{r/2} \neq \pm 1 \bmod N$$

and r is even, then

$$p = \gcd(N, (b^{r/2}-1)) \quad (5)$$

and

$$q = \gcd(N, (b^{r/2}+1)) \quad (6)$$

are factors of N. If either of the conditions (Eqs. 5 and 6) fails, Shor's algorithm should be repeated with a different b. The probability of success for this method is greater 50%. (Le Bellac).

Factoring 15

A group at IBM has successfully factored 15, the easiest case, using NMR qubits. These qubits are molecules, which are specially designed with the nucleus having a given spin. Then using the methods of NMR (how a MRI machine works) they manipulated the qubits in order to physically carry out Shor's algorithm. They were successful in showing that 15=3*5. However, it is currently not possible to entangle NMR qubits and, because it has been shown that to get running times faster than standard computers you need to entangle qubits, this is not as impressive a result as was first believed. However, in 2007 a photonic quantum computer managed to factor 15 as well with actual entanglement of the qubits. Photonic qubits use photons (particles of light) along with optical methods of beam splitters and mirrors to perform the computation. The photonic method of quantum computing has issues with scalability, but it is really easy to make quantum gates. Within the next several years, other methods of quantum computing (e.g. trapped ion and supercomputing) should be able to achieve similar results while at the same time having greater likelihood of being scalable. (Lu et. al. 2007)

Conclusion

While quantum computers are currently unable to do realistic computations they have the capability to be much faster. Like classical computers there is an analog to Moore's Law for the number of qubits, which can be manipulated. Because of this there is great hope that quantum computers have a future. Besides Shor's algorithm for factoring there are many other quantum algorithms for searching and determining the parity of a function that require far fewer steps than similar algorithms in a classical

computer. While it is unlikely that quantum computers will completely take over the functions of classical computers, they still have a bright future.

Works Cited

Le Bellac, Michel. Quantum Information and Quantum Computation. Cambridge: Cambridge University Press, 2006.

Lu, Chao-Yang. Daniel E. Browne, Tao Yang, and Jian-Wei Pan. "Demonstration of a Compiled Version of Shor's Quantum Factoring Algorithm Using Photonic Qubits." Physical Review Letters 99 (Dec. 2007): 250504.

Griffiths, David. Introduction to Quantum Mechanics. New Jersey: Pearson, Prentice Hall, 2005.