Shor's Quantum Factorization Algorithm

Tayeb Aïssiou

Department of Mathematics and Statistics
McGill University, Montreal, Quebec
Canada H3A 2K6
e-mail: tayeb.aissiou@mail.mcgill.ca

November 2, 2005

Abstract

In this paper, I will show Shor's [2] analysis to find the period of a periodic function using quantum computers. Finding the period of the function:

$$f(x) = a^x Mod(N)$$

where a is any integer relativly prime to the composed integer N, can lead us to factorize N. The use of Quantum Fourier Transform (QFT) is essential to extract the period. The algorithm does the factorization in a very elegant way.

1 Introduction

The factorization of an integer is a big challenge that kept busy many mathematicians and computer scientists. There exists an algebraic algorithm that reduces the task of factorizing an integer to a period-finding. We have also a quantum algorithm developed by Shor that finds the period of any periodinc function fastly. This paper will explain both the algebraic and the quantum algorithms. The algebraic part uses some basic number theory topics that will be introduced briefly and the quantum part uses the Quantum Fourier Transform (QFT) and its power to extract the period of any periodic function

2 Number Theory

2.1 Environnement

2.1.1 Modular Arithmetic and Groups

The ring $\mathbf{Z}/p\mathbf{Z}$ that I will denote by $\mathbf{Z}_p$, where p is a prime number, is the set of numbers $\{0, 1, 2, ..., p-1\}$ that satisfies some properties undersome operation $\star$ [1]. Now, let me impose one more condition which is the fact that each element of the set has its inverse in the set: $\{\forall x \in \mathbf{Z}_p, \exists x' \in \mathbf{Z}_p$ such that $x * x' = 1 \pmod{p}\}$. The ring $\mathbf{Z}_p$ becomes the group $\mathbf{Z}_p^\star$ which is the set $\{1, 2, ..., p-1\}$. The cardinality of this group is $p-1$.
For the general case, the ring $\mathbf{Z}_n$ where n is a composed integer contains n elements, but the group $\mathbf{Z}_n^\star$ doesn't contain $n-1$ elements.
For exemple, the group $\mathbf{Z}_6^\star$ contains only two elements: $\{1, 5\}$ because the elements $\{2, 3, 4\}$ of the ring Z_6 don't have an inverse under multiplication.
Something interesting about the group $\mathbf{Z}_p^\star$ is the following equality:

$$a^p \equiv_p a \Leftrightarrow a^{p-1} = 1 (\text{mod } p)$$

where $a \in \mathbf{Z}$ and $p \not| a$ This last equation is knows as Fermat Little Theorem proved in most Abstract Algebra texts [1].

2.1.2 Euler phi (ϕ) function

Fermat Little Theorem works for the groups $\mathbf{Z}_p^\star$ where p is a prime number. How about the group $\mathbf{Z}_n^\star$ for a composed integer n? The answer was given by Euler when he defined the ϕ function. The Euler ϕ function is a function when applied on any integer n it returns the number of elements relativly prime to n. In other words the Euler ϕ function gives us the cardinality of the group $\mathbf{Z}_n^\star$.
Given this Euler ϕ function, there exists an equivalent theorem to Fermat Little Theorem known as Euler-Fermat Theorem that uses the Euler ϕ function and gives us the following equation:

$$a^{\phi(n)} = 1 \pmod{N},$$

where $a \in \mathbf{Z}$ and $\gcd(a, n) = 1$

Here are some properties of the Euler ϕ funtion[1]:

$$\phi(p) = p - 1$$

$$\phi(p^\alpha) = p^\alpha - p^{\alpha-1}$$

$$\phi(p \times q) = \phi(p) \times \phi(q) = (p-1)(q-1)$$

where p and q are primes for a general $n = p_1^{\alpha_1} \times p_2^{\alpha_2} \times \ldots \times p_s^{\alpha_s}$ we have:

$$\phi(n) = \phi(p_1^{\alpha_1} \times p_2^{\alpha_2} \times \ldots \times p_s^{\alpha_s}) = \phi(p_1^{\alpha_1}) \times \phi(p_2^{\alpha_2}) \times \ldots \times \phi(p_s^{\alpha_s})$$

2.1.3 The Greatest Commun Divisor (GCD)

There exists a fast algorithm that calculates the Greatest Commun Divisor of the two integers a and b known as the Euclidean algorithm[1].
This is an example of how to comupute $\gcd(356, 100)$:

$$356 = \boxed{100} \times 3 + \textcircled{56}$$

$$\boxed{100} = \textcircled{56} \times 1 + 44$$

$$56 = 44 \times 1 + 12$$

$$44 = 12 \times 3 + 8$$

$$12 = 8 \times 1 + 4$$

$$8 = \boxed{4} \times 2 + 0$$

The $\gcd(356, 100)$ is 4

2.2 Application for factoring an integer

It is finally the time to stick togather all these amazing topics in mathematics to get something useful.
Let n be the integer we want to factorize. Assume $n = p \times q$ where p and q are two very large primes and assume we have a magic box that gives us the ϕ function of that integer n.

$$n \longrightarrow \boxed{\boxed{\text{MAGIC BOX}}} \longrightarrow \phi(n)$$

Let a be a random integer relatively prime to n, according to Euler-Fermat Theorem we have:

$$a^{\phi(n)} = 1 \bmod(n)$$

$$a^{\phi(n)} - 1 = 0 \bmod(n)$$

$$(a^{\phi(n)/2} + 1)(a^{\phi(n)/2} - 1) = 0 \bmod(n)$$

Now we have to assume that:

$$(a^{\phi(n)/2} - 1) \neq 0 \bmod(n) \quad \text{and} \quad (a^{\phi(n)/2} + 1) \neq 0 \bmod(n)$$

Otherwise, we must choose another random integer a relativly prime to n. Because none of $(a^{\phi(n)/2} - 1)$ and $(a^{\phi(n)/2} + 1)$ is 0 mod(n) therefore:

$$\gcd(a^{\phi(n)/2} - 1, n) = p$$

$$\gcd(a^{\phi(n)/2} + 1), n) = q$$

where p and q are the factors of n. In RSA cryptography system, we use integers that are product of two large primes because they are harder to factorize than integers composed of three or more big primes. Thus, this algorithm factorizes easily any integer n which is the product of two primes. The big advantage of this algebraic algorithm is the reduction of the factoring problem to the problem of order-finding.
Actually, to factorize an integer n, we don't necessairly need $\phi(n)$; the only thing we need is the smallest even r such that $a^r = 1 \bmod (n)$. Then, we factorize the equation $a^r - 1 = 0 \bmod (n)$.
The factors of n will be given by calculating the greatest common divisorof each factor of the following expression $(a^{r/2} - 1) \times (a^{r/2} + 1) = 0 \bmod (n)$ with n.

3 Quantum information

In this part, I will explain the mechanism of the magic box used above to find the period of the function. This magic box is actually a quantum computer. Quantum computers are much more powerful than classical computers because they use quantum laws of physics instead of classical laws. Quantum computers exploit two features of quantum mechanics: superposition and parallelism. The following sections will shine some light on the mathematical aspect of quantum computers.

3.1 Formalism and linear algebra

3.1.1 Definitions

3.1.1.1 Bits and Qubits

Classically, computers use what we call bits to store data. A classical **bit** is a state of a classical system that might be represented by '1' and '0'. For integers, we convert them from base 10 into base 2, the string of bits that represents the decimal 9 is: 1001.

In the otherhand, a **Qubit** is a quantum system whose state lies in a two dimensional **Hilbert Space** $\mathcal{H}$. In other words, a qubit is a vector in a two dimentional Hilbert space.

3.1.1.2 Hilbert Space

A two dimensional **Hilbert space** is a two dimensional vector space over **C** spanned by the orthonormal basis $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ with a complex valued inner product.

3.1.1.3 Bra 'c' ket notation (Dirac notation)

The elements of the Hilbert space are presented by either **bras** or **kets**. **Bras** define elements that are expressed as a column vector and **kets** define the row vectors. The ket $|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$ and the bra $\langle\psi| = (a^*, b^*)$ where a^* is the complex conjugate of a. The inner product is defined $\langle\psi|\psi\rangle = (a^*, b^*)\begin{pmatrix} a \\ b \end{pmatrix} = |a|^2 + |b|^2$ and the outer product $|\psi\rangle\langle\psi| = \begin{pmatrix} a \\ b \end{pmatrix}(a^*, b^*) = \begin{pmatrix} a^2 & ab \\ ba & b^2 \end{pmatrix}$.

A quantum state $|\psi\rangle$ is a linear combination of the kets $|0\rangle$ and $|1\rangle$:

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

In other words, the general quantum state $|\psi\rangle$ is the superposition of the pure states $|0\rangle$ and $|1\rangle$. The coefficients a and b are complex numbers such that $|a|^2 + |b|^2 = 1$. This last condition is a the normalisation condition because we need the thing that might be a particle for example to be somewhere in space, thus the inner product:

$$\langle\psi|\psi\rangle = (a^*, b^*)\begin{pmatrix} a \\ b \end{pmatrix} = |a|^2 + |b|^2 = 1$$

When we take a measurement on the quantum state of the particle $|\psi\rangle$ we can observe the state $|0\rangle$ with probability $|a|^2$ and the pure state $|1\rangle$ with probability $|b|^2$, but only one of the states shows up at the time

3.1.1.4 Operators

One can define a **linear operator** on a ket space $\mathcal{H}$ as a homomorphism

of $\mathcal{H}$ into $\mathcal{H}$. For our purpose, we assume all operators $\mathcal{O}$ in $\mathcal{H}_n$ to be $n \times n$ invertible matrices. One nice property that links the ket space to bra space is:

$$(\mathcal{O}|\psi\rangle)^\dagger = \langle\psi|\mathcal{O}^\dagger$$

An interesting operator used a lot in computer science is known as Hadamard operator (Hadamard Quantum Gate). Hadamard gate in $\mathcal{H}^{\otimes n}$ is the $2^n \times 2^n$ matrix $H = \frac{1}{\sqrt{2^n}} \begin{pmatrix} 1 & 1 \\ 1 & \text{-}1 \end{pmatrix}^{\otimes n}$. Notice that $H^2 = I$.

Applying a Hadamard operator on a pure qubit expands this pure qubit which might be a system of n particles into a superposition with equal probability of all the 2^n pure qubits formed by these n particles. Here is an example that shows the task of the Hadamrd gate. Let the pure qubit $|01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$

$$H_4|01\rangle = (H \otimes H) \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2^2}} \begin{pmatrix} 1 & 1 \\ 1 & \text{-}1 \end{pmatrix} \otimes \frac{1}{\sqrt{2^2}} \begin{pmatrix} 1 & 1 \\ 1 & \text{-}1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$= \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \text{-}1 & 1 & \text{-}1 \\ 1 & 1 & \text{-}1 & \text{-}1 \\ 1 & \text{-}1 & \text{-}1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1 \\ \text{-}1 \\ 1 \\ \text{-}1 \end{pmatrix}$$

$$= \frac{1}{4} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \frac{1}{4} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \frac{1}{4} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} - \frac{1}{4} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \frac{1}{4}|00\rangle - \frac{1}{4}|01\rangle + \frac{1}{4}|10\rangle - \frac{1}{4}|11\rangle$$

3.1.2 Tensor product

Given two 2×2 matrices $A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$ and $B = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix}$, their tensor product written $A \otimes B$ is the following 4×4 matrix $\begin{pmatrix} a_1 B & a_2 B \\ a_3 B & a_4 B \end{pmatrix}$

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

As an example, given the vector $a = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and the vector $b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, one can calculate the tensor product $a \otimes b =$ and find the following vector $\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$

3.1.3 n-Qubit system

In the Dirac notation, the pure state $|0\rangle$ is the state of one particle or "thing". It can represent, for example, the spin up of an electron or the vertical polarization of light...etc. How about two particles? Given two particles that can both of them be in the state $|0\rangle$ or the state $|1\rangle$, I will have 4 degrees of freedom, $|0\rangle|0\rangle$, $|0\rangle|1\rangle$, $|1\rangle|0\rangle$, $|1\rangle|1\rangle$. This implies that my vector space is a 4-dimensional Hilbert space: $\mathcal{H}_4 = \mathcal{H} \otimes \mathcal{H} = \mathcal{H}^{\otimes 2}$. We have the two systems of one particle that become one system of two particles, i.e.:

$$|1\rangle|0\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle$$

The 4-dimensional Hilbert space is spanned by the pure states (vectors):

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \text{ and } |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

For the 2^n-dimensional Hilbert space $\mathcal{H}^{\otimes n}$, its span would be the unit vectors:

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

.

3.2 Quantum computer vs. classical computer

Classical computers use bits to strore data. A string of n bits can store at most n informations; but n qubits can store up to 2^n informations. The n-qubits system is a system in the superposition of the 2^n different pure quantum states made of these n-qubits. For examples the system of 3-qubits $|\phi\rangle$ is a vector with 8 complex valued entries:

$$|\phi\rangle = a_0|000\rangle + a_1|001\rangle + a_2|010\rangle + a_3|011\rangle + a_4|100\rangle + a_5|101\rangle + a_6|110\rangle + a_7|111\rangle$$

$$|\phi\rangle = a_0|0\rangle + a_1|1\rangle + a_2|2\rangle + a_3|3\rangle + a_4|4\rangle + a_5|5\rangle + a_6|6\rangle + a_7|7\rangle$$

So the 3-qubits system can give us up to 8 informations. The most interesting thing about qubits is the fact that it is impossible according to quantum mechanics to know the 8 informations. The only thing we can do with these qubits is to manipulate them and extract some kinds of informations such as the period.

3.3 Fourier Transforms

In the Shor's algorithm, the use of QFT (Quantum Fourier Transform) is essential. What is a QFT? Quantum Fourier Transform is a Discrete Fourier Transform applied on qubits. For a n-qubits system we can apply the QFT on any pure qubit $|x\rangle$:

$$|x\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{xy}{2^n}} |y\rangle$$

Let $N = 2^n$,

$$|x\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \frac{xy}{N}} |y\rangle$$

Discrete Fourier transform maps uniform periodic states with period r, to a periodic states with period $\frac{N}{r}$. I will give an example to make clear the previous affirmation. Consider

$$f(x) = \mathbf{Z}_6 \longrightarrow \mathbf{Z}_6$$

Where $\mathbf{Z}_6$ is the ring with elements $\{0, 1, 2, 3, 4, 5\}$ and $\mathbf{R}$ is the set of real numbers. Applying the Discrete Fourier Transform on $f(x)$ will give us these equations:

$$f(0) \longrightarrow e^{2\pi i\frac{0\cdot 0}{6}} f(0)+e^{2\pi i\frac{0\cdot 1}{6}} f(1)+e^{2\pi i\frac{0\cdot 2}{6}} f(2)+e^{2\pi i\frac{0\cdot 3}{6}} f(3)+e^{2\pi i\frac{0\cdot 4}{6}} f(4)+e^{2\pi i\frac{0\cdot 5}{6}} f(5)$$

$$f(1) \longrightarrow e^{2\pi i\frac{1\cdot 0}{6}} f(0)+e^{2\pi i\frac{1\cdot 1}{6}} f(1)+e^{2\pi i\frac{1\cdot 2}{6}} f(2)+e^{2\pi i\frac{1\cdot 3}{6}} f(3)+e^{2\pi i\frac{1\cdot 4}{6}} f(4)+e^{2\pi i\frac{1\cdot 5}{6}} f(5)$$

$$f(2) \longrightarrow e^{2\pi i\frac{2\cdot 0}{6}} f(0)+e^{2\pi i\frac{2\cdot 1}{6}} f(1)+e^{2\pi i\frac{2\cdot 2}{6}} f(2)+e^{2\pi i\frac{2\cdot 3}{6}} f(3)+e^{2\pi i\frac{2\cdot 4}{6}} f(4)+e^{2\pi i\frac{2\cdot 5}{6}} f(5)$$

$$f(3) \longrightarrow e^{2\pi i\frac{3\cdot 0}{6}} f(0)+e^{2\pi i\frac{3\cdot 1}{6}} f(1)+e^{2\pi i\frac{3\cdot 2}{6}} f(2)+e^{2\pi i\frac{3\cdot 3}{6}} f(3)+e^{2\pi i\frac{3\cdot 4}{6}} f(4)+e^{2\pi i\frac{3\cdot 5}{6}} f(5)$$

$$f(4) \longrightarrow e^{2\pi i\frac{4\cdot 0}{6}} f(0)+e^{2\pi i\frac{4\cdot 1}{6}} f(1)+e^{2\pi i\frac{4\cdot 2}{6}} f(2)+e^{2\pi i\frac{4\cdot 3}{6}} f(3)+e^{2\pi i\frac{4\cdot 4}{6}} f(4)+e^{2\pi i\frac{4\cdot 5}{6}} f(5)$$

$$f(5) \longrightarrow e^{2\pi i\frac{5\cdot 0}{6}} f(0)+e^{2\pi i\frac{5\cdot 1}{6}} f(1)+e^{2\pi i\frac{5\cdot 2}{6}} f(2)+e^{2\pi i\frac{5\cdot 3}{6}} f(3)+e^{2\pi i\frac{5\cdot 4}{6}} f(4)+e^{2\pi i\frac{5\cdot 5}{6}} f(5)$$

If we assume the function $f(x)$ to be periodic with period 2, i.e., $f(0) = f(2) = f(4)$ and $f(1) = f(3) = f(5)$, the previous equations become:

$$f(0) \longrightarrow 3 \cdot f(0) + 3 \cdot f(1)$$

$$f(1) \longrightarrow 0 \cdot f(0) + 0 \cdot f(1)$$

$$f(2) \longrightarrow 0 \cdot f(0) + 0 \cdot f(1)$$

$$f(3) \longrightarrow 3 \cdot f(0) - 3 \cdot f(1)$$

$$f(4) \longrightarrow 0 \cdot f(0) + 0 \cdot f(1)$$

$$f(5) \longrightarrow 0 \cdot f(0) + 0 \cdot f(1)$$

For a periodic function with period 3, i.e., $f(0) = f(3)$, $f(1) = f(4)$ and $f(2) = f(5)$ the final output becomes:

$$f(0) \longrightarrow 2 \cdot f(0) + 2 \cdot f(1) + 2 \cdot f(2)$$

$$f(1) \longrightarrow 0 \cdot f(0) + 0 \cdot f(1) + 0 \cdot f(2)$$

$$f(2) \longrightarrow 2 \cdot f(0) + e^{\frac{2\pi i}{3}} \cdot f(1) + e^{\frac{4\pi i}{3}} \cdot f(2)$$

$$f(3) \longrightarrow 0 \cdot f(0) + 0 \cdot f(1) + 0 \cdot f(2)$$

$$f(4) \longrightarrow 2 \cdot f(0) + e^{\frac{4\pi i}{3}} \cdot f(1) + e^{\frac{2\pi i}{3}} \cdot f(2)$$
$$f(5) \longrightarrow 0 \cdot f(0) + 0 \cdot f(1) + 0 \cdot f(2)$$

As we can see in the previous sets of equations, for a periodic function of period 2, we have non-zero factors of probablity at $f(0) = f(0 \cdot \frac{N}{r}) = f(0 \cdot \frac{6}{2})$ and $f(3) = f(1 \cdot \frac{N}{r}) = f(1 \cdot \frac{6}{2})$. Therefore the QFT extracts in some sens the period of a periodic function.
Discrete Fourier Transform transformation is a matrix:

$$\mathcal{F}_{ab} = \frac{1}{\sqrt{N}} e^{2\pi i \frac{ab}{N}}$$

3.4 Shor's Analysis

Given an integer $N = pq$ we want to factorize, we let $n = \lceil Log_2 N \rceil$ be the number of qubits needed to store the integer N. We take two strings of n-qubits, the first string holds the value $x \in S = \{0, 1, 2, ..., 2^n - 1\}$ in binary; the second string holds the value of $f(x) = a^x \bmod(N)$ in binary also. The two strings $|x\rangle$ and $|f(x)\rangle$ have n-qubits each.
The quantum parallelism is the property of distribution on matrices. Let $A, B_0, B_1, B_2, ..., B_{\mathcal{L}}$ be $m \times m$ matrices:

$$A \sum_{k=0}^{\mathcal{L}} B_k = \sum_{k=0}^{\mathcal{L}} AB_k$$

For our purpose, A can be any operator or function $f : S \longrightarrow S$ and B_x are $|x\rangle$ such that $x \in S$.
I apply the Hadamed gate on $|N\rangle$ to obtain:

$$|N\rangle \xrightarrow{H^{\otimes n}} |\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle$$

i.e. a an equal probability superposition of all the pure states.
After that, I use the quantum parallelism to calculate $|f(x)\rangle$ for each $|x\rangle$. I will introduce this notation $|x, f(x)\rangle$ instead of $|x\rangle|f(x)\rangle$. Thus my quantum system $|\psi_1\rangle$ will be:

$$|\psi_1\rangle \xrightarrow{U_{f(x)}} |\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k, f(k)\rangle$$

I take a measurement on the second qubit $|f(x)\rangle$, I will get some value y_0.

Quantum mechanics says that the first quantum system has to collapse into a superposition of all values of x such that $f(x) = y_0$. Let $K = \lceil \frac{N}{r} \rceil$ where N is my integer I want to factorize and r the order of the periodic function $f(x) = a^x \bmod(N)$, $\gcd(a, N) = 1$. After taking a measurement of the second string the last sum $|\psi_2\rangle$ becomes:

$$|\psi_2\rangle \overset{measure}{\longrightarrow} |\psi_3\rangle = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} |x_0 + kr, y_0\rangle$$

where $0 \leq x_0 < r$. Let me keep the first string only, so that my superposition $|\psi_3\rangle$ becomes:

$$|\psi_3\rangle \longrightarrow |\psi_4\rangle = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} |x_0 + kr\rangle$$

At this moment, the QFT comes into play. Remember that the QFT does the following transformation $|x\rangle \overset{QFT}{\longrightarrow} \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{xy}{2^n}} |y\rangle$. When we apply it to $|\psi_4\rangle$ we get:

$$|\psi_4\rangle \overset{QFT}{\longrightarrow} |\psi_5\rangle = \frac{1}{\sqrt{r}} \sum_{\ell=0}^{r-1} e^{2\pi i \frac{x_0 \ell}{r}} |\ell \frac{N}{r}\rangle$$

We are almost done, all we have to do now is to take a measurement on the quantum system $|\psi_5\rangle$; assume we get c as a value:

$$c = \ell \frac{N}{r} \Leftrightarrow \frac{c}{N} = \frac{\ell}{r}$$

for some $0 < \ell < r$. All we need is having $\gcd(\ell, r) = 1$ so that we simpify the fraction $\frac{c}{N}$ and read the value of the period r in the denominator. Otherwise, we repeat the process again, take another measurement and get a new value of c that gives us ℓ relatively prime to r.

4 Example

Here is an example of how to factorize an integer. I would like to factorize the integer $N = 57$. $\hat{\text{I}}$ choose $a = 11$ randomly and I use the magic box to have the value of $r = 6$. I apply the algebraic algorithm

$$a^r - 1 = 0 \Leftrightarrow 11^6 - 1 = 0$$

$$(a^{r/2} - 1)(a^{r/2} + 1) = 0 \Leftrightarrow (11^3 - 1)(11^3 + 1) = 0$$

$$(a^{r/2} - 1) \neq 0, (a^{r/2} + 1) \neq 0 \bmod N \Leftrightarrow (11^3 - 1) \neq 0, (11^3 + 1) \neq 0 \bmod 57$$

$$\gcd((a^{r/2}-1) \mod N, N) = p \Leftrightarrow \gcd((11^3-1) \mod 57, 57) = 19$$
$$\gcd((a^{r/2}+1) \mod N, N) = q \Leftrightarrow \gcd((11^3+1) \mod 57, 57) = 3$$

Finanlly we have our factors of $N = 57$ that are 3 and 19

Conclusion

In this report, I present Shor's Quantum Factorization Algorithm. This very powerful algorithm, uses the laws of quantum mechanics and some topics of algebraic number theory. This quantum algorithm factorizes integers in a polynomial time [2]. The use of Quantum Fourier Transforms is crutial to get the period of the function $f(x) = a^x \bmod(N)$.Shor's algorithm destroys nicely the RSA cryptosystem. Unfortunately, we must build quantum computers to be able to use this algorithm, but many scientists work hardly on the question.

Acknowledgements

I have been studying this algorithm the whole summer 2005. During the last four months, I studied and understood many theories related to quantum mechanics or algebraic number theory. Most of what I have learned is not presented in this report and I owe all what I have done to professor Henri Darmon the coordinator of this project who supported me and helped me during this last year. I would like to thank also my collegues and friends, they supported me a lot: Anne-Sophie Charest, Mathieu Guay-Paquet, Ilya Hekimi, Louis-Francois Preville-Ratelle and Mireille Schnitzer. My friends also Frédéric Caron and Leonid Chindelevitch. This experience thought me the way we do research and the art of proving theorems, unfortunately I wasn't able to prove by myself all the theorems I met during my project but I learned many tricks that might very useful in future.

References

[1] Thomas W. Hungerford, *Abstract Algebra*

[2] P.W. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, 1994*