

Shor's Algorithm

Roger Herrigel
Supervisor: Wojciech De Roeck

April 14 2008

- ▶ Motivation
- ▶ Classical part
 - ▶ Order-finding problem
 - ▶ Number theory
 - ▶ Procedure
 - ▶ Modular exponentiation
- ▶ Quantum part
 - ▶ Fourier transform
 - ▶ Phase estimation
 - ▶ Order-finding
 - ▶ Continued fraction algorithm
 - ▶ Procedure
- ▶ Related problems
 - ▶ Period-finding
 - ▶ Discrete logarithm

Motivation

Shor's algorithm:

- ▶ Algorithm for factoring a non-prime number N of L bits.

Motivation:

- ▶ Quantum algorithms faster than classical algorithms:
 - ▶ Algorithm based on the quantum Fourier transform.
 - ▶ Quantum search algorithm (Grover).
 - ▶ Quantum simulation.
- ▶ Cracking RSA.

Complexity:

- ▶ Classical computer: $O\left(e^{L^{\frac{1}{3}}(\log L)^{\frac{2}{3}}}\right)$
- ▶ Quantum computer: $O(L^3)$

Shor's algorithm

Input: Non-prime number N .
Output: Non-trivial factor of N .

Shor's algorithm can be split into two parts:

- a) A reduction of the factoring problem to a problem of order-finding.
→ Classical computer
- b) An algorithm solving the order-finding problem.
→ Quantum computer

Order-finding problem

Definition

The **order** of an element x in a group G , is the least integer r , such that $x^r = 1_G$.

Definition

Integers x and y are **co-prime** if $\gcd(x, y) = 1$.

The group \mathbb{Z}_N^* :

- ▶ Elements: subset of $\{0, 1, \dots, N\}$ which are co-prime to N .
- ▶ Group operation: multiplication modulo N .

The order-finding problem:

Given: x and N , $x < N$ and $\gcd(x, N) = 1$.

⇒ The order of x in \mathbb{Z}_N^* is the least positive integer, r , such that $x^r = 1 \pmod{N}$

Theorem

Suppose:

- ▶ N is a non-prime number
- ▶ x a solution of $x^2 = 1 \pmod{N}$
- ▶ $x \not\equiv 1 \pmod{N}$ and $x \not\equiv -1 \pmod{N}$

\Rightarrow one of $\gcd(x - 1, N)$ and $\gcd(x + 1, N)$ is a non-trivial factor of N .

Proof: Blackboard.

Reduction of the factoring problem to a problem of order-finding:

- ▶ If y is co-prime to N .
 - ▶ y has an even order r ($y^r = 1 \pmod{N}$)
 - ▶ $y^{r/2} \not\equiv \pm 1 \pmod{N}$
- $\Rightarrow x \equiv y^{r/2} \pmod{N}$
- ▶ is a solution to $x^2 = 1 \pmod{N}$
 - ▶ and $x \not\equiv \pm 1 \pmod{N}$

Thus x satisfies all conditions of the theorem.

Classical algorithm

1. Choose a (random) number $y < N$.
 2. If $\gcd(y, N) > 1$ return $\gcd(y, N)$ and we are done.
 3. Use the order-finding subroutine to calculate the order r of y , modulo N . \rightarrow Quantum computer
 4. If r is even and $y^{r/2} \not\equiv -1 \pmod{N}$:
Set $x = y^{r/2} \pmod{N}$ and compute $\gcd(x - 1, N) > 1$ and $\gcd(x + 1, N) > 1$, one of them is a non-trivial factor of N .
Return this factor.
- What if r is odd?

Theorem

- ▶ N odd
 - ▶ N not of the form p^c where p is a prime number
 - ▶ x random chosen from \mathbb{Z}_N^* ($\gcd(x, N) = 1$)
 - ▶ r order of x modulo N
- $\Rightarrow p(r \text{ even and } x^{r/2} \not\equiv \pm 1 \pmod{N}) \geq \frac{1}{2}$

Summery of the classical part

1. If N is even return 2.
 2. Check if $N = p^c$. If so return p .
 3. Choose a random number $y < N$.
 4. If $\gcd(y, N) > 1$ return $\gcd(y, N)$ and we are done.
 5. Use the order-finding subroutine to calculate the order r of y , modulo N . → Quantum computer
 6. If r is even and $y^{r/2} \not\equiv -1 \pmod{N}$:
Set $x = y^{r/2} \pmod{N}$ and compute $\gcd(x - 1, N) > 1$ and $\gcd(x + 1, N) > 1$, one of them is a non-trivial factor of N .
Return this factor.
- ▶ Algorithm has a probability of success greater than $1/2$.
 - ▶ Step 5 has a probability of success smaller than $1!$

Complexity

1. If N is even return 2. $O(1)$
2. Check if $N = p^c$. If so return p . $O(L^3)$
3. Choose a random number $y < N$. $O(1)$
4. If $\gcd(y, N) > 1$ return $\gcd(y, N)$ and we are done. **Euclidean algorithm** $O(L^3)$
5. Use the order-finding subroutine to calculate the order r of y , modulo N . \rightarrow **Quantum computer**
6. If r is even and $y^{r/2} \not\equiv -1 \pmod{N}$:
Set $x = y^{r/2} \pmod{N}$ and compute $\gcd(x - 1, N) > 1$ and $\gcd(x + 1, N) > 1$, one of them is a non-trivial factor of N .
Return this factor. **Modular exponentiation** $O(L^3)$ and **Euclidean algorithm** $O(L^3)$

Quantum Fourier transform

The **discrete Fourier transform**:

Acting on vector of complex numbers x_0, \dots, x_{n-1}

$$y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j e^{i2\pi jk/n}.$$

Acting on orthonormal basis $|0\rangle, \dots, |n-1\rangle$

$$|j\rangle \rightarrow \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} e^{i2\pi jk/n} |k\rangle.$$

Acting on an arbitrary state:

$$\sum_{j=0}^{n-1} x_j |j\rangle \rightarrow \sum_{k=0}^{n-1} y_k |k\rangle.$$

Phase estimation I

Problem:

Given: unitary operator U and an eigenvector $|u\rangle$

$$U|u\rangle = e^{i2\pi\phi}|u\rangle$$

Goal: estimation of ϕ .

Oracles:

- ▶ Preparation of state $|u\rangle$
- ▶ Controlled U^{2^j} gates $j \geq 0$.

Phase estimation II

Preparation of registers:

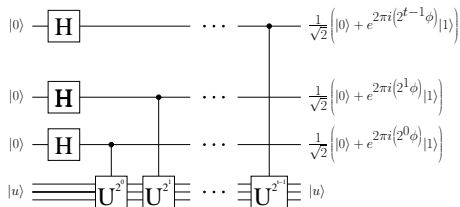
Register	initial state	bits
1	$ 0\rangle$	t qubits
2	$ u\rangle$	as many qubits as necessary

If we want:

- ▶ ϕ accurate to n bits
 - ▶ success of order finding procedure: $1 - \epsilon$
- $\Rightarrow t = n + \lceil \log \left(2 + \frac{1}{2\epsilon} \right) \rceil$

Phase estimation III

1. Apply following circuit

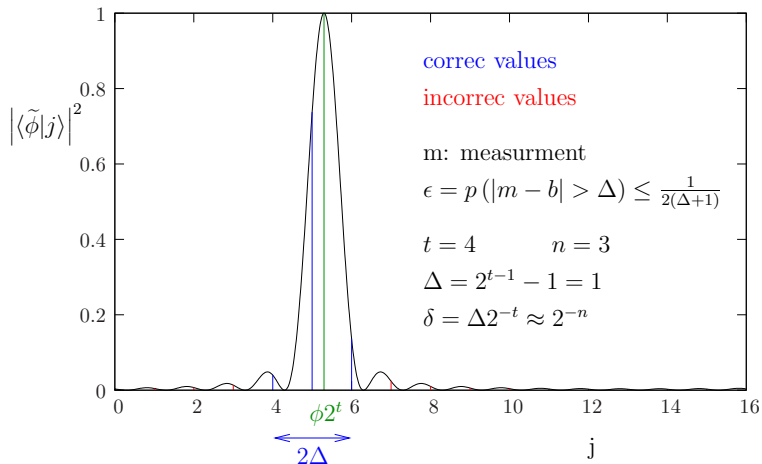


$$\begin{aligned} |0\rangle|u\rangle &\rightarrow \frac{1}{\sqrt{2^t}} \left(|0\rangle + e^{2\pi i 2^{t-1} \phi} |1\rangle \right) \left(|0\rangle + e^{2\pi i 2^{t-2} \phi} |1\rangle \right) \dots \left(|0\rangle + e^{2\pi i 2^0 \phi} |1\rangle \right) |u\rangle \\ &= \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi i \phi j} |j\rangle |u\rangle. \end{aligned}$$

2. Apply inverse Fourier transform $\sum_{j=0}^{2^t-1} e^{2\pi i \phi k} |j\rangle |u\rangle \rightarrow |\tilde{\phi}\rangle |u\rangle$.

3. Measure register 1.

Observing probabilities



Phase estimation III

1. initial state

$$|0\rangle|u\rangle$$

2. Hadamard gates

$$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|u\rangle$$

3. controlled U^j gates

$$\begin{aligned} &\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle U^j |u\rangle \\ &= \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{i2\pi j\phi_u} |j\rangle|u\rangle \end{aligned}$$

4. apply inverse Fourier transform

$$\rightarrow |\widetilde{\phi_u}\rangle|u\rangle$$

5. measure first register

$$\rightarrow \phi \pm \delta \quad \delta \approx 2^{-n}$$

$$P_{\text{success}} \geq (1 - \epsilon)$$

Order finding

Problem: (reminder)

Input: x and N , $x < N$ and $\gcd(x, N) = 1$.

Output: least positive integer, r , such that $x^r = 1 \pmod{N}$

Order finding is phase estimation applied to the operator

$$U(x, N) |y\rangle = |xy \pmod{N}\rangle$$

Eigenstates: $|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-2\pi i s k}{r}\right) |x^k \pmod{N}\rangle$

Eigenvalues: $\exp(2\pi i s/r)$

Requirements:

- ▶ Preparation of state $|u_s\rangle$.
- ▶ Controlled U^{2^j} gates $j \geq 0$.

Preparation of initial state

Operator: $U(x, N) |y\rangle = |xy \pmod N\rangle$

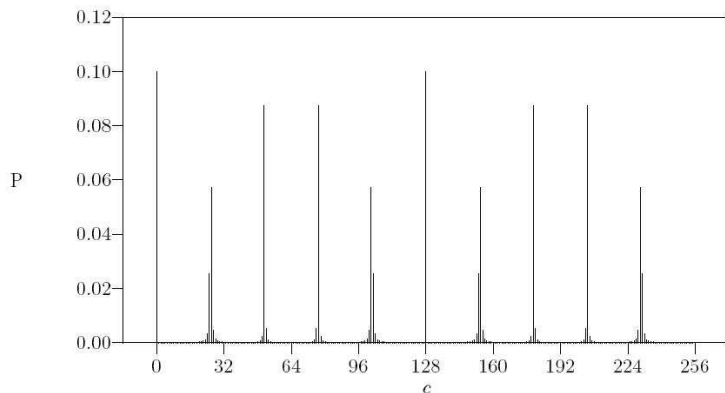
Eigenstates: $|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-i2\pi sk}{r}\right) |x^k \pmod N\rangle$

Eigenvalues: $\exp(2\pi is/r)$

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle$$

\Rightarrow Preparation register 2 in state $|1\rangle \in \text{span}\{u_s\}$.

Observing probabilities



Range: $[0, 2^t = 256)$

Number of peaks: $r = 10$

Distant between peaks: $\frac{2^t}{r}$

Order finding

1. initial state $|0\rangle|1\rangle$
2. create superposition $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|1\rangle$
3. apply $U_{x,N}$ $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|x^j \bmod N\rangle$
 $= \frac{1}{\sqrt{r2^t}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{2\pi i s j / r} |j\rangle|u_s\rangle$
4. apply inverse Fourier transform $\rightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\widetilde{s/r}\rangle|u_s\rangle$
5. measure first register $\rightarrow s/r \pm \delta$
6. apply cont. frac. algorithm $\rightarrow r$

Order finding

1. initial state $|0\rangle|1\rangle$
2. create superposition $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|1\rangle$
3. apply $U_{x,N}$ $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|x^j \bmod N\rangle$
 $= \frac{1}{\sqrt{r2^t}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{2\pi i s j / r} |j\rangle|u_s\rangle$
4. apply inverse Fourier transform $\rightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\widetilde{s/r}\rangle|u_s\rangle$
5. measure first register $\rightarrow s/r \pm \delta$
6. apply cont. frac. algorithm $\rightarrow r$

Continued fraction algorithm

Definition

Continued fraction is an expression of the form

$$[a_0, \dots, a_M] \equiv a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\dots + \frac{1}{a_M}}}}$$

where a_0, \dots, a_M are positive integers.

- ▶ Any rational number can be represented as an continued fraction.
- ▶ For a rational number p/q , where p and q are L bits numbers
 - ⇒ $M = O(L)$.
 - ⇒ Continued fraction $[a_0, \dots, a_M]$ can be computed using $O(L^3)$ operations.

Definition

We say that $[a_0, \dots, a_m]$ is a **convergent** of $[a_0, \dots, a_M]$, for $m < M$.



Continued fraction algorithm example

Example

$$\begin{aligned}\frac{31}{13} &= 2 + \frac{5}{13} = 2 + \frac{1}{\frac{13}{5}} = 2 + \frac{1}{2 + \frac{3}{5}} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{2}{3}}} \\ &= 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}} = [2, 2, 1, 1, 2]\end{aligned}$$

The 4rd convergent:

$$[2, 2, 1, 1] = 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{1}}} = \frac{7}{3} \approx \frac{31}{13} + 0.05$$

Continued fraction algorithm

Theorem

Suppose that s/r is a rational number such that

$$\left| \frac{s}{r} - \phi \right| \leq \frac{1}{2r^2}$$

Then s/r is a convergent of the continued fraction for ϕ . Thus can be computed in $O(L^3)$ steps.

Choose: $t = 2L + 1 + \lceil \log(2 + \frac{1}{2\epsilon}) \rceil$

\Rightarrow We have ϕ approximately s/r up to $2L + 1$ bit

$\Rightarrow \left| \frac{s}{r} - \phi \right| \leq 2^{-2L-1} \leq \frac{1}{2N^2} \leq \frac{1}{2r^2}$

► for $r < N$ and $\left| \frac{s}{r} - \phi \right| \leq \frac{1}{2N^2} \Rightarrow s/r$ is unique

Problems of the continued fraction algorithm

Problem: s and r may have a common factor.

Example: $s = 5$ and $r = 10 \Rightarrow s/r = 1/2$
we get: $s' = 1$ and $r' = 2$

Solution:

- ▶ Repeat phase estimation and continued fraction method twice, obtaining r'_1, s'_1 and r'_2, s'_2 .

If $\gcd(s_1, s_2) = 1 \Rightarrow r = \text{lcm}(r'_1, r'_2)$.

- ▶ If s_1 and s_2 are chosen uniformly from $(0, r - 1)$ then

$$\text{Prob}(\gcd(s_1, s_2) = 1) \geq \frac{1}{4}$$

Order finding

1. initial state $|0\rangle|1\rangle$
2. create superposition $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|1\rangle$
3. apply $U_{x,N}$ $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|x^j \bmod N\rangle$
 $= \frac{1}{\sqrt{r2^t}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{2\pi i s j / r} |j\rangle|u_s\rangle$
4. apply inverse Fourier transform $\rightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\widetilde{s/r}\rangle|u_s\rangle$
5. measure first register $\rightarrow s/r \pm \delta$
6. apply cont. frac. algorithm $\rightarrow r$

- ▶ The order r may be odd. Prob $\leq 1/2$
→ can be easily checked.
- ▶ The phase estimation may fail. Prob $\leq 1 - \epsilon$
→ can not be checked.
- ▶ r/s may not be irreducible.
Take two measurements and calculate $r = \text{lcm}(r'_1, r'_2)$, then
Prob $\leq 1/4$
→ can not be checked.

At the end we get a proposal for p and q such that $N = pq$.
Easy to check if this is true.

Period-finding

Given: $f(x+r) = f(x)$ for $0 < r < 2^L$
and black box $U|x\rangle|y\rangle = |x\rangle|f(x) \oplus y\rangle$.

Goal: find period r .

1. initial state $|0\rangle|0\rangle$
2. create superposition $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|0\rangle$
3. apply U $\rightarrow \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|f(x)\rangle$
 $\approx \frac{1}{\sqrt{r2^t}} \sum_{l=0}^{r-1} \sum_{x=0}^{2^t-1} e^{2\pi i l x / r} |x\rangle |\hat{f}(l)\rangle$
4. inverse Fourier transform $\rightarrow \frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} |\widetilde{l/r}\rangle |\hat{f}(l)\rangle$
5. measure first register $\rightarrow l/r \pm \delta$
6. apply cont. frac. algorithm $\rightarrow r$

Discrete logarithm

Problem: Given p prime number, a generator a of \mathbb{Z}_p^* and $b \in \mathbb{Z}_p^*$, we want to find s such that $a^s = b \pmod{p}$.

Consider the function:

$$f(x_1, x_2) = b^{x_1} a^{x_2} \pmod{p} = a^{sx_1 + x_2} \pmod{p}$$

$$\blacktriangleright f(x_1 + l, x_2 - ls) = f(x_1, x_2)$$

$$\Rightarrow \text{Period: } (l, -ls) \quad \Rightarrow \quad s$$

Discrete logarithm problem is a period-finding problem using the Oracle:

$$U|x_1\rangle|x_2\rangle|y\rangle = |x_1\rangle|x_2\rangle|f(x_1, x_2) \oplus y\rangle$$

The order r of a modulo p is $p - 1$.

Discrete logarithm

1. $|0\rangle|0\rangle|0\rangle$

2. $\rightarrow \frac{1}{2^t} \sum_{x_1=0}^{2^t-1} \sum_{x_2=0}^{2^t-1} |x_1\rangle|x_2\rangle|0\rangle$

3. $\rightarrow \frac{1}{2^t} \sum_{x_1=0}^{2^t-1} \sum_{x_2=0}^{2^t-1} |x_1\rangle|x_2\rangle|f(x_1, x_2)\rangle$

$$\approx \frac{1}{\sqrt{r}2^t} \sum_{l=0}^{r-1} \sum_{x_1=0}^{2^t-1} \sum_{x_2=0}^{2^t-1} e^{2\pi i(sl x_1 + lx_2)/r} |x_1\rangle|x_2\rangle|\hat{f}(ls, l)\rangle$$

$$= \frac{1}{\sqrt{r}2^t} \sum_{l=0}^{r-1} \left[\sum_{x_2=0}^{2^t-1} e^{2\pi i(l x_2)/r} |x_2\rangle \right] \left[\sum_{x_1=0}^{2^t-1} e^{2\pi i(s l x_1)/r} |x_1\rangle \right] |\hat{f}(ls, l)\rangle$$

4. $\rightarrow \frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} |\widetilde{sl/r}\rangle|\widetilde{l/r}\rangle|\hat{f}(l)\rangle$

5. $\rightarrow (sl/r \pm \delta, l/r \pm \delta)$

6. $\rightarrow s$

Summary

- ▶ The factoring problem can be split into two parts.
 - a) Reduction of the factoring problem to a problem of order-finding.
 - Classical computer
 - b) Algorithm solving the order-finding problem.
 - Quantum computer
- ▶ Order-finding problem is a phase estimation problem.
- ▶ Phase estimation can be solved efficiently on a Quantum computer.
- ▶ Related problems like period-finding and discrete logarithm.