New Algorithms for Factoring and Discrete Logarithm with NMR Quantum Computers

Noboru Kunihiro† and Shigeru Yamashita†*
†NTT Communication Science Laboratories, NTT Corporation, Atsugi, Japan
* Quantum Computation and Information, ERATO.
kunihiro@theory.brl.ntt.co.jp, ger@cslab.kecl.ntt.co.jp

Shor showed that a factoring problem and a discrete logarithm problem can be solved by a quantum Turing Machine in a probabilistic polynomial time in 1994. A NMR quantum computer is different from an ordinary quantum computer with respect to the phase of measurement. In an ordinary quantum computer, if we observe the superposition of states, we will find a pure state with probability depending on amplitudes. On the other hand, measurement in NMR quantum computers has the two features: (i) the outcome of the measurement does not become a pure state, but (ii) we observe a value depending on amplitudes. Because of (i), we cannot directly implement Shor's algorithm, which uses another measurement in addition to a final measurement.

In this report, we propose two algorithms that use NMR quantum computers with feature (ii). One is for factoring problem and the other is for solving a discrete logarithm problem. These algorithms seem much different from Shor's algorithms. The algorithms consist of two steps. First, a counting problem of a certain function is solved by using the output of the NMR quantum computer. Then, the problems are solved from the result of obtained in the first step by classical computers.

We define a *counting problem*. This problem is formulated as follows.

Definition 1 Let N variable function $f(x_1, \ldots, x_N)$ output 0 or 1. The counting problem of f is the problem for counting $|f^{-1}(1)|$, $\sum_{x_1,\ldots,x_N} f(x_1, \ldots, x_N)$, or the number of assignments satisfying $f = 1$.

The following algorithm solves the counting problem of f by NMR quantum computers.

Step 1 Apply the Walsh–Hadamard transformation to $|0, 0, \ldots, 0, 0\rangle$, which changes the state into

$$\frac{1}{\sqrt{2^N}} \sum_{x_1,\ldots,x_N} |x_1, \ldots, x_N, 0\rangle.$$

Step 2 Compute $f(x_1, x_2, \ldots, x_N)$ for all $x_1, \ldots, x_N$ and store the result in the last register.

$$\frac{1}{\sqrt{2^N}} \sum_{x_1,\ldots,x_N} |x_1, \ldots, x_N, f(x_1, x_2, \ldots, x_N)\rangle.$$

Step 3 Perform a measurement on the last register. Let θ be the value obtained. The number of assignments satisfying $f = 1$ is $2^{N-1}(1 + \theta)$.

We define the Euler ϕ function.

Definition 2 Let n be $p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$, $\phi(n)$ is defined as $\phi(n) = p_1^{e_1-1}(p_1 - 1)p_2^{e_2-1}(p_2 - 1)\cdots p_k^{e_k-1}(p_k - 1)$.

For $\phi(n)$, the following fact is widely known.

Fact 1 Let n be the composite number. If we know $\phi(n)$, we can factorize n in a probabilistic polynomial time.

Next, we define the function $f : \{0,1\}^N \to \{0,1\}$ as follows, where $N = \lceil \log_2 n \rceil$.

$$\begin{cases} f(x_1, \ldots, x_N) = 1 & \text{if } \gcd((x_1 x_2 \cdots x_N)_2, n) = 1 \\ & \quad \text{and } (x_1 x_2 \cdots x_N)_2 < n \\ f(x_1, \ldots, x_N) = 0 & \text{otherwise} \end{cases}$$

Note that $\sum_{x_1,\ldots,x_N} f(x_1, x_2, \ldots, x_N) = \phi(n)$

Algorithm for factoring

Step 1 Solve the counting problem of f by using the NMR quantum computers. So, we obtain $\phi(n)$.

Step 2 By using $\phi(n)$, factorize n.

Discrete Logarithm Problem Given a prime p, g and $a \in F_p$, find the positive smallest integer r such that $g^r \equiv a \pmod p$. Let q be the order of a cyclic group generated by g.

We can count the number, $N(w)$, of integers x ($0 \le x \le S-1$) such that $g^x \equiv a \cdot g^{-w} \pmod p$ for given integers w and S by using the NMR quantum computer. We propose an algorithm that solves the discrete logarithm problem by calling this as a subroutine at most $\log q$ times.

Algorithm for discrete logarithm

Step 1 Find $S = 2^N$ such that $\frac{S}{q} - C = \mathrm{O}(1)$ and $C + 1 - \frac{S}{q} = \mathrm{O}(1)$, where $C = \lfloor \frac{S}{q} \rfloor$.

Step 2 Find w such that $N(w) = C + 1$ and $N(w+1) = C$ by *binary search*. $N(w)$ can be calculated by using NMR quantum computers (in this case, $f(x)$ is set to 1 if $g^{x+w} \equiv a \pmod p$ and 0 otherwise). We can find such w by calling this algorithm at most $\log q$ times. Finally, set $r = w$.

It is easily verified that (i) $N(r) = C+1$ and $N(r+1) = C$ and (ii) no value except $w = r$ satisfies $N(w) = C + 1$ and $N(w + 1) = C$ and (iii) the binary search finds the correct answer r efficiently.