A Short Note on Shor's Factoring Algorithm

Harry Buhrman*
CWI
PO Box 94079
1090 GB Amsterdam
The Netherlands.
E-mail: buhrman@cwi.nl.

Abstract

This note shows that Shor's algorithm for factoring in polynomial time on a quantum computer can be made to work with zero error probability.

1 Result

The algorithm described by Shor [Sho94] is a bounded error algorithm for factorization of integers. We will show how to extend the algorithm to one that operates with zero error probability.

In order to extend the algorithm we will use the following lemma by Fellows and Koblitz [FK92].

Lemma 1.1 *Given an odd number p, and $q_1, \ldots, q_k$, the prime factorization of $p-1$, one can generate in polynomial time a certificate that either p is prime or composite.*

Theorem 1.2 *One can extend the bounded error algorithm for factorization [Sho94] to work with zero error probability. (i.e. the algorithm never makes a mistake and only with very low probability yields an inconclusive outcome).*

Proof. The algorithm described by Shor yields, on input x, with high probability a factor l of x. Note that l is not necessary a prime number. However by repeating the algorithm on l one can obtain with arbitrary high probability the prime factorization, $p_1, \ldots, p_k$ of x. The extension we propose now uses Lemma 1.1 to verify that the given prime factorization is indeed correct. It is easy to check that the product yields x. What remains is to check that all the p_i are indeed prime. In order to check that p_i is prime we do the following. Use Shor's algorithm to generate (with high probability)

*Partially supported by NWO through NFI Project ALADDIN number NF 62-376.

the prime factorization $q_1^i, \ldots, q_l^i$ of p_i. Next recursively compute for each q_j^i the prime factorization of $q_j^i - 1$, until the numbers are so small that with brute force one can determine whether the prime factorization is correct. Then use Lemma 1.1 to generate a certificate that the number whose factorization is computed is indeed prime, working all the way up to the original prime factorization of x. It is clear that with high probability all the certificates will be found and hence that each of the $p_1, \ldots, p_k$ is prime. Moreover the algorithm runs in quantum polynomial time since $O(\log^2(x))$ (prime) numbers will be generated by the above procedure. □

We note that in order to make the factoring algorithm of Shor have zero error probability one could also use the algorithm of Adleman and Huang [AH92] to check with zero error probability that the outcome of Shor's algorithm is a prime factorization. We feel however that the above construction is much simpler.

References

[AH92] L. M. Adleman and M.-D. A. Huang. *Primality Testing and Abelian Varieties over Finite Fields*, volume 1512 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1992.

[FK92] M.R. Fellows and N. Koblitz. Self-witnessing polynomial time complexity and prime factorization. In *Proceedings Structure in Complexity Theory 7th annual conference*, pages 107 – 110, 1992.

[Sho94] P.W. Shor. Algorithms for quantum computation: Discrete log and factoring. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, pages 20 – 22, 1994.