Lecture 7

Lecturer: Scott Aaronson

1 Short review and plan for this lecture

In previous lectures we started building up some intuition into the way quantum algorithms work, and we have seen examples (Bernstein-Vazirani, Simon) where, excitingly, quantum computation could be performed with less resources (queries/time) than in the classical setting. Simon's algorithm lead to an oracle separation result between BPP and BQP, namely that there exists a language A, such that $BPP^A \neq BQP^A$. While these types of statements are deemed fundamentally interesting by the complexity theory community, they are not going to attract the attention of the NSA. That is why some turn to looking into ways of breaking the RSA cryptographic system instead! If we had a quantum computer, Shor's factoring algorithm would give us a method to efficiently steal people's credit card numbers. In today's lecture we will ready ourselves for this eventual opportunity and broadly discuss the main features of Shor's factoring algorithm. Subsequently, we will place the core ideas of both Simon's and Shor's algorithms into a more general framework, namely the Hidden Subgroup Problem(HSP).

2 General overview of Shor's factoring algorithm

Imagine that you would like to decompose a really large number N into its really large prime factors, in very little time. While we do not know how to do that classically, Peter Shor discovered that the task is possible in the quantum world.

Theorem 1 (Shor '94) FACTORING $\in BQP$.

A fundamental misconception in solving the factoring question is that quantum computers can try in parallel all possible integers. Unfortunately, that is not the case, and we need to delve more into the specific structure of this type of problems. Shor's algorithm has two modular components: a classical part and a quantum subroutine. The classical part draws from Miller's insight from the '70s that factoring reduces to finding the period of a function, which is then achieved using Quantum Fourier Transforms.

To begin with, let $N = p\ q$, $G = \{x \mod N | \gcd(N, x) = 1\}$ and denote by $ord(x)$ the order of x in G, i.e. the smallest integer r s.t. $x^r = 1 \mod N$. G is a group under the operation of multiplication and contains $\phi(N) = (p-1)(q-1)$ elements.

We state without a proof the main lemma of the reduction.

Lemma 2 *With constant probability, a uniformly random element x of G has the property that $ord(x) = 2r$, for some integer $r \geq 1$, and both $\gcd(N, x^r + 1)$ and $\gcd(N, x^r - 1)$ are nontrivial factors of N.*

We now proceed with the order finding subroutine, and rephrase it first as a period finding problem. For comparison, recall that Simon's functions had the property that $f(x) = f(y)$ iff $x \oplus s = y$ for some hidden s to be computed. Similalry, in Shor's case, let $f(r) = x^r \mod N$, which implies $f(r_1) = f(r_2)$ iff $r_1 = r_2 + s$, where s is the function's period and thus it satisfies $x^s = 1$. The above Lemma 2 states that if we knew the order of some element x (of even order) we could reveal some factors of N.

Period finding quantum algorithm

1. Let $Q = 2^q \approx N^2$. First perform our favorite steps in a quantum algorithm (initialization, quantum superposition) which result into the following state

$$\frac{1}{\sqrt{Q}} \sum_{r=1}^{Q-1} |r\rangle \, |x^r \mod N\rangle \, .$$

   Note that computing $x^r \mod N$ can be done efficiently by repeated squaring.

2. Measure the second register and obtain a global state

$$\frac{1}{\sqrt{l}} \sum_{i=0}^{l} |r_0 + i \ s\rangle \, |f(r_0)\rangle \, ,$$

   where $l = \lfloor \frac{Q - r_0 - 1}{s} \rfloor$. If we now made the mistake of measuring the first register we would end up with an irrelevant random state. Instead, Shor performs the following trick:

3. Apply a Quantum Fourier Transform to the input register. A QFT is a unitary transformation that maps a state $|r\rangle$ into state $\frac{1}{\sqrt{Q}} \sum_{i=0}^{Q-1} \omega^{r \ i} \, |i\rangle$, where $\omega^Q = 1$, i.e. $\omega = e^{2\pi j/Q}$. This operation leads to a state

$$\frac{1}{\sqrt{Q}} \frac{1}{\sqrt{l}} \sum_{i=1}^{l} \sum_{r_1=0}^{Q-1} \omega^{(r_0 + is) \ r_1} \, |r_1\rangle \, |f(r_0)\rangle \, .$$

   Fortunately, QFTs can be implemented quantumly by circuits of size $O(\log^2 N)$ using Hadamard gates and controlled phase shift gates, which we will not detail in this lecture.

4. Measure now the first register and observe state $|r_1\rangle \, |f(r_0)\rangle$, with probability (ignoring the normalization factors) essentially

$$\left| \sum_{i=1}^{l} \omega^{(r_0 + is) \ r_1} \right|^2 = \left| \omega^{r_0 r_1} \sum_{i=1}^{l} (\omega^{s r_1})^i \right|^2 .$$

   This brings us to a pleasant state of affairs, since most of the states have very low amplitude and are most probably not being observed. Indeed, analytic considerations show that the quantity $\left| \sum_{i=1}^{l} (\omega^{s r_1})^i \right|^2$ is either very large, when $\omega^{r_1 \ s} \approx 1$, or very small otherwise. The

intuition is that, if the complex vector ω^{sr_1} forms a large angle with the real axis, then summing up over its periodic rotations cancels out the amplitudes, while if that angle is very small the amplitudes add up. In conclusion, if one can observe state $|r_1\rangle\,|f(r_0)\rangle$ it must be the case that $\omega^{r_1 s} \approx 1 = \omega^Q$, which means that one can estimate a multiple of the period s by $\frac{Q}{r_1}$. Sampling a couple of more times and taking the gcd of the multiples obtained reveal the value of s.

3 The Hidden Subgroup Problem

Simon's and Shor's algorithms are prominent illustrations of a general framework, the Hidden Subgroup Problem, where one is given a black box computing a very structured function and wants to determine its 'generalized period'. More formally, let G be a group, H a subgroup of G, and consider the oracle function $f : G \to \Omega$ (Ω could be any set) such that $f(x) = f(y)$ iff $\exists h \in G$ s.t. $x = hy$ for some $h \in H$ (in other words, x and y belong to the same left coset of H). The question is now of finding H (i.e. a set of generators for H) using as few queries as possible to f. Let's now state Simon's problem as a HSP. Indeed, there we had $G = \mathbb{Z}_2^n$ and $H = \{0^n, s\}$ since $f(x) = f(x \oplus s)$. Similarly, in Shor's example $G = \mathbb{Z}_{\phi(N)}$, and $H = \{0, s \mod N, 2s \mod N \ldots\}$ since $f(x) = f(x + i\ s)$. It turns out that computing H can be done efficiently quantumly for more general groups, namely all finite abelian groups.

Theorem 3 (Shor, Kitaev) HSP $\in BQP$ *for any finite abelian group.*

For non-abelian groups the question has been a huge challenge for more than a decade.

A curious student: Is HSP NP-complete?

Scott: We do not know but that would be extremely surprising, since we have a theorem that states that if SAT is reducible to HSP then the Polynomial Hierarchy collapses, which is not believed to be true. Recall that $PH = P \cup NP \cup NP^{NP} \cup NP^{NP^{NP}} \cup \ldots$ and the kth level is defined as $NP^{NP^{\ldots^{NP}}}$ with k NP oracles. For constant k, the kth level of the PH can therefore be described by problems of the form $\exists x_1 \forall x_2 \exists x_3 \ldots \exists x_k\ \phi(x_1, \ldots, x_k)$.

A curious student: If PH collapses can we conclude that $P = PSPACE$?

Scott: We do not know that either. Indeed a complete problem for PSPACE looks like $\exists x_1\ \forall x_2\ \exists x_3 \ldots \exists x_k \quad \phi(x_1, \ldots, x_k)$, but here $k = poly(n)$. We do know however that HSP$\in NP \cap coAM$ and that HSP$\in$ *Statistical Zero Knowledge (SZK).* Also *Approximate Shortest Vector* reduces to HSP over the dihedral group.

Let's prove some nice fact about HSP. We have seen that FACTORING is reducible to HSP over $\mathbb{Z}_{\phi(N)}$. We also can show that

Theorem 4 *GI (*Graph Isomorphism*)* $\leq_T$ *HSP over* S_n *(the Symmetric group on* n *elements).*

Proof: (Sketch) Let C_1 and C_2 be the two graphs given as input. We can assume that each C_i is connected. Let $C = C_1 \cup C_2$ be the disjoint union of the 2 graphs. Label the vertices of C_1 with

distinct integers $1 \ldots n_1$ and label the vertices of C_2 with distinct integers $n_1 + 1, \ldots n_1 + n_2$. Let $\mathcal{G}$ be the set of graphs in $n = n_1 + n_2$ vertices.

Let $G = S_n$ and $H = Aut(C) = \{\pi \in G | \quad \pi(C) \text{ is isomorphic to } C\}$, where $\pi(C)$ is the graph obtained from C by permuting its vertices according to π. Clearly H is a subgroup of G. Define a function $f : G \to \mathcal{G}$ by $f(\pi) = \pi(C)$. Notice that if $\pi = \tau\rho$ where $\rho \in Aut(C)$ then $f(\pi) = (\tau\rho)(C) = \tau(C) = f(\tau)$, and thus f is constant on cosets of H. Suppose that we know how to compute H. The main observation is that if $C_1 \not\cong C_2$ then, since C_1 and C_2 are each connected, the permutations that occur in H are only those that act independently on the C_i's.

□

Coming back to the question of efficiently solving HSP for non-abelian groups we state the following result for which we will sketch a proof in the next lecture.

Theorem 5 (Ettinger, Hoyer, Knill) HSP *over any finite group can be solved with a polynomial number of queries.*

6.845 Quantum Complexity Theory
Fall 2010