6.896 Quantum Complexity Theory Sep. 30, 2008

Lecture 8

Lecturer: Scott Aaronson

1 Hidden Subgroup Problem

Last time we talked about Shor's factoring algorithm without going through all the details. Before we continue, first let us say something about the quantum Fourier transform (QFT) used in Shor's algorithm. The circuit of a n-bit QFT is defined recursively, that is, a $(n-1)$-bit QFT followed by a sequence of controlled phase rotation R_k and a Hadamard, as shown in Figure 1, where

$$R_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{\pi i 2^k/2^n} \end{bmatrix}.$$

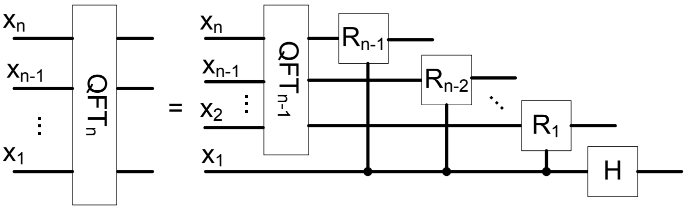


Figure 1: quantum Fourier transform

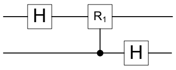
Shor's algorithm actually solved a particular instance of what we call the *Hidden Subgroup Problem (HSP)*. In this problem, we are given a finite group G which has a hidden subgroup H. Our goal is to find H, or equivalently, to find a set of generators for H. In order to do so, we are given oracle access to a function $f : G \to Z$, where f is constant on the cosets of H. (Given an element $g \in G$, a coset of H corresponding to g is the set Hg.) Shor's algorithm solves the HSP in the case where G is a cyclic group, e.g., Z_N. It was known since 1970's that if we can find the period of a periodic function, then we are able to factor integers. While finding such a period is the same thing as solving the HSP over a cyclic group where the hidden subgroup is decided by the period.

Let N be the integer that we want to factor. In Shor's algorithm, we prepare a quantum state $\frac{1}{\sqrt{N}}\sum_r |r\rangle$, query the function $f(r) = x^r \mod N$ in superposition for some x chosen randomly, and get $\frac{1}{\sqrt{N}}\sum_r |r\rangle|x^r \mod N\rangle$. Then we measure the second register (the $|x^r \mod N\rangle$ part), and what is left in the first register is a superposition over all possible values of r which could allow us to get the value we observe in the second register. These r's differ by multiples of the period P of f (the least value P such that $x^P \equiv 1 \mod N$), and thus the superposition left in the first register can be written as $|r\rangle + |r+P\rangle + |r+2P\rangle + \cdots$.

To find out P, we use the quantum Fourier transform, which is the central part of Shor's algorithm. (Notice that given a periodic function, the Fourier transform will map it to its period.) Let n be the number of qubits, and $N = 2^n$ the number of states —that is, the dimension of our system. The N-dimensional QFT is the unitary matrix

$$QFT(N) = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix},$$

where $\omega = e^{2\pi i/N}$. It is easy to check that $QFT(N)$ is indeed a unitary operation, so in principle quantum mechanism allows us to apply this operation. Shor showed that this operation can be implemented (at least approximately) using polynomially many quantum gates —polynomial in n, not in N. The circuit is what we have given in Figure 1, defined recursively, using n^2 quantum gates.

To gain some intuition about QFT, let us see what happens when $n = 2$. First of all, when $n = 1$, we have a QFT on 1 qubit, which is the Hadamard. So the circuit of QFT_2 is 

, where $R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$. The translation procedure for each possible state is as below (unnormalized):

$$\begin{aligned}
&|00\rangle \xrightarrow{H} |00\rangle + |10\rangle \xrightarrow{R_1} |00\rangle + |10\rangle \xrightarrow{H} |00\rangle + |01\rangle + |10\rangle + |11\rangle, \\
&|01\rangle \to |01\rangle + |11\rangle \to |01\rangle + i|11\rangle \to |00\rangle - |01\rangle + i|10\rangle - i|11\rangle, \\
&|10\rangle \to |00\rangle - |10\rangle \to |00\rangle - |10\rangle \to |00\rangle + |01\rangle - |10\rangle - |11\rangle, \\
&|11\rangle \to |01\rangle - |11\rangle \to |01\rangle - i|11\rangle \to |00\rangle - |01\rangle - i|10\rangle + i|11\rangle.
\end{aligned}$$

The corresponding unitary matrix is (after some reordering) $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$, which is what we want, i.e., $QFT(4)$. By induction we can prove that the circuit in Figure 1 gives QFT on n qubits.

2 Ettinger-Hoyer-Knill Theorem

Shor's algorithm is an example of this general hidden subgroup paradigm, which includes (not all, but) a huge number of quantum algorithms we know about today. As we have seen before, Simon's algorithm solves in quantum polynomial time a special case of the HSP where $G = Z_2^n$. If we can solve the HSP for general non-Abelian groups, in particular, if we can solve for the symmetric group S_n, then we can solve in quantum polynomial time the graph isomorphism problem.

We do not know how to solve HSP for arbitrary groups in quantum polynomial time, but we do know the following result given by Ettinger, Hoyer and Knill:

Theorem 1 *The hidden subgroup problem can always be solved with poly(n) queries to f.*

Proof: *(sketch)* To solve the HSP for a given group G, use our favorite procedure: First go into a superposition over all elements in G ($\frac{1}{\sqrt{|G|}}\sum_{x\in G}|x\rangle$), then query f in this superposition and get $\frac{1}{\sqrt{|G|}}\sum_{x\in G}|x\rangle|f(x)\rangle$. Now measure the $|f(x)\rangle$ register, and what's left in the first register is a superposition $|C\rangle$ over a coset of H, i.e., $|C\rangle = \sum_{h\in H}|hy\rangle$ for some $y \in G$. Repeat this procedure K times, and we get a bunch of superpositions over cosets of H with different values of y, denoted as $|C_1\rangle, \cdots, |C_K\rangle$. We claim that if K is large enough, say $\log^2|G|$, which is just polynomial in the number of qubits, then there exists some measurement (no matter polynomial or not) that can tell us the subgroup.

To prove the above claim, first notice that G can have at most $|G|^{\log|G|}$ different subgroups. This is because each subgroup can have at most $\log|G|$ generators (the size of the subgroup doubles after adding each generator).

Now we need a crucial concept: the *inner product* between two quantum states. It is a way to measure how close the two states are to each other. Let $|\psi\rangle = \alpha_1|1\rangle + \cdots + \alpha_N|N\rangle$ and $|\phi\rangle = \beta_1|1\rangle + \cdots + \beta_N|N\rangle$ be two quantum states, the inner product between them is denoted by $\langle\psi|\phi\rangle = \alpha_1^*\beta_1 + \cdots + \alpha_N^*\beta_N$. Notice that if two quantum states are identical, their inner product is 1; and if they are perfectly distinguishable, their inner product is 0, that is, they are orthogonal to each other.

Consider the coset states $|C_1\rangle\cdots|C_K\rangle$ we get when we vary the subgroup H. Let $|\psi_H\rangle = |C_1\rangle\otimes\cdots\otimes|C_K\rangle$, and consider $\langle\psi_H|\psi_{H'}\rangle$ for two subgroups $H \neq H'$. Because there exists an element x such that $x \in H\setminus H'$, and because $\forall y \in H\cap H'$, $yx \in H\setminus H'$, we have that $|H\cap H'| \leq \frac{|H|}{2}$. Therefore $|\langle H|H'\rangle| \leq \frac{1}{\sqrt{2}}$, where $|H\rangle$ and $|H'\rangle$ are two quantum states over all elements in H and H' respectively. (That means if two subgroups are almost the same, then they are actually the same. While if they are different from each other, then they are very different —by a constant factor of all of the places.) Moreover, if $\langle\psi|\phi\rangle \leq \varepsilon$, then $((\langle\psi|\otimes\langle\psi|)(|\phi\rangle\otimes|\phi\rangle) = \langle\psi|\phi\rangle\cdot\langle\psi|\phi\rangle \leq \varepsilon^2$. Therefore $\langle\psi_H|\psi_{H'}\rangle \leq (\frac{1}{\sqrt{2}})^K$, since the inner product of two cosets of H and H' can only get smaller than $\langle H|H'\rangle$. As there are at most $|G|^{\log|G|}$ distinct subgroups $H_1, H_2, \cdots$, there are at most this much $|\psi\rangle$'s, and $\langle\psi_{H_i}|\psi_{H_j}\rangle \leq (\frac{1}{\sqrt{2}})^K$ for any $i \neq j$. Choose K such that $(\frac{1}{\sqrt{2}})^K << |G|^{-2\log|G|}$. Using the Gram-Schmidt process, we can make all $|\psi\rangle$'s exactly orthogonal, while introducing a total error at most $|G|^{2\log|G|}(\frac{1}{\sqrt{2}})^K << 1$. Then there exists some unitary operation U which, when applied to our $|\psi_H\rangle$ will rotate it to a particular state such that the measurement will tell H with exponentially small error probability.

□

Remark. Notice that the above procedure may take exponential time, but when talking about query complexity, we do not care about how much time is needed for computation that does not involve queries to f. This is the distinction between query complexity and computation complexity. Thus it is possible that solving HSP requires exponential computation time. But recall that even assuming computation is free, solving HSP in the classical world may still need exponentially many queries to f, as we have met when discussing Simon's algorithm.

3 Grover's Algorithm

An important question about quantum computation is: can we design polynomial time quantum algorithm to solve NP-complete problems? Along this line we will talk about the other main quantum algorithm that we know, Grover's algorithm.

Given oracle access in superposition to a function $f : \{0,1\}^n \rightarrow \{0,1\}$, the problem is to find some $x \in \{0,1\}^n$ such that $f(x) = 1$, providing that such an x indeed exists. For simplicity, we assume that there is exactly one x for which $f(x) = 1$.

Another way to think about it is to search a database with N items for a "marked item". In classical world, any deterministic algorithm will require N queries to the database in the worst case, and any randomized algorithm will require $N/2$ queries in expectation. If we can query f in superposition, things gets more interesting. Say we can take a superposition over all items, $\sum_x \alpha_x |x\rangle$, make a query to f in this superposition and get $\sum_x \alpha_x (-1)^{f(x)} |x\rangle$, or equivalently, $\sum_x \alpha_x |x\rangle |f(x)\rangle$. Then can we find the marked item using only n^2 ($n = \log N$) queries? That is, what is the quantum query complexity of searching a database? If this can be done polynomially, and further, if the algorithm can be implemented in polynomial time, then quantum computer can solve NP-complete problems in polynomial time, and $NP \subseteq BQP$. However, a straight-forward method is not going to work. That is, if we make the above query to f and measure the second register, then most of the time we will get an x such that $f(x) = 0$. To extract the good solution, we need to explore the structure of f.

Theorem 2 (Grover) *We can search a database of N items in $O(\sqrt{N})$ queries in quantum computation.*

Remark 1. This result is tight, and we will prove this point later. Actually it is proved to be tight before the algorithm was discovered.
Remark 2. Compared with Simon's and Shor's algorithm, Grover's algorithm gives only a quadratic speedup rather than an exponential one. But it works for a much wider range of problems —any combinatorial searching problem.

The algorithm starts as every quantum algorithm: go into a superposition of all possible solutions, $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$, then query f in this superposition and get $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle$.

Now it comes the magical part: Grover Diffusion Operator. Basically what we want to do is to apply some unitary operation that takes all amplitudes and "inverts them about the average". Let the amplitude vector be $[\alpha_1 \cdots \alpha_N]^T$ ($N = 2^n$), and $S = \frac{\alpha_1 + \cdots + \alpha_N}{N}$ the average, we want to get the vector $[\alpha_1 - 2(\alpha_1 - S), \cdots, \alpha_N - 2(\alpha_N - S)]^T$. That is, we want to "flip" every amplitude around the average, as shown in Figure 2.

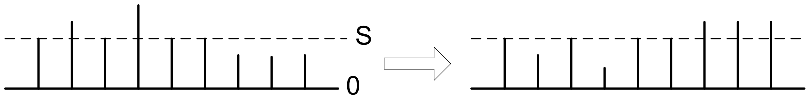


Figure 2: Invert About Average

The corresponding unitary operation is $\begin{bmatrix} \frac{2}{N} - 1 & \frac{2}{N} & & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \vdots & \frac{2}{N} & & \ddots & \frac{2}{N} \\ \frac{2}{N} & & \cdots & \frac{2}{N} & \frac{2}{N} - 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}$, where the elements in the matrix's diagonal are all $\frac{2}{N} - 1$, and the other elements are all $\frac{2}{N}$. It is easy to verify that this operation is indeed unitary.

The circuit of Grover's algorithm is shown in Figure 3, where f stands for a query to the oracle f, and D stands for a Grover diffusion operator. The basic f, D operation is repeated $\sqrt{N}$ times, and then measure.

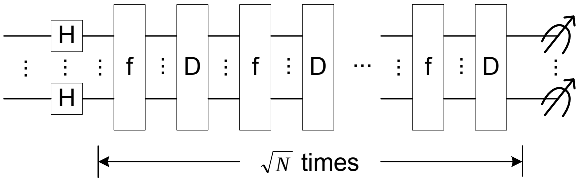


Figure 3: Grover's Algorithm

The circuit for the diffusion operation is shown in Figure 4, where U_0 is the unitary operation that maps $|x\rangle$ to $(-1)^x|x\rangle$, where $x = 0, 1$. Essentially, in the diffusion operation we first switch from the standard basis to the Fourier basis. Then in the Fourier basis, we negate all the Fourier coefficients except for the first one, which corresponds to the average. Finally we return to the standard basis.

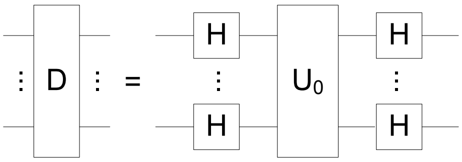


Figure 4: Grover Diffusion Operator

We will analyze this algorithm next time.

6.845 Quantum Complexity Theory
Fall 2010