CS 880: Quantum Information Processing 10/7/2010

Lecture 15: The Hidden Subgroup Problem

Instructor: Dieter van Melkebeek Scribe: Hesam Dashti

The Hidden Subgroup Problem is a particular type of symmetry finding problem. It captures a lot of quantum algorithms and instantiations of this problem give many of the previous problems that we have seen in this course. In this lecture we see how we can cast the previous problems as instantiations of this problem. We also consider an efficient Quantum algorithm to solve the Hidden Subgroup Problem on finite Abelian groups.

1 The Hidden Subgroup Problem

Let us start with the definition of the Hidden Subgroup Problem (HSP):

Definition 1. *Given access to an oracle function $f : G \to R$, from a known group G to its range, such that there exists a subgroup $H \leq G$ such that $\forall x, y \in G, f(x) = f(y) \Leftrightarrow Hx = Hy$.* ***Problem*** *is to find a set S of generators of H.*

This means that the function f, returns the same value iff x and y belong to the same coset of H within G. Note that

$$Hx = Hy \Leftrightarrow yx^{-1} \in H$$

and in general we can have right or left coset (in this case right coset). Clearly, when we consider an Abelian group G or the subset H is a normal subgroup of G, there is not any difference between right and left cosets of H, where $Hx = \{hx | h \in H\}$.
Before delving into efficient algorithms for finding a set S of generators for H, a more basic question is whether a small set of generators exists. The answer is "yes" and its proof comes as an exercise.

Exercise 1. *Show that for every subgroup H of G, there exists a set of generators of size at most $\log_2 |H|$.*

We will use $\langle S \rangle$ to denote the set of elements generated by S.
Next, we show that how HSP captures many of the problems we have seen before.

1.1 Deutsch Problem

The Deutsch problem is a simple instantiation of the HSP problem. Let us recall the problem:

$$\text{For a given } f : \{0,1\} \to \{0,1\} \text{ the problem is whether } f(0) = f(1) \text{ or not.}$$

In order to cast this as an HSP instantiation, we define the group $G = \mathbb{Z}_2$ and we also need to define our oracle function, which is the same as the function f in the Deutsch problem. When the function f is constant, the set H equals G and otherwise it is $\{0\}$. Hence, the HSP problem would be distinguishing between

$$H = \{0\} \text{ vs } H = \mathbb{Z}_2, \text{ where, } G = \mathbb{Z}_2.$$

1.2 Bernstein-Vazirani Problem

Let us recall the problem; for a given

$$f : \{0,1\}^n \to \{0,1\} : x \to ax + b$$

and the challenging goal was to find a.
First let define our group $G = \mathbb{Z}_2^n$, and the oracle function is as the same as the f function. Now we use the definition of the subset H to cast it, which is looking for $f(x) = f(y)$. Hence, in the Bernstein-Vazirani Problem, for all the elements of H we should have $a(x - y) = 0$:

$$H := \{z \in \mathbb{Z}_2^n | az = 0\}.$$

Once we have a set of generators for H, we can find a as the nontrivial solution to a homogeneous system of linear equations.

1.3 Simon's Problem

In Simon's Problem we were given a function $f : \mathbb{Z}_2^n \to R$ such that $f(x) = f(y) \Leftrightarrow x + y = s$. The function is either one-to-one ($s = 0$) or two-to-one ($s \neq 0$) and the **goal** was to finding the shift s. In order to cast it, the group $G = \mathbb{Z}_2^n$, the oracle function is this f function, and the H is the subgroup generated by $S = \{s\}$.

1.4 Period Finding

For a given function $f : \mathbb{Z} \to \mathbb{R}$, we knew that f is periodic with period of r,

$$(\forall x) f(x) = f(x + r), \text{ and } (\forall\ 0 \leq x < y < r) f(x) \neq f(y)$$

and the **goal** was to find the period r. Hence, the group $G = \mathbb{Z}$, the oracle is the same as f function, and H is a subgroup generated by r.

We note that the *Order Finding Problem* is a special case of the Period Finding Problem and falls in this category of HSP problems, as well.

1.5 Finding Discrete Logarithms

Let us first setup the problem by introducing notations; an integer $M > 0$, g generator for $\mathbb{Z}_M^*$, $a \in \mathbb{Z}_M^*$, $R = |\mathbb{Z}_M^*|$, used to find smallest l, such that $g^l = a$.
The function we used in the previous lecture was $f : \mathbb{Z}_R \times \mathbb{Z}_R \to \mathbb{Z}_M : (x, y) \to a^x g^y \mod M$. This function was designed such that it is constant on the coset of $H = \langle (1, -l) \rangle$ in $G = \mathbb{Z}_R \times \mathbb{Z}_R$. Indeed,

$$f(x_1, y_1) = f(x_2, y_2) \Leftrightarrow lx_1 + y_1 = lx_2 + y_2 \Rightarrow l(x_1 - x_2) = -(y_1 - y_2).$$

We use the function f as a HSP instantiation and can extract l as the generator of H.

1.6 Graph Automorphism and Graph Isomorphism

For a graph $A(V, E)$, say with $|V| = n$, a graph automorphism is a relabeling of vertices which preserves the edges. The graph automorphism problem for A is to find a set of generators for the group of automorphisms $Aut(A)$ of A. We can cast the problem as an HSP over the group G of all permutations of $\{1, 2, \ldots, n\}$, i.e., G is the symmetric group S_n. In addition, our function f should be constant on the automorphisms of G and the way that we can define it, is that for a permutation π, $f(\pi) = \pi(A)$.

$$\begin{aligned} f(\pi) = f(\sigma) &\Leftrightarrow \pi(A) = \sigma(A) \\ &\Leftrightarrow (\sigma^{-1}\pi)(A) = A \\ &\Leftrightarrow \sigma^{-1}\pi \in Aut(A) \\ &\Leftrightarrow \sigma Aut(A) = \pi Aut(A) \end{aligned}$$

We note that, in this case the group G is not an Abelian group and the subgroup $H = Aut(A)$ is not always normal. Based on the above definition our subgroup is a left coset.
Hence, the Graph Automorphism Problem can be cast as an instantiation of HSP problem. Next we consider the Graph Isomorphism Problem.

The Graph Isomorphism Problem reduces to The Graph Automorphism Problem
Here, we claim that if there is an efficient algorithm to solve the Graph Automorphism Problem we can use it to solve the Graph Isomorphism Problem.

Two connected graphs A_1 and A_2 are isomorphic iff there exists an automorphisms of $A_1 \dot{\cup} A_2$ that maps a vertex from A_1 to a vertex from A_2. Given a set of generators for $Aut(A_1 \dot{\cup} A_2)$, we can check the latter condition by verifying that there is a generator that maps a vertex from A_1 to a vertex from A_2.
In the general case, in which A_1 and A_2 are not connected graphs, we can add a node to each graphs, connect the new nodes to every other vertex in their associated graphs, and make a connected graph. Actually, the new nodes have maximum degree in each graph. Next, by adding an extra node to each graph which is only connected to the new node, we can handle the case that the graphs are full-connected. Then, if we can find a permutation which maps one vertex from one graph to a vertex from another graph and also preserves edges, the graphs are isomorphic.

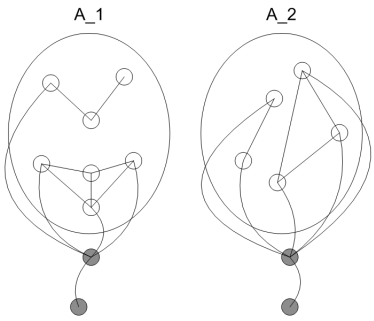

Figure 1: The extra nodes are shown with filled circles and new edges are shown by curves. We only showed some of the new edges to show general idea of using extra vertexes.

2 Efficient Quantum Algorithm for The Hidden Subgroup Problem Over Finite Abelian Groups

Each finite Abelian group is isomorphic to the direct sum of some cyclic groups

$$G = \bigoplus_{j=1}^{k} \mathbb{Z}_{N_j}$$

In lecture 8 we developed the Fourier Transform over such G, which was a mapping from the standard δ basis into orthonormal basis that maps convolutions into the point wise products. We use the Fourier Transform here, because the Fourier Transform of a group interacts very nicely with the symmetries in the group. In particular, it perfectly works for a coset state $|Hg\rangle$, which is the uniform superposition of all elements of the coset H,

$$|Hg\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |hg\rangle$$

Now, let us complete the Fourier Transform of a coset state

$$F\,|Hg\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} \frac{1}{\sqrt{|G|}} \sum_{y \in G} \chi_y(hg)\,|y\rangle \tag{1}$$

$$= \frac{1}{\sqrt{|H||G|}} \sum_{y \in G} \chi_y(g)(\sum_{h \in H} \chi_y(h))\,|y\rangle\,, \tag{2}$$

where,

$$\chi_y(h) = e^{2\pi i \sum_{j=1}^{k} \frac{y_j h_j}{N_j}}.$$

Exercise 2. *Show that*

$$\sum_{h \in H} \chi_y(h) = \begin{cases} |H| & \text{if } y \in H^{\perp} \\ = 0 & \text{otherwise} \end{cases}$$

where

$$H^{\perp} = \{y \in G | (\forall h \in H) \sum_{j=1}^{k} \frac{y_j h_j}{N_j} \in \mathbb{Z}\}$$

Plugging in Exercise 2 into Equation (2) gives us:

$$F\,|Hg\rangle = \sqrt{\frac{|H|}{|G|}} \sum_{y \in H^{\perp}} \chi_y(g)\,|y\rangle\,. \tag{3}$$

This is the reason of using the Fourier Transform; we started with the coset state H_g of all elements and by using the Fourier Transform we get an equally weighted superposition over $H^{\perp}$. In particular, if $g = 0$ we get the coset state which is a uniform superposition over $H^{\perp}$.
The quantum algorithm for solving the HSP over G starts with the uniform superposition over G:

$$|G\rangle = \frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle$$

By applying our blackbox fo f we obtain

$$\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle \, |f(x)\rangle \, .$$

Next, we observe the second register $|f(x)\rangle$, which leaves us in the first register the coset state $|Hg\rangle$ for a uniformly at random chosen $g \in G$.
The next step is applying F^{-1}, which by Equation (3) gives us:

$$\sqrt{\frac{|H|}{|G|}} \sum_{y \in H^\perp} \overline{\chi_y(g)} \, |y\rangle \, .$$

Then, we observe the first register to get a uniform $y \in H^\perp$.

Lemma 1. *When we run these steps* $\log |H^\perp|$ *times and collect the y's, then*

$$\Pr[\langle y_1, y_2, \ldots, y_{\log |H^\perp|} \rangle = H^\perp] \geq \delta^*,$$

where δ^* *is a universal positive constant.*

The proof is the same as what we explained for Simon's problem in Lecture 5.
When we find a set of generators for $H^\perp$, we can use it to find $(H^\perp)^\perp$ to get the hidden subgroup H by solving a linear system of modular equations. This way we solve the HSP over a finite Abelian group G in poly-logarithmic time in $|G|$.

To solve the problem over finitely generated Abelian groups that are not finite, we perform a similar process as we did for the Period Finding problem.

3 Hidden Subgroup Problem for Non-Abelian Groups

In general, the HSP over any finite group G can be solved using only polylog($|G|$) many queries to the blackbox f. This is something we will prove in the next homework. However, this does not mean that the overall algorithm runs in polylogarithmic time in $|G|$. In fact, we only know how to do the latter for a few non-Abelian finite groups. We do not know it for the following groups, for which efficient solutions to the HSP would have interesting consequences.

1. **Dihedral Groups (D_N)**: The group D_N consiste of the symmetries of a regular N-gon (rotations and reflections). A solution to the HSP problem on the Dihedral groups that would allow us to solve the $g(N)$-unique SVP (Shortest Vector Problem) for polynomial functions $g(N)$. The SVP is a Lattice Problem on a real N-dimensional space with N basis vectors, where every element in the lattice is a linear combination of the basis vectors with integer coefficients. The SVP asks for a shortest nonzero vector in the lattice. The SVP does not have a unique solution because for every vector in the lattice its minus is also in the lattice. The unique SVP is a promise version of SVP in which we are told that the solution is unique up to the sign. The term "$g(N)$-unique" means that the second shortest lattice vector up to sign is of length at least $g(N)$ times the length of the shortest nonzero lattice vector. Solving the $g(N)$-unique SVP for polynomial $g(N)$ is considered hard in the classical setting, and is used to design lattice-based cryptosystems. Efficiently solving the HSP over the dihedral group would break those cryptosystems.

2. **Symmetric Group**: As we saw, Graph Isomorphism reduces to the HSP over the symmetric group, and the coset states $|Hg\rangle$ contain enough information to solve the problem in principle. However, it turns out that Fourier sampling loses that information. More specifically, there are positive and negative instances of Graph Isomorphism for which the dsitributions that result after Fourier sampling are exponentially close. This means we would need exponentially many runs in order to have a good chance of distinguishing the positive from the negative instances.

Now, we are done with the Hidden Subgroup Problem and in the next lecture, we will talk about "Quantum Walks".