Quantum algorithms (CO 781, Winter 2008)
Prof. Andrew Childs, University of Waterloo

LECTURE 6: Quantum query complexity of the HSP

So far, we have considered the hidden subgroup problem in abelian groups. We now turn to the case where the group might be nonabelian. We will look at some of the potential applications of the HSP, and then show that the general problem has polynomial quantum query complexity.

The nonabelian HSP and its applications Recall that in the hidden subgroup problem for a group G, we are given a black box function $f : G \to S$, where S is a finite set. We say that f *hides* a subgroup $H \leq G$ provided

$$f(x) = f(y) \text{ if and only if } x^{-1}y \in H. \tag{1}$$

In other words, f is constant on left cosets $H, g_1H, g_2H, \ldots$ of H in G, and distinct on different left cosets. When G is a nonabelian group, we refer to this problem as the nonabelian HSP.

The nonabelian HSP is of interest not only because it generalizes the abelian case in a natural way, but because a solution of certain nonabelian hidden subgroup problems would have particularly useful applications. The most well-known (and also the most straightforward) applications are to the *graph automorphism problem* and the *graph isomorphism problem*, problems for which no efficient classical algorithm is currently known.

In the graph automorphism problem, we are given a graph Γ on n vertices, and the goal is to determine whether it has some nontrivial automorphism. In other words, we would like to know whether there is any nontrival permutation $\pi \in S_n$ such that $\pi(\Gamma) = \Gamma$. The automorphisms of Γ form a subgroup $\operatorname{Aut}\Gamma \leq S_n$; if $\operatorname{Aut}\Gamma$ is trivial then we say Γ is *rigid*. We may cast the graph automorphism problem as an HSP over S_n by considering the function $f(\pi) := \pi(\Gamma)$, which hides $\operatorname{Aut}\Gamma$. If we could solve the HSP in S_n, then by checking whether or not the automorphism group is trivial, we could decide graph automorphism.

In the graph isomorphism problem, we are given two graphs Γ, Γ', each on n vertices, and our goal is to determine whether there is any permutation $\pi \in S_n$ such that $\pi(\Gamma) = \Gamma'$, in which case we say that Γ and Γ' are isomorphic. We can cast graph isomorphism as an HSP in the wreath product $S_n \wr S_2 \leq S_{2n}$, the subgroup of S_{2n} generated by permutations of the first n points, permutations of the second n points, and swapping the two sets of points. Writing elements of $S_n \wr S_2$ in the form (σ, τ, b) where $\sigma, \tau \in S_n$ represent permutations of Γ, Γ', respectively, and $b \in \{0, 1\}$ denotes whether to swap the two graphs, we can define a function

$$f(\sigma, \tau, b) := \begin{cases} (\sigma(\Gamma), \tau(\Gamma')) & b = 0 \\ (\sigma(\Gamma'), \tau(\Gamma)) & b = 1. \end{cases} \tag{2}$$

This function hides the automorphism group of the disjoint union of Γ and Γ', which contains an element that swaps the two graphs if and only if they are isomorphic. In particular, if Γ and Γ' are rigid (which seems to be the hardest case for the HSP approach to graph isomorphism), the hidden subgroup is trivial when Γ, Γ' are non-isomorphic; and has order two, with its nontrival element the involution $(\pi, \pi^{-1}, 1)$, when $\Gamma = \pi(\Gamma')$.

The second major potential application of the hidden subgroup problem is to lattice problems. An *n-dimensional lattice* is the set of all integer linear combinations of n linearly independent

vectors in $\mathbb{R}^n$ (a *basis* for the lattice). In the *shortest vector problem*, we are asked to find a shortest nonzero vector in the lattice. In particular, in the *g(n)-unique shortest vector problem*, we are promised that the shortest nonzero vector is unique (up to its sign), and is shorter than any other non-parallel vector by a factor $g(n)$. This problem can be solved in polynomial time on a classical computer if $g(n)$ is sufficiently large (say, if it is exponentially large), and is NP-hard if $g(n) = O(1)$. Less is known about intermediate cases, but the problem is suspected to be classically hard even for $g(n) = \text{poly}(n)$, to the extent that cryptosystems have been designed based on this assumption.

Regev showed that an efficient quantum algorithm for the dihedral hidden subgroup problem based on the so-called *standard method* (described below) could be used to solve the poly(n)-unique shortest vector problem. Such an algorithm would be significant since it would break lattice cryptosystems, which are some of the few proposed cryptosystems that are not compromised by Shor's algorithm.

So far, only the symmetric and dihedral hidden subgroup problems are known to have significant applications. Nevertheless, there has been considerable interest in understanding the complexity of the HSP for general groups. There are at least three reasons for this. First, the problem is simply of fundamental interest: it appears to be a natural setting for exploring the extent of the advantage of quantum computers over classical ones. Second, techniques developed for other HSPs may eventually find application to the symmetric or dihedral groups. Finally, exploring the limitations of quantum computers for HSPs may suggest cryptosystems that could be robust even to quantum attacks.

The standard method Nearly all known algorithms for the nonabelian hidden subgroup problem use the black box for f in essentially the same way as in the abelian HSP. This approach has therefore come to be known as the *standard method*.

In the standard method, we begin by preparing a uniform superposition over group elements:

$$|G\rangle := \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle. \tag{3}$$

We then compute the value $f(g)$ in an ancilla register, giving the state

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g, f(g)\rangle. \tag{4}$$

Finally, we measure the second register and discard the result (or equivalently, simply discard the second register). If we obtain the outcome $s \in S$, then the state is projected onto the uniform superposition of those $g \in G$ such that $f(g) = s$, which by the definition of f is simply some left coset of H. Since every coset contains the same number of elements, each left coset occurs with equal probability. Thus this procedure produces the *coset state*

$$|gH\rangle := \frac{1}{\sqrt{|H|}} \sum_{h \in H} |gh\rangle \text{ with } g \in G \text{ uniformly random} \tag{5}$$

(or, equivalently, we can view g as being chosen uniformly at random from some left transversal of H in G).

Depending on context, it may be more convenient to view the outcome either as a random pure state, or equivalently, as the mixed quantum state

$$\rho_H := \frac{1}{|G|} \sum_{g \in G} |gH\rangle\langle gH| \tag{6}$$

which we refer to as a *hidden subgroup state*. In the standard approach to the hidden subgroup problem, we attempt to determine H using samples of this hidden subgroup state. In other words, given $\rho_H^{\otimes k}$ for some $k = \mathrm{poly}(\log |G|)$, we try to find a generating set for H.

Query complexity of the HSP As a first step toward understanding the quantum complexity of the HSP, we can ask how many queries of the hiding function are required to solve the problem. If we could show that an exponential number of quantum queries were required, then we would know that there was no efficient quantum algorithm. But it turns out that this is not the case: as shown by Ettiner, Høyer, and Knill, $\mathrm{poly}(\log |G|)$ queries to f suffice to determine H. In particular, they showed this within the framework of the standard method: $\rho_H^{\otimes \mathrm{poly}(\log |G|)}$ contains enough information to recover H. Of course, this does not necessarily mean that the quantum *computational complexity* of the HSP is polynomial, since it is not clear in general how to perform the quantum post-processing of the hidden subgroup states efficiently. Nevertheless, this is an important observation since it already shows a difference between quantum and classical computation, and offers some clues as to how we might design efficient quantum algorithms.

To show that the query complexity of the HSP is polynomial, it is sufficient to show that the (single-copy) hidden subgroup states are pairwise statistically distinguishable, as measured by the quantum fidelity

$$F(\rho, \rho') := \mathrm{tr}\, |\sqrt{\rho}\sqrt{\rho'}|. \tag{7}$$

This follows from a result of Barnum and Knill, who showed the following.

Theorem. *Suppose ρ is drawn from an ensemble $\{\rho_1, \ldots, \rho_N\}$, where each ρ_i occurs with some fixed prior probability p_i. Then there exists a quantum measurement (namely, the so-called* pretty good measurement*) that identifies ρ with probability at least*

$$1 - N \sqrt{\max_{i \neq j} F(\rho_i, \rho_j)}. \tag{8}$$

In fact, by the minimax theorem, this holds even without assuming a prior distribution for the ensemble.

Given only one copy of the hidden subgroup state, (8) will typically give only a trivial bound. However, by taking multiple copies of the hidden subgroup states, we can ensure that the overall states are nearly orthogonal, and hence distinguishable. In particular, using k copies of ρ, we see that there is a measurement for identifying ρ with probability at least

$$1 - N \sqrt{\max_{i \neq j} F(\rho_i^{\otimes k}, \rho_j^{\otimes k})} = 1 - N \sqrt{\max_{i \neq j} F(\rho_i, \rho_j)^k} \tag{9}$$

(since the fidelity is multiplicative under tensor products). Setting this expression equal to $1 - \epsilon$ and solving for k, we see that arbitrarily small error probability ϵ can be achieved provided we use

$$k \geq \left\lceil \frac{2(\log N - \log \epsilon)}{\log \left(1 / \max_{i \neq j} F(\rho_i, \rho_j)\right)} \right\rceil \tag{10}$$

copies of ρ.

Provided that G does not have too many subgroups, and that the fidelity between two distinct hidden subgroup states is not too close to 1, this shows that polynomially many copies of ρ_H suffice to solve the HSP. The total number of subgroups of G is $2^{O(\log^2 |G|)}$, which can be seen as follows. Any group K can be specified in terms of at most $\log_2 |K|$ generators, since every additional (non-redundant) generator increases the size of the group by at least a factor of 2. Since every subgroup of G can be specified by a subset of at most $\log_2 |G|$ elements of G, the number of subgroups of G is upper bounded by $|G|^{\log_2 |G|} = 2^{(\log_2 |G|)^2}$. This shows that we can take $\log N = \mathrm{poly}(\log |G|)$ in (10). Thus $k = \mathrm{poly}(\log |G|)$ copies of ρ_H suffice to identify H with constant probability provided the maximum fidelity is bounded away from 1 by at least $1/\mathrm{poly}(\log |G|)$.

To upper bound the fidelity between two states ρ, ρ', consider the two-outcome measurement that projects onto the support of ρ or its orthogonal complement. The classical fidelity of the resulting distribution is an upper bound on the quantum fidelity, so

$$F(\rho, \rho') \leq \sqrt{\operatorname{tr} \Pi_\rho \rho \operatorname{tr} \Pi_\rho \rho'} + \sqrt{\operatorname{tr}((1 - \Pi_\rho)\rho) \operatorname{tr}((1 - \Pi_\rho)\rho')} \tag{11}$$

$$= \sqrt{\operatorname{tr} \Pi_\rho \rho'}. \tag{12}$$

where Π_ρ denotes the projector onto the support of ρ.

Now consider the fidelity between ρ_H and $\rho_{H'}$ for two distinct subgroups $H, H' \leq G$. Let $|H| \geq |H'|$ without loss of generality. We can write (6) as

$$\rho_H = \frac{1}{|G|} \sum_{g \in G} |gH\rangle\langle gH| = \frac{|H|}{|G|} \sum_{g \in T_H} |gH\rangle\langle gH|. \tag{13}$$

where T_H denotes some left transversal of H in G. Since the right hand expression is a spectral decomposition of ρ_H, we have

$$\Pi_{\rho_H} = \sum_{g \in T_H} |gH\rangle\langle gH| = \frac{1}{|H|} \sum_{g \in G} |gH\rangle\langle gH|. \tag{14}$$

Then we have

$$F(\rho_H, \rho_{H'})^2 \leq \operatorname{tr} \Pi_{\rho_H} \rho_{H'} \tag{15}$$

$$= \frac{1}{|H| \cdot |G|} \sum_{g, g' \in G} |\langle gH | g'H' \rangle|^2 \tag{16}$$

$$= \frac{1}{|H| \cdot |G|} \sum_{g, g' \in G} \frac{|gH \cap g'H'|^2}{|H| \cdot |H'|} \tag{17}$$

$$= \frac{1}{|G| \cdot |H|^2 \cdot |H|'} \sum_{g, g' \in G} |gH \cap g'H'|^2. \tag{18}$$

Now

$$|gH \cap g'H'| = |\{(h, h') \in H \times H' : gh = g'h'\}| \tag{19}$$

$$= |\{(h, h') \in H \times H' : hh' = g^{-1}g'\}| \tag{20}$$

$$= \begin{cases} |H \cap H'| & \text{if } g^{-1}g' \in HH' \\ 0 & \text{if } g^{-1}g' \notin HH', \end{cases} \tag{21}$$

so

$$\sum_{g,g' \in G} |gH \cap gH'|^2 = |G| \cdot |HH'| \cdot |H \cap H'|^2 \tag{22}$$

$$= |G| \cdot |H| \cdot |H'| \cdot |H \cap H'|. \tag{23}$$

Thus we have

$$F(\rho_H, \rho_{H'})^2 = \frac{|G| \cdot |H| \cdot |H'| \cdot |H \cap H'|}{|G| \cdot |H|^2 \cdot |H'|} \tag{24}$$

$$= \frac{|H \cap H'|}{|H|} \tag{25}$$

$$\leq \frac{1}{2}. \tag{26}$$

This shows that $F(\rho_H, \rho_{H'}) \leq 1/\sqrt{2}$, thereby establishing that the query complexity of the HSP is $\mathrm{poly}(\log |G|)$.