CS 593/MA 592 - Intro to Quantum Computing
Spring 2024
Thursday, March 28 - Lecture 11.2

Today's scribe: Ethan Dickey [Note: not proofread by Eric]

Reading: None.

Agenda:

1. Hidden Subgroup problem
2. Graph isomorphism (and reduction to HSP)
3. HSP for $\mathbb{Z}/n\mathbb{Z}$

1 Hidden Subgroup Problem

Remark. *Since I (Eric) am looking for new applications of quantum computing, I take the perspective that QFT is, in some sense, Pontragian Duality.*

1.1 Definitions

Definition 1 (Hidden Subgroup)**.** *Given a (finite[1]) group G that we know "explicitly," a set X that we know "explicitly," and oracle access to a function $f : G \to X$ which is promised to satisfy the following:*

$$\exists \text{ subgroup } H \subseteq G \text{ such that } f(g_1) = f(g_2) \iff g_1H = g_2H$$

We say that f "hides H."

The intuition behind **??** is that f is an H-periodic function on G valued at X.

Definition 2 (Hidden Subgroup Problem)**.** *Given a hidden subgroup as in **??**, determine H explicitly (i.e. find elements $g_1, \ldots, g_l \in G$ such that $\langle g_1, \ldots, g_l \rangle = H$).*

Assuming $|G| < \inf$, we can express the elements of G using bitstrings of length $L = \mathrm{O}(log|G|)$. Equipped with these bitstrings, "to know G explicitly" means we have access to functions U_m and i_n that encode the group multiplication and inversion:

$$\begin{aligned} U_m &: GxG \to G \\ &(g_1, g_2) \mapsto g_1 g_2 \\ i_n &: G \to G \\ &g \mapsto g^{-1} \end{aligned}$$

Ideally, we can find generators of H in time O(polylog$|G|$)[2]. Time O(poly$|G|$) is not interesting because one can simply iterate through all of the group elements and test for collisions (because $H = g_1^{-1} g_2 H$ implies $g_1^{-1} g_2$ is a group member).

In particular, this means we can only call the oracle for G O(polylog$|G|$) times.

To formulate quantum oracle access, we will assume G, f, X are encoded as follows:

[1] For this course, we will assume G (a group) is finite.
[2] Polylog definition

1. Elements of G are encoded as bitstrings:

$$G \subseteq \{0,1\}^L \implies \mathbb{C}G \leq (\mathbb{C}^2)^{\otimes L}$$

Where $\leq$ is used as subspace notation. We also assume we have a unity:

$$E : (\mathbb{C}^2)^{\otimes L} \to (\mathbb{C}^2)^{\otimes L}$$
$$E|0\dots0\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle$$

This E generalizes $H^{\otimes L}$ when $N = 2^L$.

2. Similarly, we will assume $X \subseteq \{0,1\}^M$, M = O(poly(L)).

3. Quantum oracle access to f will mean we have a unitary $U_f :\ (\mathbb{C}^2)^{\otimes L} \otimes (\mathbb{C}^2)^{\otimes M} \to (\mathbb{C}^2)^{\otimes L} \otimes (\mathbb{C}^2)^{\otimes M}$ such that $U_f|g, 0_1 \dots 0_m\rangle = |g, f(g)\rangle$ (note that we do not care what U_f does to other computational basis vectors).

   Note that while we don't need it, having "explicit" quantum oracle access to G means we also have
   $U_m : (\mathbb{C}^2)^{\otimes L} \otimes (\mathbb{C}^2)^{\otimes L} \otimes (\mathbb{C}^2)^{\otimes L} \to (\mathbb{C}^2)^{\otimes L} \otimes (\mathbb{C}^2)^{\otimes L} \otimes (\mathbb{C}^2)^{\otimes L}$ with
   $U_m|g_1, g_2, 0\dots0\rangle = |g_1, g_2, g_1 g_2\rangle$

1.2 What is known about HSP?

HSP provides a framework that captures nearly all examples of exponential quantum advantage decision problems. In short, there are basically no efficient classical algorithms for HSP on *any* infinite families of groups that we know of.

Oracle problems:

- Factoring, order finding, Deutsch-Jozsa, Simon, Bernstein-Vazirani, period finding, etc. all reduce to HSP for *abelian* groups.
- Ettenger-Høyer-Knill (2004)[3]: for *any* HSP (on a finite group), we can solve it using O(polylog$|G|$) quantum oracle queries. (Classically, in general, proved that we need at least $\theta(|G|)$ queries, even for abelian groups.) However, we need to perform an exponential amount of quantum postprocessing (unfortunately).
- If $H \triangleleft G$ (and $|G| <$ inf), then Hallgren-Russell-(Ta-Shma) (2000)[4] showed that HSP can be solved efficiently quantumly. This is strongly related to the "Fourier Sampling" problem.
- There are many examples of groups that are "close" to abelian groups that admit efficient quantum solutions.

Some important problems (other than factoring) reduce to non-abelian HSP:

- Certain flavors of the "shortest vector problem" (SVP) reduce to HSP for $G = D_N$ a dihedral group (symmetries of a regular N-gon). (There are lattice-based cryptography algorithms that depend on SVP.)
  - There is no known efficient algorithm for dihedral HSP, but there is a subexponential time quantum algorithm due to Kuperberg (2005)[5] and Regev (2004)[6]
- Graph isomorphism reduces to HSP for $G = S_n$, the symmetric group.

[3] The quantum query complexity of the hidden subgroup problem is polynomial
[4] Normal subgroup reconstruction and quantum computation using group representations
[5] A subexponential-time quantum algorithm for the dihedral hidden subgroup problem
[6] A Subexponential Time Algorithm for the Dihedral Hidden Subgroup Problem with Polynomial Space

2 Graph isomorphism

(Aside: In the past 10 years, it has been shown to be in quasi-polynomial time of O(n^{logn}) ish[7].)

Definition 3 (Graph Isomorphism Problem)**.** *Input:* $\Gamma_1 = (V_1, E_1)$, $\Gamma_2 = (V_2, E_2)$, *both are assumed to be connected. Output: YES if* $\Gamma_1 \simeq \Gamma_2$, *NO otherwise.*

We will convert each instance of this problem to an instance of HSP with $G = S_n$ with $n = |V_1| + |V_2|$.

- To do this, we will build X and $f : G \to X$ that hides a subgroup that "knows" whether or not $\Gamma_1 \simeq \Gamma_2$.

Proof. Assume $V_1 = \{1, \dots, n\}$ and $V_2 = \{n+1, \dots, 2n\}$. Consider $G = S_{2n}$. We can identify (abstractly) the automorphism: $Aut(\Gamma_1 \cup \Gamma_2) \subseteq S_{2n}$. We will build an f that hides this.

This construction is sufficient for 2 reasons:

1. There is an automorphism of their union that swaps the 2:

$$\Gamma_1 \simeq \Gamma_2 \iff \exists \alpha \in Aut(\Gamma_1 \cup \Gamma_2) \text{ s.t. } \alpha(\Gamma_1) = \Gamma_2$$

2. $\exists$ such an $\alpha \iff \forall$ generating sets $g_1, \dots, g_l$ of $Aut(\Gamma_1 \cup \Gamma_2)$, some g_i swaps Γ_1 and Γ_2.

Let $X =$ all graphs Γ with $V(\Gamma) = \{1, 2, \dots, 2n\}$ and $\Gamma \simeq \Gamma_1 \cup \Gamma_2$.
Define $F : S_{2n} \to X$, $\sigma \mapsto \sigma * (\Gamma_1 \cup \Gamma_2)$
Here we note that the size of X doesn't really matter because we can write down an element of X efficiently.

Proposition 1. *F hides* $Aut(\Gamma_1 \cup \Gamma_2)$. *That is,* $F(\alpha) = F(\tau) \iff Aut(\Gamma_1 \cup \Gamma_2) = \tau Aut(\Gamma_1 \cup \Gamma_2)$

We do not prove **??** in this proof. □

3 HSP for A = $\mathbb{Z}/n\mathbb{Z}$

We start with the following definitions: N an integer in binary, elements of $A = \mathbb{Z}/n\mathbb{Z}$ are represented by $0, 1, \dots, N-1$ (in binary).

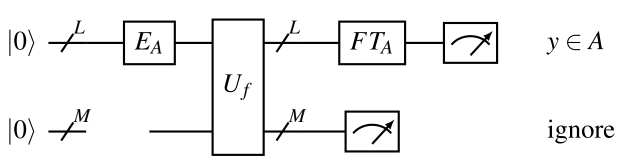
We are given an oracle $f : \mathbb{Z}/n\mathbb{Z} \to X$ that satisfies: $f(a) = f(b) \iff a + H = b + H$ where $H \subseteq \mathbb{Z}/n\mathbb{Z}$.

Since A is cyclic, so are all of its subgroups (including H). Thus, $\exists h \in A$ s.t. $H = \langle h \rangle$, and we want to find this h. We take a step back to define some notation. Let $\omega = \exp 2\pi i/N$, then we can consider FT_A as FT_A: $A \to \hat{A} = Hom(A, V(1))$ (dual of A) for $a \mapsto \rho_a$ where $\rho_a = A \to V(1)$ for $b \mapsto \omega^{ab}$.

Next, we define $H^\perp = \{\rho \in A \,|\, \rho(h) = 1 \,\forall\, h \in H\}$. In words, $H^\perp$ is the set (group) of irreducible representations of A that are trivial whne restricted to H. What follows is a generalization of 1b on Homework 7.

Lemma 2. *$H^\perp$ determines H. In particular, if we know generators of $H^\perp$, then we can find h in classical polynomial time.*

Lemma 3. *The output of the following circuit is a uniformly random chosen element of A such that $FT_A(y) \in H^\perp$*



By the same reasoning as for Simon's problem, only a few applications of this circuit are necessary to find a generating set of $H^\perp$.

[7]Graph Isomorphism in Quasipolynomial Time