Survey: Quantum Computation and The Hidden Subgroup Problem

Luis von Ahn
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
biglou+@cs.cmu.edu

Abstract

The hidden subgroup problem is fundamental in quantum computation. We give a survey of its applications and of the known quantum algorithms for its solution. We also give a brief tutorial of the mathematics necessary to understand the current research related to the hidden subgroup problem.

1 Introduction

1.1 Overview

The Hidden Subgroup Problem (HSP) has many useful applications. For abelian groups, an efficient solution to this problem implies efficient algorithms for factoring and for finding the discrete logarithm—such algorithms render most current cryptosystems useless! For nonabelian groups, an efficient solution would imply a polynomial-time algorithm for deciding if two graphs are isomorphic or not: a very useful tool in computational biology.

There is no known efficient solution to this problem with classical computers. However, several cases of the problem are known to be efficiently solvable using quantum computation. And, in fact, *so far every problem that can be solved exponentially faster with quantum computers than with classical computers is an instance of the Hidden Subgroup Problem.* We could even go as far as conjecturing that a quantum computer is nothing but a classical computer with an oracle for HSP.

The purpose of this paper is to describe the current state of knowledge about HSP. We explain the problem, several of its applications, and the "standard method" of solution using quantum computation (this "standard method" being a generalization of the technique used in Shor's algorithm and in the abelian case of HSP). We also present a brief tutorial of group theory (including representation and character theory), necessary to understand the algorithms and the current research on the problem. The emphasis was on clarity: the tutorial on representation theory and Fourier transforms is *necessary* in order to get a good intuition and understanding of the main algorithm presented in this paper. The tutorial also allows us to be brief in the description of the algorithm.

1.2 Background and Notation

1.2.1 Group Theory

The hidden subgroup problem deals with finite groups. In this section, we outline the fundamental picture of a finite group and the notation we want the reader to have throughout the paper. For a complete introduction to group theory, see [6].

Let G be a finite group, with identity element 1 and with group operation $(s, t) \mapsto st$. Recall that a subgroup H of G is a subset of G which is closed under the group operation. A (left) *coset* of H is a set of the form $gH = \{gh : h \in H\}$, with $g \in G$. Notice that $|H| = |gH|$ (the map $h \mapsto gh$ is a bijection—recall the group cancellation property). For $g_1, g_2 \in G$, g_1H and g_2H are either identical or disjoint: if $g_1g_2^{-1} \in H$, then $g_1H = g_2H$; otherwise g_1H and g_2H are disjoint.

The basic picture of a group necessary for this paper is depicted in figure 1. In particular, for any subgroup H of G, *G can be decomposed as the disjoint union of cosets of H*: every element of G is in one and only one coset of H. Also, the total number of *different* cosets of H in G, denoted by $[G : H]$, is given by $[G : H] = \frac{|G|}{|H|}$.

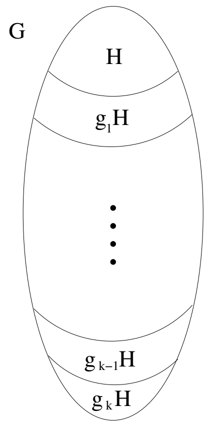

Figure 1: Decomposition of G into disjoint cosets of H

For $g \in G$ and H a subgroup of G, we will use the symbol gHg^{-1} to denote the set $gHg^{-1} = \{ghg^{-1} | h \in H\}$. Recall that a subgroup H of G is *normal* if $H = gHg^{-1}$ for all $g \in G$.

An important concept (especially when dealing with groups in computer science) is that of a set of generators. In any group G a subset, S, of elements with the property that every element of G can be written as a product of elements of S and their inverses is called a *set of generators* of G and we shall indicate this by $G = \langle S \rangle$. Also, given elements $g_1, ...g_k$ of a group G, we will denote by $\langle g_1, ..., g_k \rangle$ the subgroup generated by $g_1, ..., g_k$ (that is, the subgroup that results if we take all the possible products of elements of $\{g_1, ..., g_k\}$ and their inverses). If we take G to be the set of integers under addition, we can easily see that $G = \langle 1 \rangle$. The following theorems will be useful later:

Theorem 1.1 *Every group G of size $|G| > 1$ has a generating set of size at most*

$\lfloor \log_2 |G| \rfloor$.

Proof: Let $g_1 \in G$ be such that $g_1 \neq 1$. Then $|\langle g_1 \rangle| \geq 2$ (at least g_1 and g_1^2 are in $\langle g_1 \rangle$). If $G \neq \langle g_1 \rangle$ then let $g_2 \in G - \langle g_1 \rangle$. We now have that $|\langle g_1, g_2 \rangle| \geq 2^2$ ($g_1, g_1^2, g_2 g_1$, and $g_2 g_1^2$ are all different). If $\langle g_1, g_2 \rangle \neq G$ then let $g_3 \in G - \langle g_1, g_2 \rangle$. We now have that $\langle g_1, g_2, g_3 \rangle \geq 2^3$. If we continue this procedure, we see that the conclusion follows. ∎

Corollary 1.2 *The number of subgroups of G is at most $2^{\log^2 |G|}$.*

Proof: Every subgroup of G has a generating set of size at most $\lfloor \log_2 |G| \rfloor$, so the number of subgroups is at most $|G|^{\lfloor \log_2 |G| \rfloor} \leq 2^{\log^2 |G|}$. ∎

1.2.2 Representation Theory

The main tool used by polynomial time quantum algorithms for HSP is the Fourier transform. To define the Fourier transform (over a group), we require the basic elements of representation theory, introduced below. In particular, we introduce representations, characters, and the fundamental correspondence between these two. A more thorough treatment of these topics can be found in [3].

Throughout, we let V be a vector space over the field $\mathbf{C}$ of complex numbers, and we let $GL(V)$ be the group of isomorphisms (or linear invertible mappings) of V onto itself. We will always assume that V is finite dimensional. (Recall that if V is of dimension n, then $GL(V)$ is "the same" as the group [under matrix multiplication] of $n \times n$ invertible matrices with complex coefficients.)

The main idea of linear representation theory is that finite groups can be identified by (or *represented* as) subgroups of $GL(V)$; that is, group elements can be represented as linear invertible mappings in a complex vector space V (or, equivalently, group elements can be represented as invertible matrices with complex coefficients). Let's start with the formal definition of a representation:

Definition 1.1 *Let G be a finite group, and let V be a finite dimensional vector space over $\mathbf{C}$. A linear representation of G in V is a homomorphism ρ from the group G into the group $GL(V)$. In other words, we associate with each element $g \in G$ an element $\rho(g)$ of $GL(V)$ in such a way that we have the equality $\rho(gh) = \rho(g) \cdot \rho(h)$ for $g, h \in G$. (We will usually write ρ_g instead of $\rho(g)$.) If the dimension of V is n, then we say that ρ is a representation of degree n.*

Observe that the preceding definition implies that $\rho(1) = I$ and $\rho(s^{-1}) = \rho(s)^{-1}$. When ρ is given, we say that V is a *representation space* of G. Most of the time, however, we will abuse the language and say that V is a *representation* of G.

Example 1: The Trivial Representation. The trivial representation $\mathbf{1}_G$ maps every group element $g \in G$ to the identity $1 \in \mathbf{C}^*$ (here $\mathbf{C}^*$ denotes the multiplicative group of non-zero complex numbers). Equivalently, $\mathbf{1}_G$ maps every group element $g \in G$ to the 1 by 1 matrix (1).

Example 2: The Regular Representation. Let V be a vector space of dimension $|G|$, with basis $\{e_h\}_{h\in G}$ indexed by the elements h of G. The regular representation $\mathrm{reg}_G : G \to GL(V)$ results if we let $\mathrm{reg}_G(g)$ be the linear map of V into V which sends e_h to e_{gh}.

We now introduce the concept of two representations being "the same":

Definition 1.2 *Let ρ and ρ' be two representations of the same group G in vector spaces V and V'. These representations are said to be isomorphic if there exists a linear isomorphism $\tau : V \to V'$ which "transforms" ρ into ρ', that is, which satisfies the identity $\tau \circ \rho(g) = \rho'(g) \circ \tau$ for all $g \in G$.*

The preceding definition might be ellucidated if we consider ρ and ρ' as mappings from G to the group of invertible complex *matrices*. Let R_g be the matrix corresponding to ρ_g and R'_g be the matrix corresponding to ρ'_g. To say that ρ and ρ' are isomorphism is to say that there exists an invertible matrix T such that $T \cdot R_g = R'_g \cdot T$ for all $g \in G$. That is:

$$R'_g = T \cdot R_g \cdot T^{-1}.$$

Next is the concept of a subrepresentation:

Definition 1.3 *Let $\rho : G \to GL(V)$ be a linear representation and let W be a vector subspace of V. Suppose that W is stable under the action of G: $x \in W$ implies $\rho_g x \in W$, for all $g \in G$. The restriction ρ_g^W of ρ_g to W is then an isomorphism of W onto itself, and we have $\rho_{gh}^W = \rho_g^W \cdot \rho_h^W$. Thus $\rho^W : G \to GL(W)$ is a linear representation of G in W. W is said to be a subrepresentation of V.*

Example 3: Let $\rho = \mathrm{reg}_G$ be the regular representation of G, and let W be the subspace of dimension 1 of V spanned by $x = \sum_{h\in G} e_h$. Then we have that $\rho_g(x) = x$ for all $g \in G$. Consequently we have that ρ^W is a subrepresentation of ρ. ρ^W is isomorphic to the trivial representation.

Recall a concept from linear algebra: direct sum decompositions. Let W and W' be two subspaces of V. V is said to be the *direct sum* of W and W' (written $V = W \oplus W'$) if each $x \in V$ can be written uniquely in the form $x = w + w'$, with $w \in W$ and $w' \in W'$. (We say that W' is the complement of W in V.) A given W can have more than one complement W' in V. The next theorem (which we state without proof) will allow us to decompose *representations* as direct sums:

Theorem 1.3 *Let $\rho : G \to GL(V)$ be a linear representation of G in V and let W be a vector subspace of V stable under G. Then there exists a complement W^0 of W in V which is stable under G.*

Let W and W' be stable under G and such that $V = W \oplus W'$. Then we say that V (as a *representation*) is the direct sum of W and W'. If W and W' are given in matrix form by R_g and R'_g, $W \oplus W'$ is given in matrix form by

$$\begin{pmatrix} R_s & 0 \\ 0 & R'_s \end{pmatrix}$$

We say that representation V is *decomposed* as the direct sum of representations W and W'.

Definition 1.4 *Let $\rho : G \to GL(V)$ be a linear representation of G. We say that ρ is irreducible if V is not 0 and if no vector subspace of V is stable under G, except of course* 0 *and V.*

EXAMPLE 4: Every representation ρ of degree 1 is irreducible: the only subspaces of a 1-dimensional space V are 0 and V itself.

Much like prime numbers, irreducible representations can be used to construct *every* representation by means of direct sums:

Theorem 1.4 *Every representation of G is a direct sum of irreducible representations of G.*

Proof: Let V be a linear representation of G. We proceed by induction on the dimension of V. If $dim(V) = 0$, then the theorem is obvious (0 is the sum of the *empty family* of irreducible representations). Assume then that $dim(V) \geq 1$. If V is irreducible, then there is nothing to prove. Otherwise (by theorem 1.3), V can be written as $V = W \oplus W'$ where both W and W have degree smaller than $dim(V)$. By the induction hypothesis W and W' are direct sums of irreducible representations, and so the same is true of V. ∎

There is a sense in which this decomposition of a representation into irreducibles is unique. We will have more to say about this later.

For now, we introduce a fundamental concept in representation theory: the character of a representation. Characters are fundamental because the study of representations can be *completely reduced* to the study of their characters; that is (and we will show this!), all that we need to know to completely specify a representation is its character.

Definition 1.5 *Let $\rho : G \to GL(V)$ be a linear representation. For each $g \in G$, let $\chi_\rho(g) = Tr(\rho_g)$. The complex valued function χ_ρ on G thus obtained is called the character of the representation ρ.*

The next two theorems provide a few basic facts about characters.

Theorem 1.5 *If χ is the character of a representation ρ of degree n, we have:*

(i) $\chi(1) = n$.

(ii) $\chi(g^{-1}) = \chi(g)^*$ *for all* $g \in G$ *(a^* is the conjugate of a in* $\mathbf{C}$*).*

(iii) $\chi(hgh^{-1}) = \chi(g)$ *for all* $g, h \in G$.

Proof: (i) is trivial: $\rho(1)$ is the identity. For (ii) we observe that ρ_g has finite order; consequently the same is true of its eigenvalues $\lambda_1, ..., \lambda_m$, and so these have absolute value equal to 1. Thus

$$\chi(g)^* = Tr(\rho_g)^* = \sum \lambda_i^* = \sum \lambda_i^{-1} = Tr(\rho_g^{-1}) = Tr(\rho_{g^{-1}}) = \chi(g^{-1}).$$

(iii) follows from the fact that $Tr(ab) = Tr(ba)$, valid for any two arbitrary linear mappings a and b of V into itself. (Note that this also implies that the characters of two isomorphic representations are the same.) ■

Theorem 1.6 *Let* $\rho^1 : G \to GL(V_1)$ *and* $\rho^2 : G \to GL(V_2)$ *be two linear representations of* G*, and let* χ_1 *and* χ_2 *be their characters. Then the character* χ *of the direct sum representation* $V_1 \oplus V_2$ *is equal to* $\chi_1 + \chi_2$*.*

Proof: Let us be given ρ^1 and ρ^2 in matrix form: R_g^1, R_g^2. The representation $V_1 \oplus V_2$ is given by

$$R_g = \begin{pmatrix} R_g^1 & 0 \\ 0 & R_g^2 \end{pmatrix}$$

Then $Tr(R_g) = Tr(R_g^1) + Tr(R_g^2)$. This proves the assertion. ■

EXAMPLE 5: The character r_G of the regular representation is given by the formulas:

- $r_G(1) = |G|$
- $r_G(g) = 0$ if $g \neq 1$.

This is because if $g \in G$ is such that $g \neq 1$, then we have $gh \neq h$ for all $h \in G$, which shows that the diagonal terms of the matrix of $\text{reg}_G(g)$ are zero; in particular we have $Tr(\text{reg}_G(g)) = 0$. On the other hand, for $g = 1$, we have $Tr(\text{reg}_G(g)) = Tr(1) = |G|$.

The next theorem defines an inner product which turns out to be crucial to the development of our theory. It will eventually allow us to prove that the decomposition given by theorem 1.4 is unique.

Theorem 1.7 *If* ϕ *and* ψ *are two complex valued functions on* G*, let* $(\phi|\psi) = \frac{1}{|G|}\sum_{g\in G} \phi(g)\psi(g)^*$*. Then* $(\cdot|\cdot)$ *is an inner product.*

Proof: We only need to verify three properties:

- $(\cdot|\cdot)$ is linear in the first argument: $(c_1\phi_1 + c_2\phi_2, \psi) = \frac{1}{|G|}\sum_{g\in G}(c_1\phi_1 + c_2\phi_2)(g)\psi(g)^* = \frac{1}{|G|}\sum_{g\in G} c_1\phi_1(g)\psi(g)^* + \frac{1}{|G|}\sum_{g\in G} c_2\phi_2(g)\psi(g)^* = c_1(\phi_1|\psi) + c_2(\phi_2|\psi)$.

- $(\phi|\psi)^* = (\psi|\phi)$.
- $(\phi|\phi) \geq 0$, with equality if and only if $\phi = 0$: $aa^* \geq 0$ for any complex number a, with equality if and only if $a = 0$.

∎

We now show why the inner product just defined is interesting. The proof of the following theorem can be found in the appendix.

Theorem 1.8 *(Orthogonality Relations)*

1. *If χ is the character of an irreducible representation, we have $(\chi|\chi) = 1$.*
2. *If χ and χ' are the characters of two non-isomorphic irreducible representations, we have $(\chi|\chi') = 0$.*

Theorem 1.9 *Let V be a linear representation of G, with character ϕ, and suppose V decomposes into a direct sum of irreducible representations $V = W_1 \oplus ... \oplus W_k$. Then, if W is an irreducible representation with character χ, the number of W_i isomorphic to W is equal to $(\phi|\chi)$.*

Proof: Let χ_i be the character of W_i. By theorem 1.6, we have: $\phi = \chi_1 + ... + \chi_k$. Thus $(\phi|\chi) = (\chi_1|\chi) + ... + (\chi_k|\chi)$. But, according to theorem 1.8, $(\chi_i|\chi)$ is equal to 1 or 0, depending on whether W_i is isomorphic to W. The result follows. ∎

Corollary 1.10 *The number of W_i isomorphic to W does not depend on the chosen decomposition.*

Proof: Indeed, $(\phi|\chi)$ does not depend on the decomposition. ∎

It is in this sense that a representation's decomposition into irreducibles in unique. The next corollary shows that the study of representations can be completely reduced to the study of their characters:

Corollary 1.11 *Two representations with the same character are isomorphic.*

Proof: From corollary 1.10, if two representations have the same character, they have the same decompositions into irreducibles. This is enough for the assertion. ∎

We now show why the regular representation is interesting: it contains all irreducible representations! Though the next result is interesting in its own right, the main reason we include it is because it allows us to prove Theorem 1.13.

Lemma 1.12 *Every irreducible representation W_i of G is contained in the regular representation of G with multiplicity equal to its degree n_i.*

Proof: Let W_i be an irreducible representation of G with character χ_i. Then (by example 5) $(\chi_i|r_G) = \frac{1}{|G|}(Tr(1) \cdot |G|) = n_i$ ∎

Theorem 1.13 *Let $W_1, ..., W_k$ be all the (different) irreducible representations of G and let $n_1, ..., n_k$ be their associated degrees. Then $\sum_{i=1}^k n_i^2 = |G|$.*

Proof: The dimension of the regular representation is $|G|$. Also, this representation can be decomposed (by lemma 1.12) as the following direct sum:

$$n_1 W_1 + n_2 W_2 + ... + n_k W_k$$

The dimension of this decomposition is $\sum_{i=1}^k n_i^2$. ■

1.2.3 Fourier Transforms Over Groups

From now on we let $\hat{G} = \{\rho_1, ..., \rho_k\}$ be the set of all the irreducible representations of G. We may assume, without loss of generality, that $\rho_1, ..., \rho_k$ are unitary *matrices.* Why? Assume $\rho : G \to GL(V)$ is a representation, and assume that V is endowed with an inner product (x, y) satisfying the usual conditions: linearity in x, semi-linearity in y, and $(x, x) > 0$ if $x \neq 0$. Then we can define a new inner product $(x|y)_G$ on V by $(x|y)_G = \sum_{h \in G} (\rho_h x, \rho_h y)$. This new inner product is invariant under G: $(\rho_g x | \rho_g y)_G = (x|y)_G$. Hence, if (e_i) is an orthonormal basis of V, the matrix of ρ_g with respect to this basis in a unitary matrix.

Definition 1.6 *Let $f : G \to \mathbf{C}$. The Fourier transform of f at the irreducible representation ρ (of dimension d_ρ) is the $d_\rho \times d_\rho$ matrix*

$$\hat{f}(\rho) = \sqrt{\frac{d_\rho}{|G|}} \sum_{g \in G} f(g)\rho(g).$$

We refer to the collection $\langle \hat{f}(\rho) \rangle_{\rho \in \hat{G}}$ as the Fourier transform of f.

Thus f is mapped into $|\hat{G}|$ matrices of different dimensions. The total number of entries in these matrices is $\sum_{i=1}^k n_i^2 = |G|$ (by theorem 1.13). Hence, we can think of the Fourier transform as mapping the $|G|$ complex numbers $\langle f(g) \rangle_{g \in G}$ to $|G|$ complex numbers organized into matrices. It is not very hard to see that the Fourier transform is linear in f.

We can think of the Fourier transform over G as a linear map from $\mathbf{C}^G$ (the space of complex linear combinations of group elements) to a space of dimension $|G|$; the map being defined by the following: $g \in G$ goes to a vector with value $\sqrt{\frac{d_\rho}{|G|}} \rho_{ij}(g)$ in the ρ, i, j entry ($\rho_{ij}(g)$ indicates the i, j entry in $\rho(g)$). Thinking of the Fourier transform in this way, it is easy to show that it is a unitary operation: Let F be the matrix for this Fourier transform (with respect to the basis G of $\mathbf{C}^G$); then each column of F corresponds to all the values $\sqrt{\frac{d_\rho}{|G|}} \rho_{ij}(g)$ of a single group element g, and if c_g, c_h are two distinct columns (corresponding to $g, h \in G$), we have that:

$$c_g \cdot c_h^* = \frac{1}{|G|} \sum_{\rho, i, j} d_\rho \rho_{ij}(g) \rho_{ij}(h)^*$$

$$= \frac{1}{|G|} \sum_{\rho} d_\rho \sum_i \sum_j \rho_{ij}(g) \rho_{ij}(h)^*$$

$$= \frac{1}{|G|} \sum_{\rho} d_\rho \sum_i \rho_{ii}(gh^{-1})$$

$$= \frac{1}{|G|} \sum_{\rho} d_\rho \chi_\rho(gh^{-1})$$

$$= \frac{r_G(gh^{-1})}{|G|} = 0$$

Similarily, $c_1 \cdot c_1^* = 1$.

In the quantum setting we identify the superposition $\sum_{g \in G} f_g |g\rangle$ with the function $f : G \to \mathbf{C}$ defined by $f(g) = f_g$. In this notation, $\sum_{g \in G} f_g |g\rangle$ is mapped under the Fourier transform to $\frac{1}{\sqrt{|G|}} \sum_{\rho \in \hat{G}, 1 \leq i,j \leq d_\rho} \sqrt{d_\rho} \sum_{g \in G} f(g) \rho_{ij}(g) |\rho, i, j\rangle$.

2 The Hidden Subgroup Problem

We are finally ready to start working with the HSP. Formally:

Definition 2.1 *Let f be a function from a finitely generated group G to a finite set X such that f is constant on the cosets of a subgroup K, and distinct on each coset. The Hidden Subgroup Problem is to find a generating set for K given a way of computing f.*

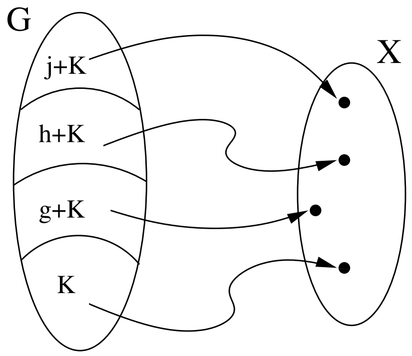

Figure 2: The function f is constant on the cosets of K and distinct on each coset

Notice that not all finitely generated groups are finite: $\mathbf{Z}$, the group of integers under addition, is finitely generated. In this paper we will deal only with finite groups; however, all our algorithms and analyses can be extended to finitely generated groups. (Notice that non-finitely generated groups are out of the question when it comes to computer science: how can they be represented?)

We want to find an *efficient* algorithm to solve HSP. In order to talk about efficiency, however, we need to state what it means for an algorithm that solves HSP to be efficient. We will mean this: An algorithm that solves HSP is said to be *efficient* if it runs in time polynomial in $\log_2[G:K]$ (recall that $[G:K]$ is the number of different cosets of K in G).

3 Applications of HSP

Before introducing an efficient algorithm for solving the HSP (for certain types of groups), we briefly outline some of the implications of an algorithm that solves the general HSP. This section can be skipped without loss of continuity.

The reason for caring so much about the HSP is that it encompasses several important problems: we will see that factoring, finding discrete logarithms, finding graph isomorphisms, and many other problems can be expressed as instances of HSP (each one with a different underlying group).

3.1 Simon's Problem

We start with Simon's problem. Recall that $\mathbf{Z}_2^n$ is the group of binary strings of length n under bitwise *xor* (bitwise *xor* will be denoted by $\oplus$). Formally:

Definition 3.1 *Let $f : \mathbf{Z}_2^n \to \mathbf{Z}_2^n$ be a 2-to-1 function with periodicity ε (that is, $f(x) = f(y)$ if and only if $y = x \oplus \varepsilon$). Simon's problem is to find ε given a way of computing f.*

The idea is that if you are given a black box for computing a function that "hides" a value ϵ, you want to find a way to query this black box repeatedly in such a way that you can find ϵ.

We say that an algorithm that solves Simon's problem is efficient if it runs in time polynomial in n (the number of bits of ϵ). Notice that an algorithm that simply evaluates f at half (plus one) the values in $\mathbf{Z}_2^n$ is guaranteed to solve Simon's problem; this algorithm, however, runs in time greater than $2^{n/2}$ steps and is therefore not efficient.

We now show that an efficient algorithm to HSP implies an efficient algorithm for solving Simon's problem: Let $f : \mathbf{Z}_2^n \to \mathbf{Z}_2^n$ be the 2-to-1 function with periodicity ε from Simon's problem. Notice that f is constant on the cosets of the subgroup $\{0, \epsilon\}$ of $\mathbf{Z}_2^n$ and distinct on each coset. Simon's problem is to find ϵ, which is the only generator for $\{0, \epsilon\}$. Simon's problem, then, is just a HSP with $G = \mathbf{Z}_2^n, X = \mathbf{Z}_2^n$ and f as given above. So, an efficient solution to HSP would yield f in time less than $\log_2 2^n = n$ steps.

3.2 Order Finding

We now show that factoring reduces to the HSP. We do this by showing that an efficient solution to the HSP yields an efficient algorithm for finding the order of elements in $\mathbf{Z}_n^*$ (recall that $\mathbf{Z}_n^*$ is the group of non-negative integers relatively prime to n (and smaller than n) with respect to multiplication modulo n).

Definition 3.2 *Let n be a natural number and y be relatively prime to n. The problem of Order Finding is to find the smallest positive integer r for which $y^r = 1$ $(mod\ n)$.*

An efficient algorithm that solves the problem of order finding is one which runs in time polynomial in $\log_2(n)$. We now show how order finding reduces to HSP. Let $f : \mathbf{Z} \to \mathbf{Z}_n$ be defined by $f(x) = y^x$ $(mod\ n)$ (recall that $\mathbf{Z}_n$ is the set $\{0, 1, ..., n-1\}$). Notice that $\mathbf{Z}$ is a finitely generated. Notice also that f is constant on the cosets of the subgroup $\{..., -2r, -r, 0, r, 2r, ...\}$ of $\mathbf{Z}$ and distinct on each coset (here r is the smallest positive integer for which $y^r = 1$ $(mod\ n)$). The problem of Order Finding is to find r. This is equivalent to finding $-r$. Since r and $-r$ are the only generators of $\{..., -2r, -r, 0, r, 2r, ...\}$ in $\mathbf{Z}$, we see that order finding is just a HSP with $G = \mathbf{Z}, X = \mathbf{Z}_n$ and f as defined above. Hence, an efficient solution to the HSP would yield a solution to order finding that runs in time polynomial in $\log_2(n)$ steps (n is an upper bound to the number of cosets of $\{..., -2r, -r, 0, r, 2r, ...\}$ in $\mathbf{Z}$, since r must be smaller than n).

It can be easily shown that en efficient solution to the problem of order finding yields an efficient algorithm for factoring. And, an efficient algorithm for factoring would render a big chunk of the current cryptosystems *useless*.

3.3 Discrete Logarithms

Solving the discrete logarithm problem would also have major implications for cryptography. The problem can be defined as follows:

Definition 3.3 *Let a, b, s, n be integers such that $b = a^s$ $(mod\ n)$. The Discrete Logarithm Problem is to find s given a, b and n.*

(An efficient solution is one that runs in time polynomial in $\log_2(n)$.) Showing that the discrete logarithm problem reduces to the HSP is a bit more complicated than it was for factoring and Simon's problem. This time, we consider the function $f : \mathbf{Z}_r \times \mathbf{Z}_r \to \{a^j (mod\ n) | j \in \mathbf{Z}_r\}$ defined by $f(x_1, x_2) = a^{sx_1+x_2}(mod\ n)$. Here r is the smallest positive integer such that $a^r = 1(mod\ n)$. Notice that this function is constant on the cosets of $\langle(1, -s)\rangle$ and different on each coset: for any integer l, $f(x_1+l, x_2-ls) = f(x_1, x_2)$. A solution to the HSP yields, in this case, a generator for $\langle(1, -s)\rangle$, which in turn yields s.

So how do we (efficiently) solve the problem of discrete logarithms given an efficient algorithm for the HSP? We are given a, b and n. We first use the solution to the HSP to find the order r of a modulo n (this can be done in time polynomial in $\log_2(n)$ as was shown in the previous section). We then construct a way of calculating the function $f : \mathbf{Z}_r \times \mathbf{Z}_r \to \{a^j (mod\ n) | j \in \mathbf{Z}_r\}$ defined by $f(x_1, x_2) = b^{x_1} a^{x_2} (mod\ n)$. Finally, we feed this into the HSP algorithm to find s.

3.4 Graph Isomorphism

Determining if two graphs are isomorphic has important consequences for computational biology. We now show that an efficient solution to the HSP would yield an efficient way of determining if two graphs are isomorphic. Recall first the definition of an isomorphism between two graphs:

Definition 3.4 *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. A bijection $f : V_1 \to V_2$ is called an isomorphism between G_1 and G_2 if the following holds for all $u, v \in V_1$: $(u, v) \in E_1$ if and only if $(f(u), f(v)) \in E_2$.*

We can now formally state the problem of Graph Isomorphisms:

Definition 3.5 *Let G_1, G_2 be two graphs with n vertices. The Graph Isomorphism Problem is to decide whether G_1 and G_2 are isomorphic.*

Reducing Graph Isomorphism to the HSP is even more complicated than reducing Discrete Logarithms to the HSP. The reduction will in fact be in steps: first we reduce the problem of Automorphism Generators to the HSP and then we reduce Graph Isomorphism to the problem of Automorphism Generators.

We start with a definition:

Definition 3.6 *Let $G = (V, E)$ be a graph. An automorphism of G is a bijection $f : V \to V$ such that the following holds for all $u, v \in V$: $(u, v) \in E$ if and only if $(f(u), f(v)) \in E$.*

The reason we introduce automorphisms is because the automorphisms of a graph from a group:

Theorem 3.1 *Let G be a graph with n vertices. Let $Aut(G)$ be the set of all automorphisms of G. Then $Aut(G)$ is a subgroup of S_n (recall that S_n, the symmetric group, is the group of all bijections from the set $\{1, ..., n\}$ to itself; the group operation here is function composition).*

Proof: Clearly $Aut(G)$ is inside S_n (or an isomorphic copy of S_n), since $Aut(G)$ consists of bijections from a set of n elements to itself. It is very easy to check that $Aut(G)$ is indeed a group under function composition. ■

We can think of S_n as acting on a graph G of order n as follows: Each element σ of S_n is a permutation (or re-ordering) of the numbers $\{1, 2, ..., n\}$; we can think of $\sigma(G)$ as a relabeling of the n vertices of G. For example, if σ is the permutation that sends $\{1, 2, 3, 4\}$ to $(4, 3, 2, 1)$ and G is as shown in figure 3, then $\sigma(G)$ is as shown in figure 4.

A good way of visualizing an automorphism on G is by thinking about this action of S_n on G. Elements of S_n relabel the vertices of G: We start with a graph G (as that in figure 3) and then, without touching any of the edges, we simply re-label the vertices—the result of this process is shown in the left side of figure 4. Afer this, we can move the vertices to their original positions; the result of this movement is shown in the right side of figure 4. A automorphism of a graph

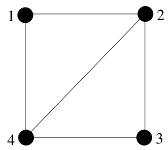

Figure 3: The graph G

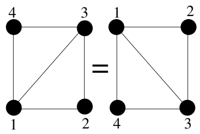

Figure 4: The graph $\sigma(G)$

is an element of S_n for which the initial and final graphs of this process perfectly coincide. We denote by $fix(\sigma(G))$ the picture of the graph that results after this procces (so, in our example, the picture on the right side of Figure 4 is $fix(\sigma(G))$). With this notation, σ is an automorphism if and only if $fix(\sigma(G)) = fix(e(G))$, where e is the identity permutation on S_n.

Definition 3.7 *Given a graph G, the Automorphism Generators Problem is to find a set of generators for $Aut(G)$.*

The Automorphism Generators problem is a HSP: Let G be a graph with n vertices; label its vertices from 1 to n. Let f_G be a function on S_n that maps each element σ to $fix(\sigma(G))$. Then f_G is constant on the cosets of $Aut(G)$ and different on each coset.

Finally, we can show that an efficient algorithm for solving the Automorphism Generators Problem implies an efficient algorithm for solving the Graph Isomorphism Problem:

Lemma 3.2 *Let G_1, G_2 be two connected graphs with n vertices, and let $G = G_1 \cup G_2$. Then G_1 and G_2 are isomorphic if and only if there exist $g_1 \in G_1, g_2 \in G_2$ and σ_i a generator of $Aut(G)$ such that $\sigma_i(g_1) = g_2$.*

The proof of this last lemma is left as an exercise to the reader.

4 The Standard Method of Solution

There is no known quantum algorithm for efficiently solving the general HSP. However, the HSP can be solved efficiently (using quantum algorithms) for a large number of different groups. Simon was the first to find an efficient algorithm to solve an instance of the HSP (Simon's Problem) [8]. Shor then found how to solve the problem of Order Finding using the quantum Fourier transform [9]. An efficient algorithm to solve the HSP for *all* abelian groups was then presented by Jozsa [10]. His method also relied on the quantum Fourier transform.

Here we present an algorithm that works whenever the hidden subgroup is normal in the group G. Recall that a subgroup H of G is normal if $gHg^{-1} = H$ for all $g \in G$. It is easy to see that if G is abelian, then every subgroup of G is normal.

The algorithm was first introduced by [7] and is a simple generalization of the algorithm for abelian groups.

Assume H is normal in G.

The first part of the algorithm goes through the following 4 steps:

1. Form the uniform-amplitudes superposition over group elements: $\frac{1}{|G|}\sum_{g\in G}|g,0\rangle$.
2. Compute f over this superposition to obtain: $\frac{1}{\sqrt{|G|}}\sum_{g\in G}|g,f(g)\rangle$.
3. Measure the second register. The resulting superposition is $\frac{1}{\sqrt{|H|}}\sum_{h\in H}|gh\rangle$, for some element g of G, uniformly distributed over G. (Recall that f has the same value for all elements of each coset of H, but different values for elements of different cosets.)
4. Apply the Fourier transform to the get: $\frac{1}{\sqrt{|G||H|}}\sum_{\rho,i,j}\sqrt{d_\rho}\sum_{h\in H}\rho_{ij}(gh)|\rho,i,j\rangle$. (Recall section 1.2.3.)

Let us now deviate and introduce some useful notation. For a subset S of G, define $|S\rangle = \frac{1}{\sqrt{|S|}}\sum_{g\in S}|g\rangle$ and $\rho(S) = \rho(|S\rangle) = \frac{1}{\sqrt{|S|}}\sum_{g\in S}\rho(g)$. For example, if $\mathbf{1}_G$ is the trivial represnetation, we have

$$\mathbf{1}_G(G) = \frac{1}{\sqrt{|G|}}\sum_{g\in G}\mathbf{1}_G(g) = \frac{1}{\sqrt{|G|}}\sum_{g\in G}1 = \frac{|G|}{\sqrt{|G|}} = \sqrt{|G|}.$$

Since the trivial representation is irreducible, we have that for any other representation ρ,

$$\rho(G) = \frac{1}{\sqrt{|G|}}\sum_{g\in G}\rho(g) = (\rho|\mathbf{1}_G) = 0$$

We end this deviation with a useful lemma:

Lemma 4.1 $\rho(H) = \frac{1}{\sqrt{|H|}}\sum_{h\in H}\rho(h)$ *is* $\sqrt{|H|}$ *times a projection matrix, and* $rank(\rho(H)) = \frac{1}{|H|}\sum_{h\in H}\chi_\rho(h)$.

Proof: Restricted to H, ρ decomposes into the direct sum of several irreducible representations $\sigma_1,...,\sigma_k$. $\rho(H)$ is the direct sum of $\sigma_i(H)$; as discussed above, $\sigma_i(H)$ is $\sqrt{|H|}$ if σ_i is the trivial representation of H and zero otherwise. ■

Now back to the main thread. After step 4 above (assuming the element we got in step 3 was g), the probability of measuring a given $|\rho,i,j\rangle$ is:

$$Pr_g(|\rho,i,j\rangle) = \frac{d_\rho}{|G||H|}|\sum_{h\in H}\rho_{ij}(gh)|^2 = \frac{d_\rho}{|G|}|\rho(gH)_{ij}|^2$$

Since g is unformly distributed over G in step 3, the actual probability of measuring a given $|\rho, i, j\rangle$ is:

$$Pr(|\rho, i, j\rangle) = \frac{1}{|G|} \sum_{g \in G} Pr_g(|\rho, i, j\rangle | g, H)$$

$$= \frac{1}{|G|} \sum_{g \in G} \frac{d_\rho}{|G|} |\rho(gH)_{ij}|^2$$

The idea now is that measuring ρ will give us some information about the identity of H. In fact, we will show that if we sample the above distribution a polynomial (in $\log_2 |G|$) number of times, we can identify H with very high probability.

First, we show that $Pr_g(|\rho\rangle)$, the probability of measuring ρ, is independent of g:

Lemma 4.2 *The Probability of measuring ρ is the same for the uniform superposition on the coset gH as for the superposition on H.*

Proof: $\rho(gH) = \rho(g)\rho(H)$ and $\rho(g)$ is unitary. The result now follows from the fact that:

$$Pr_g(|\rho, i, j\rangle) = \frac{d_\rho}{|G|} |\rho(gH)_{ij}|^2$$

■

Corollary 4.3 $Pr_g(|\rho\rangle) = Pr(|\rho\rangle) = \frac{d_\rho}{|G|} \sum_{h \in H} \chi_\rho(h) = \frac{|H| d_\rho}{|G|} rank(\rho(H))$.

Since the probability of sampling ρ is independent of the particular coset gH, we will concentrate on the uniform superposition on H. The following lemma is crucial in our analysis:

Lemma 4.4 *If H is a normal subgroup, $\rho(H)$ is a nonnegative scalar multiple of the identity I, nonzero if and only if $H \subset Ker(\rho)$ ($Ker(\rho)$ is the set of elements of G that are mapped by ρ to the identity).*

Proof: Let $\sigma_1, ..., \sigma_k$ be the decomposition of ρ for H. We claim that if σ_1 is trivial, so are the all the rest.

Let V be the space ρ acts on. Let W be the 1-dimensional subspace of V which σ_1 acts on. Since ρ is irreducible over G, the elements g of G carry W to a set of subspaces spanning V. We now show that H acts trivially on V: $w \in gV$ implies that $w = \rho_g(v)$, so $\rho_h w = \rho_h \rho_g v = \rho_g \rho_h v = \rho_g v = w$. The result follows form the discussion above lemma 4.1. ■

By the above lemma, ρ has non-zero probability of being sampled if and only if H is in its kernel. *The task of reconstructing H from a sequence of such samples is just that of intersecting the kernels of the sampled ρ's.* We will show that $O(\log_2 |G|)$ repetitions of the sampling process suffice so that the samples uniquely identify the hidden subgroup H.

We will proceed as follows (from [2]): The number of subgroups of G is at most $2^{\log^2 |G|}$ (corollary 1.2). We will calculate the number of samples from the distribution necessary to make the probability of mistaking some subgroup H' for the real H, less than $\frac{1}{2^{\log^2 |G|}} = (\frac{1}{2})^{\log^2 |G|}$. We will do this using a *large deviation bound.*

We begin with two uniform superpositions $|H\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |h\rangle$ and $|H'\rangle = \frac{1}{\sqrt{|H'|}} \sum_{h \in H'} |h\rangle$. The L_2 distance between $|H\rangle$ and $|H'\rangle$ is a constant because $H \cap H'$ is at most half the size of one of these groups, say H'. So, examining only the elements in $H' - H$, we get that $||H\rangle - |H'\rangle|_2^2 \geq 1/2$. Let $F|H\rangle$ and $F|H'\rangle$ be the Fourier transforms of $|H\rangle$ and $|H'\rangle$. What we want is a lower bound on the $L_!$ distance between the distributions arising from measuring these superpositions (whose probabilities are the norm squares of the amplitudes). We know that $|F|H\rangle - F|H'\rangle|_2 = ||H\rangle - |H'\rangle|_2$ (since F is unitary), and we also know that the amplitudes of $F|H\rangle$ and $F|H'\rangle$ are non-negative reals. The following lemma will imply our result:

Lemma 4.5 *Let v, w be non-negative real vectors, and denote by v^2 the vector that results from squaring the entries in v. Then:* $|v^2 - w^2|_1 \geq |v - w|_2^2$.

Proof: $|v^2 - w^2|_1 = \sum_j |v_j^2 - w_j^2| = \sum_j |v_j - w_j| \cdot |v_j + w_j| \geq \sum_j |v_j - w_j|^2 = |v - w|_2^2$. ∎

5 Epilogue

We have presented an algorithm for efficiently solving the HSP when the hidden subgroup is normal. It still remains open whether the general HSP can be solved using quantum computation. There is evidence [2] that the method described here cannot be generalized to solving the HSP when the group is S_n, for instance.

6 Appendix

6.1 Orthogonality Relations for Group Characters

Lemma 6.1 *(Schur's Lemma) Let $\rho^1 : G \to GL(V_1)$ and $\rho^2 : G \to GL(V_2)$ be two irreducible representations of G, and let f be a linear mapping of V_1 into V_2 such that $\rho_s^2 \circ f = f \circ \rho_s^1$ for all $s \in G$. Then:*

(i) If ρ^1 and ρ^2 are not isomorphic, we have $f = 0$.

(ii) If $V_1 = V_2$ and $\rho^1 = \rho^2$, f is a scalar multiple of the identity.

Proof: The case $f = 0$ is trivial, so suppose that $f \neq 0$.

We first show that $ker(f) \subset V_1$ is stable under G: Assume $v \in ker(f)$. Let $s \in G$. We will show that $f(\rho_s^1(v)) = 0$: $f(\rho_s^1(v)) = f \circ \rho_s^1(v) = \rho^2 \circ f(v) = 0$. Hence, either $ker(f) = 0$ or $ker(f) = V_1$ (as ρ_1 is irreducible).

The same argument shows that the image of f is equal to W_2 or 0.

Since $f \neq 0$ we have that the image of f is V_2 and that the kernel of f is 0. This proves (i).

Suppose now that $V_1 = V_2$ and $\rho^1 = \rho^2$, and let $lambda$ be an eigenvalue of f: there exists at least one since the field of scalars is the field of complex numbers. Let $f' = f - \lambda$. Since λ is an eigenvalue of f, the kernel of f' is not 0; on the other hand, we have $\rho_s^2 \circ f' = f' \circ \rho_s^1$. The first part of the proof shows that these properties are possible only if $f' = 0$, that is, f is equal to λ. ■

Corollary 6.2 *Let h be a linear mapping of V_1 into V_2, and let $h^0 = \frac{1}{g}\sum_{t \in G}(\rho_t^2)^{-1} h \rho_t^1$. Then:*

(i) If ρ^1 and ρ^2 are not isomorphic, we have $h^0 = 0$.

(ii) If $V_1 = V_2$ and $\rho^1 = \rho^2$, h^0 is equal to the identity times $(1/n)Tr(h)$, with $n = dim(V_1)$.

Proof: Let's show that $\rho_s^2 h^0 = h^0 \rho_s^1$. Indeed:

$$(\rho_s^2)^{-1} h^0 \rho_s^1 = \frac{1}{g}\sum_{t \in G}(\rho_s^2)^{-1}(p_t^2)^{-1} h \rho_t^1 \rho_s^1$$

$$= \frac{1}{g}\sum_{t \in G}(\rho_{ts}^2)^{-1} h \rho_{ts}^1 = h^0$$

Applying the previous lemma to $f = h^0$ we see in case (i) that $h^0 = 0$, and in case (ii) that h^0 is equal to a scalar λ. Moreover, in the latter case, we have:

$$Tr(h^0) = \frac{1}{g}\sum_{t \in G} Tr((\rho_t^1)^{-1} h \rho_t^1) = Tr(h)$$

And since $Tr(\lambda) = n \cdot \lambda$, we get $\lambda = (1/n)Tr(h)$. ■

These results can be used to prove the orthogonality relations for characters (theorem 1.8).

References

[1] Michele Mosca and Artur Ekert, The Hidden Subgroup Problem and Eigenvalue Estimation on a Quantum Computer. *arXive e-print quant-ph/9903071*, 1999.

[2] Michelangelo Grigni, Leonard J. Schulman and Umesh Vazirani, Quantum Mechanical Algorithms for the Nonabelian Hidden Subgroup Problem. Unpublished, 1997.

[3] Jean-Pierre Serre. *Linear Representation of Finite Groups.* Springer-Verlag, 1977.

[4] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[5] Rudolf Mathon. A Note on the Graph Isomorphism Counting Problem. *Information Processing Letters, Volume 8, number 3*, 1979.

[6] David Dummit and Richard Foote. *Abstract Algebra.* Prentice Hall, 1991.

[7] Sean Hallgren, Alexander Russell and Amnon Ta-Shma. Normal Subgroup Reconstruction and Quantum Computation Using Group Representations. *Thirty-Second Annual ACM Symposium on Theory of Computing (STOC)*, 2000.

[8] D. Simon. On the power of quantum computation. In *Preceedings, 35^{th} Annual Symposium on Foundations of Computer Science (FOCS)*, 1994.

[9] P. W. Shor. Algorithms for quantum computations: discrete logarithms and factoring. In *Preceedings, 35^{th} Annual Symposium on Foundations of Computer Science (FOCS)*, 1994.

[10] R. Jozsa. Quantum Algorithms and the Fourier transform. *arXive e-print quant-ph/9707033*, 1997.