Hidden Subgroups and Quantum Computation

Lecture 07

Dheeran E. Wiggins

Summer 2025
Illinois Mathematics and Science Academy

August 01, 2025

Overview

Today we will walk through the abelian HSP algorithm. We will then look at Simon's and Shor's algorithms.

We use the same notation as from the previous lecture, where G will be an abelian group (using additive notation).

Begin with the computational 0 state $|0\rangle \otimes |0\rangle$ on a pair of registers.

Apply the abelian QFT F_G on the first register:

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle \otimes |0\rangle .$$

Apply our black box function f which separates H-cosets:

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

Let $C = \{c_1, \ldots, c_m\}$ be a set of coset representatives for the subgroup $H \leq G$.

Using the fact that f is constant and is different for each H-coset, we get

$$\frac{1}{\sqrt{|C|}} \sum_{c \in C} |c + H\rangle \otimes |f(c)\rangle = \frac{1}{\sqrt{|C|}} \sum_{c \in C} \tau_c |H\rangle \otimes |f(c)\rangle .$$

Applying F_G to the first register again, we get

$$\frac{1}{\sqrt{|C|}} \sum_{c \in C} F_G \tau_c \left| H \right\rangle \otimes \left| f(c) \right\rangle .$$

Using the relation $F_G \tau_c = \varphi_c F_G$, and the fact that F_G takes $|H\rangle$ to $|H^\perp\rangle$, we simplify to

$$\frac{1}{\sqrt{|C|}} \sum_{c \in C} \varphi_c \, |H^\perp\rangle \otimes |f(c)\rangle \, .$$

Further, since $|G|/|H| = |H^\perp|$, we know that $|C| = |H^\perp|$, so we have

$$\frac{1}{\sqrt{|H^\perp|}} \sum_{c \in C} \varphi_c \left| H^\perp \right\rangle \otimes \left| f(c) \right\rangle .$$

Perform a measurement on the first register. This returns, in a uniformly distributed manner, a random element of $H^{\perp}$.

Choosing $c + \lceil \log |\Sigma| \rceil$ elements of a group Σ will generate Σ, with probability bounded below by $1 - 2^{-c}$.

By sampling solutions of a system of equations, based on a supposed generating set of $H^{\perp}$, and using a diagonalization argument, we may determine generators of H with probability no less than

$$(1-2^{-c})(1-2^{-c'}),$$

where we run the algorithm $c + \lceil \log |G| \rceil$ times, sampling solutions to the system to get $c' + \lceil \log |G| \rceil$ samples of H.

Assigned reading: the last page of §3.5 of Lomont's review to work through the details of the sampling procedure and the time complexity.

`https://arxiv.org/pdf/quant-ph/0411037`.

Thus, there is a quantum algorithm which outputs a generating set for the hidden subgroup $H \leq G$ with probability no less than

$$1 - |G|^{-1}.$$

It uses $O(\log|G|)$ calls of f, running in time polynomial in $\log|G|$ and the time to compute f, using a circuit of size

$$O(\log|G| \log\log|G|).$$

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Simon's algorithm asks for the following:

GIVEN a function $f : (\mathbb{Z}/2)^n \to (\mathbb{Z}/2)^m$, where $m \geq n$, and so that there is a constant $s \in (\mathbb{Z}/2)^n$ such that $f(x) = f(x')$ if and only if $x = x' \oplus s$.

FIND the constant $s \in (\mathbb{Z}/2)^n$.

Here, per usual, $\oplus$ is componentwise, binary addition.

Then, our subgroup which is fixed by the black box function is $H = \{0, s\}$. Using the abelian HSP algorithm, we can find it efficiently.

On the other hand, the classical solution would involve calling the function $O(|G|)$ times to find s!

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

RSA public key cryptography uses integer factorization, which is extraordinarily difficult to compute classically.

Shor's factoring algorithm looks to factor some composite integer $N > 0$. It suffices to look for a nontrivial solution to $x^2 \equiv 1 \pmod{N}$, since then $(x+1)$ or $(x-1)$ factors into N.

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

The order of an integer x modulo N is the smallest power r such that $x^r \equiv 1 \pmod{N}$.

Randomly choosing an integer y such that $\gcd(y, N) = 1$ is likely to yield y with even order, so one solution is $x = y^{r/2}$.

Besides computing order, we have efficient classic algorithms for the rest of the problem. Thus, we want our HSP to give us a way to efficiently compute r.

We set $f(a) \equiv x^a \pmod{N}$, so that $f(a+r) = f(a)$ for all a.
This is our black box function f.

The group is the cyclic group $\mathbb{Z}/N$, and using the cyclic HSP algorithm, we can efficiently find the generator r of the subgroup $\langle r \rangle = H$ in $\mathbb{Z}/N$.

Next time we will discuss

(i) the dihedral group D_n.

(ii) the general QFT.

(iii) the dihedral hidden subgroup problem.