The hidden subgroup problem for infinite groups

Greg Kuperberg

UC Davis

October 29, 2020

(Paper in preparation.)

Shor's algorithm

Let $f : \mathbb{Z} \to X$ be a function to a set X such that:

- We can compute f in polynomial time.
- $f(x+h) = f(x)$ for an unknown period h.
- $f(x) \neq f(y)$ when $h \nmid x-y$.

Finding h from f is the hidden period problem, or the hidden subgroup problem for the integers $\mathbb{Z}$.

Theorem (Shor) A quantum computer can solve the hidden period problem in time $\mathrm{poly}(\log h)$.

I.e., in quantum polynomial time in $\|h\|_{\mathrm{bit}}$. Note: If f is black box, then this takes $\widetilde{\Omega}(\sqrt{h}) = \exp(\Omega(\|h\|_{\mathrm{bit}}))$ classical queries.

Factoring integers

Corollary (Shor) Integers can be factored in quantum polynomial time.

Suppose that N is odd and not a prime power. Shor's algorithm reveals the order $\operatorname{ord}(a)$ of a prime residue $a \in (\mathbb{Z}/N)^\times$ via

$$f(x) = a^x \in \mathbb{Z}/N.$$

If a is random, then $\operatorname{ord}(a)$ is even and $b = a^{\operatorname{ord}(a)/2} \neq \pm 1$ with good odds, whence

$$N | b^2 - 1 = (b+1)(b-1) \qquad N \nmid b \pm 1$$

yields a factor of N.

Shor-Kitaev

In a second example of HSP, let $f : \mathbb{Z}^k \to X$ be periodic with respect to a finite-index sublattice $H \leq \mathbb{Z}^k$. (So that $f(x) = f(y)$ if and only if $x - y \in H$.) Then

Theorem (Shor-Kitaev) We can calculate H in quantum polynomial time, uniformly in k and $\|H\|_{\text{bit}}$.

Corollary (Generalized discrete logarithm) If A is an algorithmic finite abelian group, then an isomorphism

$$\phi : A \xrightarrow{\cong} (\mathbb{Z}/a_1) \times (\mathbb{Z}/a_2) \times \cdots \times (\mathbb{Z}/a_\ell)$$

can be constructed and evaluated in quantum polynomial time.

Effect on cryptography

Corollary (Generalized discrete logarithm) If A is an algorithmic finite abelian group, then an isomorphism

$$\phi : A \xrightarrow{\cong} (\mathbb{Z}/a_1) \times (\mathbb{Z}/a_2) \times \cdots \times (\mathbb{Z}/a_\ell)$$

can be constructed and evaluated in quantum polynomial time.

E.g., A can be an elliptic curve or an abelian variety over a finite field $\mathbb{F}_q$.

Computer science corollary: Quantum computers can defeat all public key cryptography which is currently standard. The goal of "post-quantum cryptography" is to remedy this with new (classical) cryptographic standards.

The hidden subgroup problem

Suppose that

$$\begin{tikzcd} G \arrow[rr, "f"] \arrow[dr, two heads] & & X \\ & G/H \arrow[ur, hook] & \end{tikzcd}$$

where G is a discrete group, f can be computed in polynomial time, and $H \leq G$ is a hidden subgroup. Then finding H from f is the hidden subgroup problem (HSP).

- If $G = \mathbb{Z}^k$ or any explicit quotient, this is Shor-Kitaev.
- Most of the other progress for HSP concerns finite groups: H normal, G almost abelian, G Heisenberg, G dihedral, etc.
- Some finite G look hard even for QC, *e.g.*, $G = S_n$.

Negative results

Theorem (K.) If $G = (\mathbb{Q}, +)$, then HSP is NP-hard.

Theorem (K.) If $G = F_k$ is non-abelian free, then normal HSP is NP-hard.

Theorem (K.) If $G = \mathbb{Z}^k$ with unary vector encoding, then HSP is uSVP-hard. (Unique short vector in a lattice.)

Note: The nature of HSP for infinite G is sensitive to how elements are encoded. We encode elements of $\mathbb{Q}$ as ordinary fractions; elements of F_k as reduced words; and in unary $\mathbb{Z}^k$ as uncompressed commutative words.

$$\frac{993470124}{6798515} \in \mathbb{Q} \qquad aba^{-1}ba \in F_2 \qquad aaaab^{-1}b^{-1}b^{-1}ccccc \in \mathbb{Z}^3.$$

Positive results

Theorem (K.) If $G = \mathbb{Z}^k$ with binary encoding and H has infinite index, then H can be found in quantum polynomial time, uniformly in k and $\|H\|_{\text{bit}}$.

Corollary If G is finitely generated abelian with efficient encoding of elements, then H can be found in quantum polynomial time.

We also get a result for $G = \mathbb{Z}^\infty$, but only with dense encoding of vectors.

Theorem (K.) If G is finitely generated, virtually abelian with efficient encoding of elements, then an arbitrary H can be found in time $\exp(\sqrt{\|H\|_{\text{bit}}})$.

This reuses my earlier result on the dihedral hidden subgroup problem, and refinements found since then.

Quantum computing in 60 seconds

For hardcore algebraists

The tensor category $(\text{set}, \times)$ is generated by the object $\mathbb{Z}/2$ together with morphisms AND, OR, NOT, and COPY called gates. (Karoubi-generated as a $\otimes$-category.) A digital circuit is then a tensor network. An algorithm in P is equivalent to a doubly periodic tensor network, or cellular automaton, with polynomially many repetitions.

For BPP, we use the category of finite stochastic maps, densely generated by a finite set of gates. For BQP, we use the category of finite quantum maps = TPCP maps acting on matrices:

$$E : M(a, \mathbb{C})^{\Delta} \to M(b, \mathbb{C})^{\Delta}$$

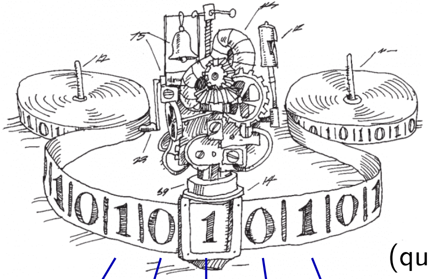
We again densely Karoubi-generate the quantum map $\otimes$-category with a finite set of gates.

Quantum computing in 60 seconds

For hardcore CS theorists

We can model a quantum computer as a (classical) Turing machine with together with a tape of qubits. The TM can:

- initialize or measure individual qubits
- apply unitary operators to pairs of qubits



$|\psi\rangle \in \mathbb{C}^2$

(qubits also entangled)

$|\psi\rangle \quad |\psi\rangle \quad |\psi\rangle \quad |\psi\rangle \quad |\psi\rangle$

A little complexity zoo

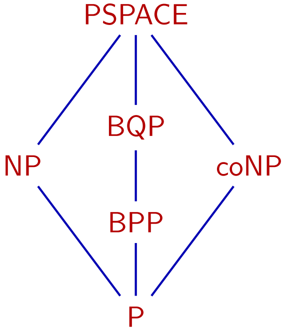
A complexity class is a set of decision or function or decision problems

$$d : \{\text{inputs}\} \to \{\text{yes}, \text{no}\} \qquad f : \{\text{inputs}\} \to \{\text{outputs}\}$$

reachable with particular complexity resources.

- P $=$ deterministic polynomial time
- BPP $=$ randomized polynomial time, probably correct answer
- NP $=$ yes-no polynomial time the aid of a prover
- coNP $=$ like NP but with a disprover
- BQP $=$ quantum polynomial time
- PSPACE $=$ polynomial space, unrestricted time otherwise

A little complexity zoo



- These are the known inclusions.
- Conjecture: $\mathsf{P} = \mathsf{BPP}$
- Conjecture: $\mathsf{BQP} \not\subseteq \mathsf{NP} \not\subseteq \mathsf{BQP}$
- Conjecture: $\mathsf{NP} \neq \mathsf{coNP}$
- Conjecture: $\mathsf{PSPACE} \neq \mathsf{BQP}, \mathsf{NP}$
- $\mathsf{P} \neq \mathsf{PSPACE}$ is also open

All of these classes (including P vs BPP) can be distinguished in the presence of oracles or black boxes.

A problem that is NP-hard is unlikely to be in BQP.

NP hardness and HSP

A decision problem

$$d : \{\text{inputs}\} \to \{\text{yes}, \text{no}\}$$

is Post-Karp NP-hard means that every $e \in \mathsf{NP}$ can be converted to a special case of d:

$$e(x) = d(f(x)) \qquad f \in \mathsf{P}$$

There is another standard (Turing-Cook) that e can be computed with polynomially many oracle calls to d.

We must convert HSP to a decision problem for NP-hardness.

- If $G = \mathbb{Q}$, we choose "Is $H \neq \mathbb{Z}$?"
- In other cases, we choose "Is $H \neq 1$?"

HSP in $\mathbb{Q}$

$d \in$ NP means that there is a predicate $z \in$ P such that

$$d(x) = \exists y, z(x,y).$$

The data string y, with $|y| = \text{poly}(|x|)$, is a certificate.

Step 1: We can take each y to be a prime number, by using the left $1/3$ of its bits as a data string certificate. Theorem of Ingham: When n is large enough, there is a prime p such that $n^3 < p < (n+1)^3$.

Step 2: We need to make an instance of HSP in $\mathbb{Q}$ from the predicate z, so that if you can learn $H \leq \mathbb{Q}$ from $f : \mathbb{Q} \to X$, then I can use you to evaluate $d(x)$. We generate H by 1 and reciprocals of all witnesses:

$$H = \left\langle \left\{ \frac{1}{y} \mid z(x,y) = \text{yes} \right\} \cup \{1\} \right\rangle.$$

Partial fractions, for actual fractions

Step 3: We need an H-periodic function $f : \mathbb{Q} \to X$. We set $X = \mathbb{Q}$ and calculate a canonical representative $f(a/b) \in H + a/b$ for each coset of H.

Partial fractions in $\mathbb{R}[x]$, taught in calculus, can also be done in $\mathbb{Q}$:

$$\frac{x^8+5}{x^4+x} = x^4 - x - \frac{3x-2}{x^2-x+1} - \frac{2}{x+1} + \frac{5}{x}$$

$$\frac{1}{60} = -2 + \frac{1}{2} + \frac{1}{4} + \frac{2}{3} + \frac{3}{5}$$

The right side is a canonical form with terms r/p^k with $1 \leq r < p$ with p prime, plus an integer. Calculating these partial fractions requires integer factorization, but we have that in BQP!

The hiding function

To calculate $f(a/b)$, expand a/b in partial fractions:

$$\frac{1}{60} = -2 + \frac{1}{2} + \frac{1}{4} + \frac{2}{3} + \frac{3}{5}$$

Then strike the integer term, and each term r/p with p an accepted witness:

$$f\left(\frac{1}{60}\right) = \cancel{-2} + \frac{1}{2} + \frac{1}{4} + \frac{2}{3} + \cancel{\frac{3}{5}} = \frac{1}{2} + \frac{1}{4} + \frac{2}{3} = \frac{17}{12}$$

Key point: You don't need to know the accepted witnesses, you only need to be able to ask the predicate $z(x, p)$.

Conclusion: If you can calculate whether $H \not\supseteq \mathbb{Z}$ from this f, then you can calculate $d(x)$ with $d \in \mathrm{NP}$.

HSP in F_k

If $G = F_k$, the general outline is the same. Given $d \in$ NP with a predicate $z \in$ P, we make a hidden subgroup $N \trianglelefteq G$ which is normally generated by witnesses. We then define $f : G \to G$ as a canonical rep. function $f(w) \in wN$. Since N is normal, $f(w)$ is a canonical word for w in the presented group F_k/N.

We can express a witness y as a word:

$$y(a,b) = ababbbaabb$$

Let $k = 14$, and let N be generated by the relator

$$r_y = y(a_1,b_1)y(a_2,b_2)\cdots y(a_7,b_7)$$

for each accepted y. Claim: We can compute canonical words in F_{14}/N without seeing relators, only with guess-and-check.

Small cancellation

Our group is

$$F_{14}/N = \langle a_1, b_1, \ldots, a_7, b_7 | \{y(a_1, b_1)y(a_2, b_2)\cdots y(a_7, b_7)\}\rangle.$$

By construction, it has $C'(1/6)$ small cancellation.

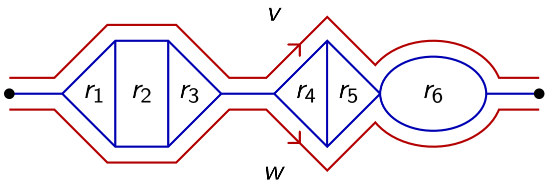
Theorem (Greendlinger) The word problem in any $C'(1/6)$ group can be solved by the greedy algorithm (Dehn's algorithm).

Theorem (Partly folklore) A word w in any $C'(1/6)$ group K can be canonicalized into shortlex form with an extended greedy algorithm.

We can also canonicalize w in polynomial time with the presentation and w as input. If $K = F_{14}/N$, it is still poly time with only guess-and-check access to relators.

Thin diagrams

A key concept in the proof is a thin equality diagram for a word equivalence $v \sim w$ modulo N. An equality (or van Kampen) diagram is a tree of disks cellulated by relators to indicate equivalence. It is thin when each relator borders both v and w.



- If $v \sim w$ are Dehn-reduced, then they have a thin diagram.
- All shortest words for w live in one thin equality diagram.
- We can build these diagrams by guess-and-check because $|r \cap v| \geq |r|/6$ for every r in the diagram.

An HSP algorithm in $\mathbb{Z}^k$

Suppose that $f : \mathbb{Z}^k \to X$ hides a sublattice $H \le \mathbb{Z}^k$ of some rank $\ell \le k$. Given two parameters $Q \gg S \gg 1$, a standard first part of a quantum algorithm for this HSP goes as follows.

1. Prepare an approximate Gaussian state on a cube in $\mathbb{Z}^k$:

$$|\psi_G\rangle \propto \sum_{\substack{\vec{x}\in\mathbb{Z}^k \\ \|\vec{x}\|_\infty < Q/2}} \exp(-\pi\|\vec{x}\|_2^2/S^2)|\vec{x}\rangle$$

2. Apply the hiding function f to $|\psi_G\rangle$ to obtain:

$$U_f|\psi_G\rangle \propto \sum_{\vec{x}} \exp(-\pi\|\vec{x}\|_2^2/S^2)|\vec{x}, f(\vec{x})\rangle$$

   Throw away the output, leaving a mixed state on $\mathbb{C}[(\mathbb{Z}/Q)^k]$.

3. Apply the quantum Fourier operator $F_{(\mathbb{Z}/Q)^k}$ and measure a Fourier mode $\vec{y}_0 \in (\mathbb{Z}/Q)^k$.

Dual samples

The quantum part of the algorithm produces a sample $\vec{y}_0 \in (\mathbb{Z}/Q)^k$ which we can rescale to obtain:

$$\vec{y}_1 = \frac{\vec{y}_0}{Q} \in (\mathbb{R}/\mathbb{Z})^k$$

Then $\vec{y}_1$ is approximately a randomly chosen element of the dual group

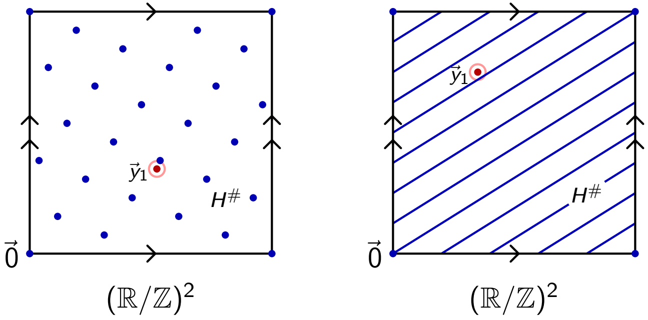
$$H^{\#} = \widehat{\mathbb{Z}^k/H} \le (\mathbb{R}/\mathbb{Z})^k,$$

Explicitly, $H^{\#}$ consists of those $\vec{y}$ such that $\vec{x} \cdot \vec{y} \in \mathbb{Z}$ for all $\vec{x} \in H$.

The sample $\vec{y}_1$ also has noise due to both Gaussian blur and discretization. This noise is exponentially small, but so is the feature scale of $H^{\#}$ when the generators of H are exponentially large.

Examples of $H^\#$

Here are two examples of $H^\#$ and a sample $\vec{y}_1 \underset{\sim}{\in} H^\#$.



On the left, H has full rank and $H^\#$ is a finite group. On the right, H has lower rank and $H^\#$ is a Lie group with fine stripes.

Solving for $H^{\#}$ from random samples

The easy case

Goal: Find $H^{\#} \leq (\mathbb{R}/\mathbb{Z})^k$ from noisy random samples $\vec{y}_1 \underset{\sim}{\in} H^{\#}$.

Shor-Kitaev: If H has full rank and $H^{\#}$ is finite, then we can find rational approximations to the coordinates of $\vec{y}_1$ using the continued fraction algorithm. In this case, $O(\log |H^{\#}|)$ samples are enough to probably generate $H^{\#}$. This includes Shor's case $H = h\mathbb{Z} \leq \mathbb{Z}$, whence $H^{\#} = \frac{1}{h}\mathbb{Z}/h \leq \mathbb{R}/\mathbb{Z}$.

New: If H has rank $\ell < k$, then $\dim H^{\#} = k - \ell$. In this case, any one coordinate of $\vec{y}_1$ is uniformly random in $\mathbb{R}/\mathbb{Z}$. Rational approximation of the coordinates does not work. Happily, a higher-dimensional "continued fraction" algorithm called LLL (Lenstra-Lenstra-Lovasz) does work.

Solving for $H^\#$ from random samples

The hard case

Idea: An ideal random $\vec{y}_0 \in H^\#$ almost surely densely generates the connected subgroup $H_1^\#$, so look for multiples of $\vec{y}_1 \underset{\sim}{\in} H^\#$ near $\vec{0}$.

- Using a single sample $\vec{y}_1$, make a lattice $L \leq \mathbb{R}^{k+1}$ with basis

  $$\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_k, (\widetilde{\vec{y}}_1, 1/T),$$

  where $S \gg T \gg R$, and $1/R$ is the feature scale of $H^\#$.
- Find a LLL basis of short vectors of L:

  $$\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_{k+1} \in L \leq \mathbb{R}^{k+1}$$

  The first $k-\ell+1$ vectors are approx. tangent to $H^\# \oplus \mathbb{R}$ at $\vec{0}$.
- Put the first $k+\ell-1$ LLL vectors in RREF form, then clean them up with rational approximation to find $T_{\vec{0}}(H^\# \oplus \mathbb{R})$ and $H_{\mathbb{R}} = H \otimes \mathbb{R}$. This reduces the problem to Shor-Kitaev.

Last comments and open problems

- The QC difficulty of HSP is a novel property of a discrete group G, which depends on element encoding.
- HSP is probably hard for most infinite groups, but they have a wide variety of behaviors.
- There might be a good quantum algorithm for HSP in nilpotent groups.
- Unary vs binary notation for $\vec{x} \in \mathbb{Z}^k$ is related to canonical words vs canonical compressed words in groups. There is a crazy theorem from the computer science of text editors that compressed words in F_k can be efficiently canonicalized. My NP-hardness might extend to this encoding.
- Efficient algorithms for canonical compressed words are another good question in combinatorial group theory.