Lecture 14

17 MAR 2014

Shor's algorithm is now known to be just one of many that arise in the hidden subgroup framework.

HSP: Let $f: G \to X$ map a group G to some finite set X w/ the property that $\exists$ some subgroup $S \leq G$ such that $\forall x, y \in G$: where $x+S = \{x+s : s \in S\}$

$$f(x) = f(y) \text{ iff } x+S = y+S$$

i.e., f is constant on the cosets of S + distinct on different cosets.

Deutsch–Jozsa + Simon's algorithm fall in this framework + order finding as well ($G = \mathbb{Z}$, $S = r\mathbb{Z}$)

- outline for a q. algorithm to solve the abelian version
- one generalization of Hadamard is $H \otimes H \otimes \cdots \otimes H$

Another is QFT_N for N arbitrarily large

We can perform both kinds of generalizations for HSPs.

$$QFT_N^{\otimes n} = QFT_N \otimes QFT_N \otimes \cdots \otimes QFT_N$$

or even

$$QFT_{N_1} \otimes QFT_{N_2} \otimes \cdots \otimes QFT_{N_k}$$

(operates on a space $H_{N_1} \otimes H_{N_2} \otimes \cdots \otimes H_{N_k}$)

easy to see that

$$QFT_{N_1, N_2, \ldots, N_k} |x_1\rangle |x_2\rangle \cdots |x_k\rangle$$

$$= \sum_{(y_1, y_2, \ldots, y_k) \in \mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2} \times \cdots \times \mathbb{Z}_{N_k}} e^{2\pi i \left(\frac{y_1 x_1}{N_1} + \frac{y_2 x_2}{N_2} + \cdots + \frac{x_k y_k}{N_k} \right)} |y_1\rangle \cdots |y_k\rangle$$

If $N_1 = N_2 = \cdots = N_k$, we can write

$$QFT_N^{\otimes k} |x\rangle = \frac{1}{\sqrt{N^k}} \sum_{y \in \mathbb{Z}_N^k} e^{\frac{2\pi i}{N} x \cdot y} |y\rangle$$

Let S be any subgroup of $\mathbb{Z}_N^k$ & suppose it is possible to prepare a uniform superposition over elements of the subgroup

$$|S\rangle \equiv \frac{1}{\sqrt{|S|}} \sum_{s \in S} |s\rangle$$

Let $S^\perp = \{t : t \cdot s = 0 \quad \forall s \in S\}$

Then $QFT_N^{\otimes k} |S\rangle$

$$= \frac{1}{\sqrt{|S^\perp|}} \sum_{t \in S^\perp} |t\rangle$$

follows b/c elements of $S^\perp$ are the only ones w/ non-zero amplitude. all others have the phases cancel by summing over unit circle. (similar thing in Simon's alg.)

Also, for any $b \in \mathbb{Z}_N^k$

let $b+S = \{b+s : s \in S\}$ &

$$|b+S\rangle = \frac{1}{\sqrt{|S|}} \sum_{s \in S} |b+s\rangle$$

Then $QFT_N^{\otimes k} |b+S\rangle = \frac{1}{\sqrt{|S^\perp|}} \sum_{t \in S^\perp} e^{\frac{2\pi i t \cdot b}{N}} |t\rangle$

More generally, consider any finite abelian group that is isomorphic to $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2} \times \cdots \times \mathbb{Z}_{N_k}$

& for any subgroup $S \subseteq G$

define

$$|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{s \in S} |s\rangle$$

& for any coset $b+S$ of S define

$$|b+S\rangle = \sum_{s \in S} |b+s\rangle$$

Let $S^\perp = \{t : \frac{t_1 s_1}{N_1} + \frac{t_2 s_2}{N_2} + \cdots + \frac{t_k s_k}{N_k} = 0 \bmod 1 \ \forall s \in S\}$

$x = 0 \bmod 1$ if $x \in \mathbb{Z}$ same as $e^{2\pi i \left(\frac{t_1 s_1}{N_1} + \cdots + \frac{t_k s_k}{N_k}\right)} = 1$

So $QFT_G |x\rangle \to \sum_{y \in G} e^{2\pi i \left(\frac{x_1 y_1}{N_1} + \frac{x_2 y_2}{N_2} + \cdots + \frac{x_k y_k}{N_k}\right)} |y\rangle$

+

$QFT_G^{-1} |b+S\rangle = \sum_{t \in S^\perp} e^{2\pi i \left(\frac{t_1 b_1}{N_1} + \cdots + \frac{t_k b_k}{N_k}\right)} |t\rangle$

Now we give an algorithm for solving the HSP over abelian groups.

For simplicity, suppose we can perform QFT exactly for any N

Let $N = p_1^{n_1} p_2^{n_2} \cdots p_\ell^{n_\ell} = N_1 N_2 \cdots N_\ell$ be a prime factorization of N

+ let $n = \sum_j n_j$

1. Set $i = 1$
2. Initialize $|0\rangle|0\rangle \cdots |0\rangle|0\rangle \in$

$H_{N_1} \otimes H_{N_2} \otimes \cdots \otimes H_{N_\ell} \otimes H_X$

3. Apply $QFT_{N_1, N_2, \ldots, N_R}$ to the input to make

$$\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |0\rangle = \frac{1}{\sqrt{|G||S|}} \sum_{y \in G/S} |y+S\rangle |0\rangle$$

4. Apply U_f to make

$$\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |f(x)\rangle$$

5. ~~Measure the second register~~

$$= \frac{1}{\sqrt{|G||S|}} \sum_{y \in G/S} |y+S\rangle |f(y)\rangle$$

5. Measure the 2nd register

After doing so, the state of the 1st register is a random coset state

$$|y+S\rangle$$

6. Apply QFT^{-1} & measure to obtain a (uniformly random) value $t_i \in S^\perp$

Repeat this $n+4$ times so that the samples generate $S^\perp$ w/ prob. $\geq 2/3$

can solve the linear system of eqn's

$$\underline{\underline{T}}\,\underline{x} = \underline{0} \bmod 1$$

where the i^{th} row of T

is $\left(\frac{t_{i,1}}{N_1}, \frac{t_{i,2}}{N_2}, \cdots, \frac{t_{i,k}}{N_k}\right)$.

solve this linear system of eqn's to get the hidden subgroup.
↑ generators for

result: ∃ a BQP algorithm for finding generators for the hidden subgroup S using $O(\log N)$ evaluations of f & $O(\log^3 N)$ other elementary operations.

difficult for non-abelian groups,

for example, for symmetric group would solve graph automorphism problem, which is not known to be easy or hard

Grover's search algorithm

applies to an unstructured search problem

For example, given is a Boolean circuit which outputs 1 for one input & zero for all others. Suppose It is possible to effeciently execute this circuit. Classically, we would have to search through all possibilities (2^n) but quantumly we can reduce this to $\sqrt{2^n}$. (sound familiar? SAT...)

So, given is a black box to compute $f : \{0,1\}^n \to \{0,1\}$ where

$$f(x) = \begin{cases} 1 & \text{if } x = w \\ 0 & \text{otherwise} \end{cases}$$

Classically, if we are allowed just one query, the best we can do is to guess a solution x_1 uniformly @ random & then check if $f(x_1) = 1$. If not, guess x_2 unif @ random & output x_2.

Proc. is correct w/ prob. $\frac{2}{2^n}$

w/ two queries, the best we can do is $\frac{3}{2^n}$

for k queries $(k < 2^n)$, the best we can do is $\frac{k+1}{2^n}$.

Each additional query boosts the success prob. by $\frac{1}{2^n}$

Quantumly, naive algorithm makes a guess w/ prob. $\frac{1}{2^n}$ & thus a prob. amplitude of $\frac{1}{\sqrt{2^n}}$. So if we could boost amplitude by $\frac{1}{\sqrt{2^n}}$ for each query, we would require

only $O(\sqrt{2^n})$ queries.

Grover's algorithm accomplishes this...

Suppose we have a black box to compute:

$$U_f|x\rangle|b\rangle \rightarrow |x\rangle|b \oplus f(x)\rangle$$

(if we have a description of a circuit to make f, then we can use the techniques of reversible computation to realize U_f.)

So then we can do

$$|x\rangle|0\rangle \xrightarrow{U_f} |x\rangle|f(x)\rangle$$

By measuring target, we get the answer to the oracle query to f. But no better than classical approach...

need some quantum magic...

Can instead prepare the 1st register in a superposition

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \qquad \left(\text{where } N = 2^n\right)$$

split the sum into 2 parts: one for which $f(x)=1$ & other for which $f(x)=0$

Define $X_{good} = \{w\}$

$X_{bad} = \{x : f(x)=0\}$

Let $|\psi_{good}\rangle = |w\rangle$

$$|\psi_{bad}\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \in X_{bad}} |x\rangle$$

$$|\psi_n\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle = \frac{1}{\sqrt{N}} |w\rangle + \sqrt{\frac{N-1}{N}} |\psi_{bad}\rangle$$

So

we could ~~[illegible]~~ execute U_f & get

$$\frac{1}{\sqrt{N}} |w\rangle |1\rangle + \sqrt{\frac{N-1}{N}} |\psi_{bad}\rangle |0\rangle$$

but this is no better than classical.

We will need to use phases to do better...

set target register to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

to use phase kickback trick

$$|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} (-1)^{f(x)}|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

can write this simply as

$$|x\rangle \rightarrow (-1)^{f(x)}|x\rangle$$

Define an operator (unitary) which is a reflection about $|t_n\rangle$

$$U_R = \begin{cases} |t_n\rangle \rightarrow |t_n\rangle \\ |t\rangle \rightarrow -|t\rangle \text{ if } |t\rangle \perp |t_n\rangle \end{cases}$$

$$U_R = |t_n\rangle\langle t_n| - (I - |t_n\rangle\langle t_n|)$$

$$= 2|t_n\rangle\langle t_n| - I$$

Algorithm will be

1) prepare $|0\rangle^{\otimes n}$

2) Apply $H^{\otimes n}$ to make $|+_n\rangle$

3) Apply $U_f U_R$ $\left\lfloor \pi/4 \frac{1}{\sqrt{N}} \right\rfloor$ times

4) measure the resulting state..

next time: why does this work?