Lecture 8

17 FEB 2014

We finished off the universality proof, that any n-qubit unitary U can be simulated w/ gates from a discrete gate set, using for example $\{CNOT, H, T\}$, up to accuracy ε w/ no more than

$$O\left(n^2 4^n \log^c\left(\frac{n^2 4^n}{\varepsilon}\right)\right)$$ gates.

| This is b/c the algorithm to decompose ~~an~~ arbitrary ~~n~~-qubit unitary into CNOTs + single qubit unitaries gives a circuit w/ $n^2 4^n$ gates
- overhead from exponential size of matrix U + Gray code construction
- combined w/ overhead from Solovay-Kitaev + the need for each gate to have $\frac{\varepsilon}{L}$ accuracy where L is the # of gates

We would like to argue that there are unitaries acting on n qubits that cannot be approximated w/ polynomial-size quantum circuits.

- We will do so by arguing that there are states that cannot be realized by polynomial size q. circuits.

First, let's figure out how many states we can reach w/ a circuit w/ m gates

(Suppose we have g different kinds of gates & each one acts on no more than f qubits. E.g., {CNOT, H, T} would have $g=3$, $f=2$.)

For the first gate, there are no more than

$$\binom{n}{f}^g = O(n^f)^g = O(n^{fg})$$

possible choices.

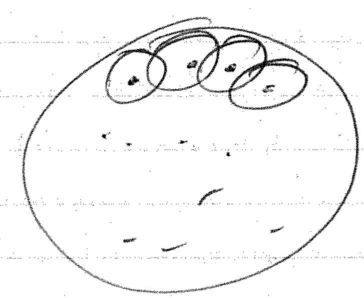
then no more than

$$O(n^{tgm}) \text{ possible states}$$

can be computed w/ m gates.

Now suppose that we wish to approximate a state $|\psi\rangle$ on n qubits up to $\varepsilon > 0$.

We can cover ~~the~~ the space of all n-qubit pure states w/ an ε-net. (discretization of Hilbert space).



Important question: How many states do we require to have an ε-net?
in dimensional

An ε-net for dimension d is defined to be a set of M pure states $\{|\psi_i\rangle\}_{i=1}^{M}$ such that for all $|\psi\rangle \in \mathbb{C}^d$, pure states $\exists\, i \in \{1, \ldots, M\}$ such that

$$\| |\psi\rangle - |\psi_i\rangle \|_2 \leq \varepsilon$$

~~Bo~~ Bound on Size:

~~Formula~~ every state in $d = 2^n$ dimensions has 2^n amplitudes α_j obeying

$$\sum_{j=1}^{2^n} |\alpha_j|^2 = 1$$

$$= \sum_{j=1}^{2^n} (\alpha_j^R)^2 + (\alpha_j^I)^2 = 1$$

↑ equation for a sphere in $2 \cdot 2^n$ dimensions

~~#~~ surface area of ~~such~~ a sphere of radius r in d dimensions is

$$S_d(r) = 2\pi^{d/2} r^{d-1} / \Gamma(d/2)$$

can approximate surface area of radius ε sphere by volume of radius ε in one less dimension

$$V_d(r) = \frac{2\pi^{d/2}\, r^d}{d\,\Gamma(d/2)}$$

so we need a number of patches covering the space to go like

$$\frac{S_{2^{n+1}}(1)}{V_{2^{n+1}-1}(\varepsilon)} = \sqrt{\pi}\,\frac{\Gamma(2^n - 1/2)}{\Gamma(2^n)}\,\frac{[2^{n+1}-1]}{\varepsilon^{2^{n+1}-1}}$$

there is the inequality $\Gamma(2^n - 1/2)$
$\geq \Gamma(2^n)\cdot 2^n$

$\Rightarrow$

$$\text{ratio} \geq \Omega\left(\frac{1}{\varepsilon^{(2^{n+1}-1)}}\right)$$

i.e., # of patches need to cover space grows doubly exponentially fast w/ n
(i.e., # of states is growing this way too)

In order to reach all the states in the ε-net w/ a circuit of m gates, we would require

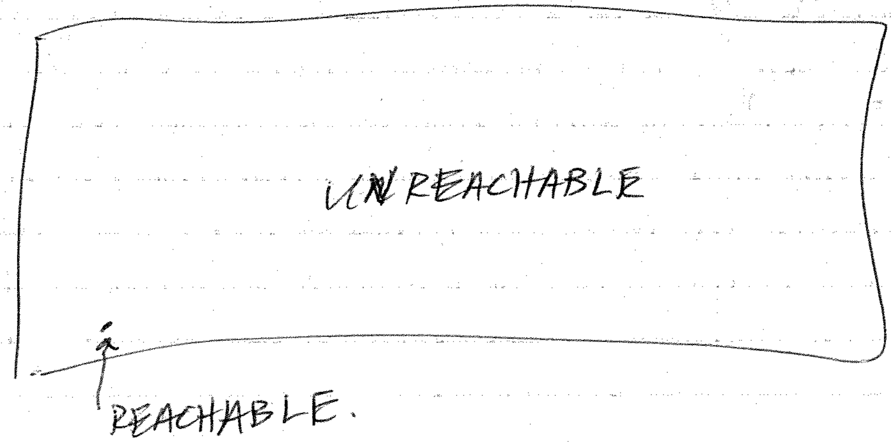
$$O\left(n^{fgm}\right) \geq \Omega\left(\left(\frac{1}{\varepsilon}\right)^{2^{n+1}-1}\right)$$

(i.e., # of states we can reach should be larger than the size of the ε-net.)

$$\Rightarrow \quad m = \Omega\left(\frac{2^n \log\left(\frac{1}{\varepsilon}\right)}{\log(n)}\right)$$

we would need an exponential number of gates.

So these states are difficult to reach & of course there are many of them. Suggests the following picture of Hilbert space



We can now formally define BQP as:

Let $A = A_{yes} \cup A_{no} \subseteq \{0,1\}^*$.

$A \in BQP$ if $\forall x \in A$ $\exists$ a deterministic polynomial-time Turing machine that generates a description of a quantum circuit Q_x acting on $p(n)$ qubits such that

(circuit elements should be from some universal family.)

(completeness) If $x \in A_{yes}$, then

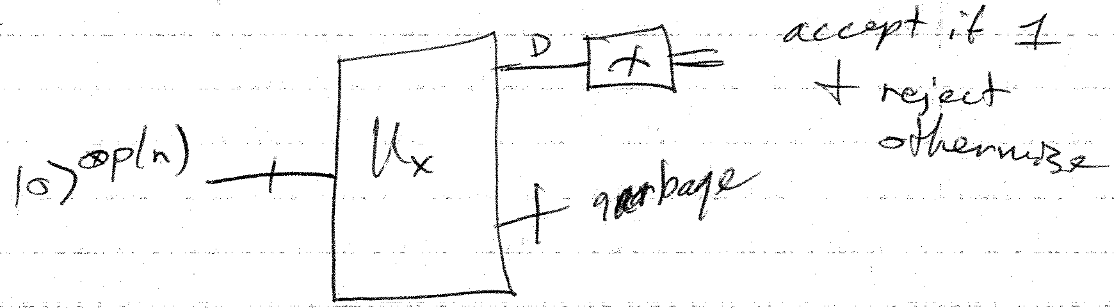
$$\Pr\{Q \text{ accepts } x\} = \mathrm{Tr}\left\{\left[|1\rangle\langle 1| \otimes I^{\otimes p(n)-1}\right] U_x |0\rangle\langle 0|^{\otimes p(n)} U_x^\dagger\right\} \geq 2/3$$

(soundness) If $x \in A_{no}$, then

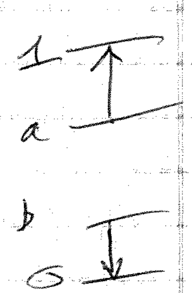
$$\Pr\{Q \text{ rejects } x\} = \mathrm{Tr}\left\{\left[|0\rangle\langle 0| \otimes I^{\otimes p(n)-1}\right] U_x |0\rangle\langle 0|^{\otimes p(n)} U_x^\dagger\right\} \geq 2/3$$

Intuitive picture: $x \to M(x) \to$ output Q_x

then



As in the case of BPP,
we can amplify probabilities to
be exponentially close to
their extremes if



$x \in A_{yes} \iff \Pr\{\text{accept}\} \geq a(|x|)$

$x \in A_{no} \iff \Pr\{\text{reject}\} \leq b(|x|)$

& $\quad a(n) - b(n) \geq \dfrac{1}{\text{poly}(n)}$

class is robust under a wide variety
of error parameters

BQP Subroutine theorem:

What if we want to ~~[illegible]~~ use a BQP algorithm as a subroutine for some other algorithm?

This is commonly done in computer science, + we should understand how to do this properly w/ q. computers,

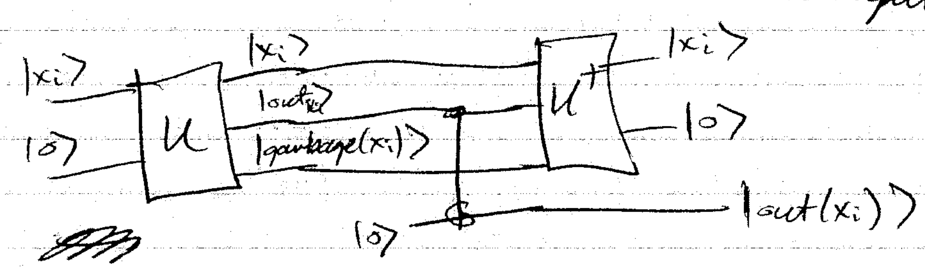
For example, we have circuit Q_{x_1} for input x_1 + circuit Q_{x_2} for input x_2

we can then devise a circuit U such that

$$U|x_i\rangle|0\rangle \to |x_i\rangle|out(Q_{x_i})\rangle|garbage(Q_{x_i})\rangle$$

these are undesirable when we query in superposition.

To eliminate this problem, we "uncompute" i.e.



w/ BQP defined, we can now prove a simple, yet important theorem

$BQP \subseteq PSPACE$

(better than obvious $BQP \subseteq EXP$)

Need to show the existence of a PSPACE algorithm for simulating any promise problem in BQP

Consider that any BQP circuit consists of $U_1, \ldots, U_L$ & acceptance probability is given by

$$\langle 0|^{\otimes p(n)} U_1^\dagger \cdots U_L^\dagger \left[|1\rangle\langle 1| \otimes I^{\otimes [p(n)-1]}\right] U_L \cdots U_1 |0\rangle^{\otimes p(n)}$$

insert identity matrices

$$\Rightarrow \quad \langle 1| \otimes I \left(U_L \cdots U_1\right) |0\rangle^{\otimes p(n)} =$$

$$\langle 1| \otimes I \; U_L \, I \, U_{L-1} \cdots U_2 \, I \, U_1 \, |0\rangle^{\otimes p(n)}$$

So

$$= U_L^{\dagger}\sum_{y_{L+1}}|y_{L+1}\rangle\langle y_{L+1}|\,|1\rangle\langle 1|\sum_{y_L}|y_L\rangle\langle y_L|U_L\sum_{y_{L-1}}|y_{L-1}\rangle\langle y_{L-1}|U_{L-1}\cdots U_2\sum_{y_1}|y_1\rangle\langle y_1|U_1|0\rangle$$

$$=\sum_{y_1,\ldots,y_L,\ldots,y_{2L}}\langle 0|U_1^{\dagger}|y_{2L}\rangle\langle y_{2L}|U_2^{\dagger}|y_{2L-1}\rangle\cdots$$
$$\langle y_{L+1}|U_L^{\dagger}|1\rangle|y_L\rangle\ \langle 1|\langle y_L|U_L|y_{L-1}\rangle\ \langle y_{L-1}|U_{L-1}|y_{L-2}\rangle$$
$$\cdots\ \langle y_2|U_2|y_1\rangle\langle y_1|U_1|0\rangle$$

each U_i acts on just one or two qubits, so from the description of the circuit, it is easy to compute entries like

$\langle y_j|U_j|y_{j-1}\rangle$ in polynomial time

+ for each path we can store the result in polynomial space.

Since we can erase calculated terms after adding them to the running totals, we can use only poly-space + have enough precision to decide whether to accept or reject

same idea as Feynman path integral