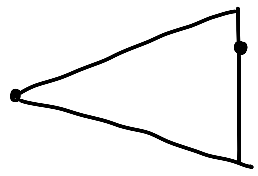
LECTURE 5 (January 31)

RECAP Theorem 3 $BQP \subseteq PP$ where $PP =$ probabilistic poly-time algorithms w/ error $< \frac{1}{2}$

We converted a circuit that takes n-bit inputs and uses h Hadamard gates into a tree with 2^h leaves



$\pm \frac{1}{\sqrt{2^h}} |y\rangle$ for $y \in \{0,1\}^{poly(n)}$ at each leaf

The final state $|\psi\rangle = \frac{1}{\sqrt{2^h}} \sum_p \text{sign}(p) |\text{label}(p)\rangle$

$$= \frac{1}{\sqrt{2^h}} \sum_{y \in \{0,1\}^{poly(n)}} \underbrace{\left(\sum_{\substack{\text{paths } p \\ \text{with label } y}} \text{sign}(p) \right)}_{:= \alpha_y} |y\rangle$$

Then, $\alpha_y^2 = \frac{1}{2^h} \sum_{\substack{p, p' \\ \text{label}(p) = \text{label}(p') = y}} \text{sign}(p) \cdot \text{sign}(p')$

$\mathbb{P}$[BQP algorithm outputs 1] $-$ $\mathbb{P}$[BQP algorithm outputs 0]

$$= \sum_y |\alpha_y|^2 (-1)^{\mathbb{1}[y_1 = 0]}$$

$$= \frac{1}{2^h} \sum_{\substack{p, p' \\ \text{w/ same} \\ \text{labels}}} \underbrace{\text{sign}(p) \cdot \text{sign}(p') (-1)^{\mathbb{1}[(\text{label}(p))_1 = 0]}}_{\beta(p,p')} = \begin{cases} > \frac{1}{3} & \text{if correct answer is } 1 \\ < -\frac{1}{3} & \text{if correct answer is } 0 \end{cases}$$

PP algorithm Randomly select two paths p, p' in the tree

- If labels are different, just accept / reject w.p. $1/2$
- If labels are same,

$$\text{accept iff } \underbrace{\text{sign}(p) \cdot \text{sign}(p') (-1)^{\mathbb{1}[(\text{label}(p))_1 = 0]}}_{:= \beta(p,p')} > 0$$

Time $=$ poly(n)

$$\mathbb{P}[\text{Accept}] - \mathbb{P}[\text{Reject}] = \frac{1}{2^{2h}} \sum_{\substack{p, p' \\ \text{label}(p) = \text{label}(p')}} \beta(p,p') = \begin{cases} \geq \frac{1}{2^h \cdot 3} & \text{if correct answer is } 1 \\ \leq -\frac{1}{2^h 3} & \text{if correct answer is } 0 \end{cases}$$

$\square$

Theme for the next few lectures

Quantum Advantage
- How to establish it?
- How to show that there is no advantage?
- What sort of structure is needed for quantum advantage?

TODAY

Intro to oracle separations & query complexity

BQP vs BPP

How do we show that BQP is more powerful than BPP?

or Other classical complexity classes?

or that BQP can not solve NP-hard problems?

- Proving unconditional separations is out of reach
- Proving separations based on standard assumptions such as $P \neq NP$, cryptographic assumptions is also extremely difficult
- Oracle or Black-box separations — still very difficult in many cases

Show that $\exists$ oracle O s.t. $BQP^O \neq BPP^O$

Disclaimer: Oracles increase the computational power of classes differently. So, this is only a heuristic

For instance, $\exists$ complexity classes A, B s.t. $A = B$ but $\exists$ oracle O s.t. $A^O \neq B^O$

So, what is the point?

Showing black-box separations reduces to unconditional separations for query algorithms

- Understanding algorithms in this simplified model gives useful algorithmic ideas and can also lead to a candidate for quantum advantage in the real-world
- If separations don't hold in the simplified model, it gives indications that new principles are needed to establish unconditional separations

Query Algorithms

Suppose we have an oracle O_f that computes $f: \{0,1\}^n \to \{0,1\}$

To use this oracle as part of the quantum circuit we must define a unitary U_f that implements calls to the oracle

Moreover, ideally if we have an efficient quantum circuit for f, we should be able convert it into an efficient quantum circuit for U_f

There are two standard ways of doing this

Bit oracle $U_f|x,b\rangle = |x, b\oplus f(x)\rangle$ — This is what we used before

Note that $U_f^{\dagger} = U_f$

Phase oracle $V_f|x\rangle = (-1)^{f(x)}|x\rangle$ — Value of f is returned in the phase

Also, satisfies $V_f^{\dagger} = V_f$

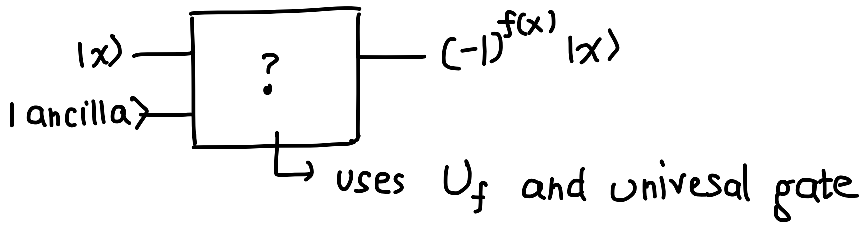
Note that although global phases don't matter if V_f is applied to a superposition it can create relative phases

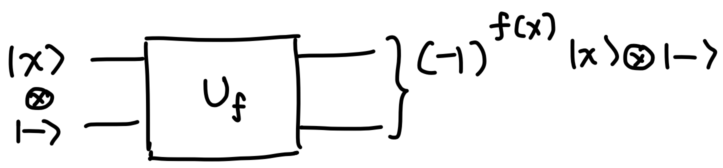
For example, if $f:\{0,1\} \to \{0,1\}$ is defined as $f(x) = x$

Then, $V_f\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$

The two oracles are equivalent in the sense that given access to one (and some ancillas) the other can also be implemented efficiently

Bit oracle to phase oracle





$$U_f\frac{|x\rangle|0\rangle - |x\rangle|1\rangle}{\sqrt{2}} = \frac{|x\rangle|f(x)\rangle - |x\rangle|1\oplus f(x)\rangle}{\sqrt{2}}$$

$$= \begin{cases} |x\rangle|-\rangle & \text{if } f(x) = 0 \\ -|x\rangle|-\rangle & \text{if } f(x) = 1 \end{cases}$$

This is called the phase kickback trick

Phase Oracle to bit oracle Exercise

Query Complexity Given black-box or query access to U_f or V_f, how many queries need to be made to the black-box in order to solve a problem ?

If classically, we need exponentially many queries but quantumly, only polynomially many

This is an evidence of quantum algorithms having an exponential advantage in terms of query complexity for a problem

Moreover, query complexity separation $\Rightarrow$ oracle separation

And if we can find an efficient circuit for the black-box or a new quantum algorithmic technique, this can give rise to real-world problems with practical quantum advantage

For instance, Simon's problem shows an exponential separation in terms of classical versus quantum query complexity and also inspired Shor's algorithm

Simon's problem Given a black-box $f: \{0,1\}^n \to \{0,1\}^n$ promised that either

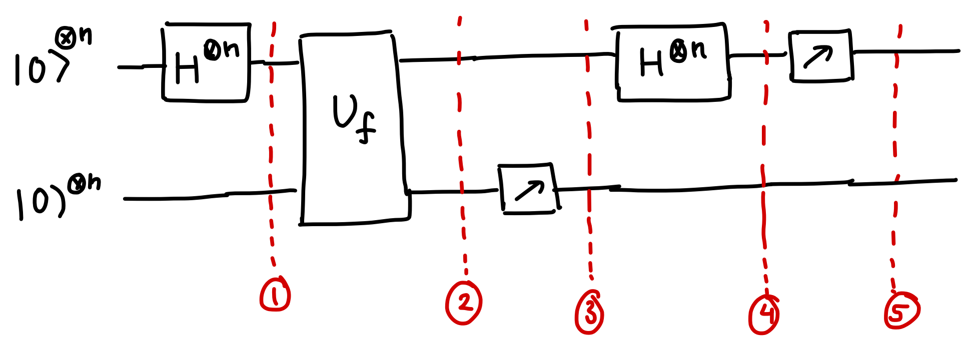
- f is 1-to-1
- OR $\exists$ an unknown string $s \neq 0$ s.t. $\forall\, x \neq y$, $f(x) = f(y)$ iff $y = x \oplus s$

Figure out which case we are in

If we think of the hypercube $\{0,1\}^n$ as being colored by the corresponding f-values, the first condition means all colors are distinct, the second means all pairs $(x, x\oplus s)$ are colored with distinct color but the color within a pair is the same

Theorem (Simon) (a) $\exists$ a quantum algorithm solving the problem with $O(n)$ queries with constant error
(b) any classical algorithm requires $\Theta(2^{n/2})$ queries for constant error

Quantum Algorithm The algorithm runs the following quantum circuit $O(n)$ times and does some classical post-processing afterwards



What does this circuit do? for f s.t. $f(x) = f(x\oplus s)$

① $= |+\rangle^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0\rangle^{\otimes n}$ Prepare uniform superposition over inputs

Query f ② $= \frac{1}{\sqrt{2^n}} \sum_x |x\rangle|f(x)\rangle$

Measure ③ $= \frac{1}{\sqrt{2}} (|x^*\rangle + |x^*\oplus s\rangle) \otimes |f(x^*)\rangle$ for some random x^*

Discard second register

Are we done here? Can we measure this state $O(1)$ times and obtain s? Why NOT?

Hadamard on first n qubits

$$④ = \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} \left((-1)^{x^*\cdot y} + (-1)^{(x^* \oplus s)\cdot y} \right) |y\rangle$$

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} (-1)^{x^*\cdot y} \left(1 + (-1)^{s\cdot y}\right) |y\rangle$$

$$= \frac{1}{\sqrt{2^{n-1}}} \sum_{\substack{y \text{ s.t.} \\ s\cdot y = 0}} (-1)^{x^*\cdot y} |y\rangle$$

Measure first n qubits

⑤ = Get a uniformly random y s.t. $s \cdot y = 0$

$\Rightarrow s_1 y_1 \oplus \cdots \oplus s_n y_n = 0$

This is a random linear equation over $\mathbb{F}_2$ in the variables $s_1, \ldots, s_n \in \mathbb{F}_2$

If we have $n-1$ linearly independent equations $s \cdot y_i = 0$ we can solve for s $\Big\} \Rightarrow$ 2 solutions $s = 0$ & $s \neq 0$

$\Rightarrow$ If we run this process $O(n)$ times, we get n linearly independent linear equations with constant probability

$\Rightarrow$ If $\exists$ non-zero solution s, output that f is not 1-1

otherwise, output that f is 1-1

(Exercise) Make the above rigorous by showing that if we run the above quantum circuit $m = O(n)$ times and obtain $y_1, \ldots, y_m$

Then, with high probability the system of linear equation over $\mathbb{F}_2$

$$s \cdot y_1 = 0$$
$$\vdots$$
$$s \cdot y_m = 0$$

- has a non-zero solution if f is not 1-to-1
- only has a zero solution if f is 1-to-1

Classical Lower Bound Every randomized algorithm needs $\Omega(2^{n/2})$ queries

Recipe ① Come up with a hard candidate distribution on inputs

$$f = \begin{cases} \text{uniformly random 1-1 function w.p. } 1/2 \\ \text{uniformly random function satisfying Simon' property w.p. } 1/2 \end{cases}$$

② Suffices to consider deterministic algorithms

$$\mathbb{E}_f \; \mathbb{E}_r \left[\mathbb{1}\left[\text{Algorithm with randomness } r \text{ succeeds} \right] \right] \geq \frac{2}{3}$$

$$\Rightarrow \mathbb{E}_r \; \mathbb{E}_f \left[\mathbb{1}\left[\text{———} \;\; '' \;\; \text{———} \right] \right] \geq \frac{2}{3}$$

$$\Rightarrow \exists r \text{ s.t. } \mathbb{E}_f \left[\mathbb{1}\left[\text{Algorithm with randomness } r \text{ succeeds} \right] \right] \geq \frac{2}{3}$$

③ Lower Bound for Deterministic algorithms

First query $(\underset{\text{input}}{x_1}, \underset{\text{output}}{y_1})$ $\longrightarrow$ No information about s since $y_1 \in \{0,1\}^n$ is uniform

Second query (x_2, y_2) $\longrightarrow$ Either $x_1 \oplus x_2 = s$ (collision) $\longrightarrow$ w.p. $\approx 2^{-n}$

or $x_1 \oplus x_2 \neq s$ (no collision)

$\downarrow$

can't distinguish 1-1 inputs

k queries $(x_1, y_1), \ldots, (x_k, y_k)$

Can't distinguish unless $x_i \oplus x_j = r$ for some pair (i,j)

➡ $\mathbb{P}$[any collision among k queries] $= O\left(\frac{k^2}{2^n - k^2}\right)$ where $k \leq \frac{2^{n/2}}{100}$