Lecture 4

We have now seen that $R(\mathrm{OR}_n^{0,1}) \geq n/3$ but $Q(\mathrm{OR}_n^{0,1}) \leq \frac{\pi}{4}\sqrt{n} + \frac{1}{2}$, which completes our first rigorous proof of a (quadratic) quantum speedup in terms of n within the query model.

In this lecture, we'll see two very useful principles of quantum algorithm design given as the two items in fact 1 below. We will apply these two principles to show how the quantum query complexity of OR_n (without any restriction on domain) is also $O(\sqrt{n})$. In later lecture, we will take these principles for granted and not explcitly mention them.

Fact 1. 1. Quantum (query) algorithms can efficiently simulate randomized (query) algorithms. In particular $Q(f) \leq R(f)$ for any f. Reference: Section 2.3.3 of de Wolf's PhD thesis.

Proof sketch. We will see how a quantum query algorithm can simulate a DDT first by way of an example: consider the obvious depth-2 DDT T that computes $(\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_3)$ with 1 labelling the root.

We will use the following

Fact (*). Suppose $g\colon \{0, 1 \ldots, a-1\} \to \{0, 1, \ldots, b-1\}$, then there exists a unitary U_f (in fact permutation matrix) acting on the space $\mathbb{C}^a \otimes \mathbb{C}^b = \mathbb{C}^{ab}$ ($U_f \in \mathbb{C}^{ab \times ab}$) such that

$$U_f |x\rangle |0\rangle = |x\rangle |f(x)\rangle \tag{51}$$

for all $x \in \{0, 1, \ldots, a-1\}$.

Let $I \coloneqq \{0,1\} \to \{2,3\}$ be defined by $I(0) = 2$ and $I(1) = 3$. (I maps the bit value of x_1 to the index that is read next.) Let $I - 1$ denote the function that first applies I and then subtracts 1. Let $h\colon \{0,1\} \times \{0,1,2\} \times \{0,1\}$ by

$$h(0, 2-1, 0) = 0,\ h(0, 2-1, 1) = 1,\ h(1, 3-1, 0) = 1,\ h(1, 3-1, 1) = 0. \tag{52}$$

We have defined h such that $h(a, I-1, b)$ is defined to be the value that T outputs if $x_1 = a$, I is the index of the variable queried next, and $x_I = b$.

Register dimensions $\mathbb{C}^3 \otimes \mathbb{C}^2 \otimes \mathbb{C}^3 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$:

$$\begin{aligned}
&\underbrace{|0\rangle |0\rangle}_{\text{query registers}} \quad \underbrace{|0\rangle |0\rangle |0\rangle}_{\text{workspace registers}} \\
&\overset{O_x}{\mapsto} |0\rangle |x_1\rangle |0\rangle |0\rangle |0\rangle \\
&\overset{U_{I-1}}{\mapsto} |0\rangle |x_1\rangle |I(x_1) - 1\rangle |0\rangle |0\rangle && \text{notation follows fact (*)} \\
&\overset{O_x}{\mapsto} |0\rangle |x_1\rangle |I(x_1) - 1\rangle |x_{I(x_1)}\rangle |0\rangle \\
&\overset{U_h}{\mapsto} |0\rangle |x_1\rangle |I(x_1) - 1\rangle |x_{I(x_1)}\rangle |h(x_1, I(x_1) - 1, x_{I(x_1)})\rangle && \text{notation follows fact (*)} \\
&= |0\rangle |x_1\rangle |I(x_1) - 1\rangle |x_{I(x_1)}\rangle |h(x_1, I(x_1) - 1, x_{I(x_1)})\rangle |T(x)\rangle
\end{aligned}$$

where the $\overset{A}{\mapsto}$ notation means application of matrix A (suitably tensored with identity matrices), and the last line uses the definition of h. Then measuring using $\{\Pi_0 \coloneqq \mathbb{1}_{36} \otimes |0\rangle\langle 0|, \Pi_1 \coloneqq \mathbb{1}_{36} \otimes |1\rangle\langle 1|\}$ gives outcome $T(x)$ (with probability 1).

What about RDTs? Recall an RDT is a distribution $(p_i, T_i)_{i=0}^{K-1}$ over DDTs. We have seen how T_i can be simulated by a quantum query algorithm $\mathcal{A}_i$ for each i. Suppose $\mathcal{A}_i$ is specified by unitaries $\{U_j^i\}_{j=0,\ldots,d}$. Then the RDT can be simulated by a quantum query algorithm $\mathcal{A}$ that starts with the state

$$|\psi_0\rangle \coloneqq \sum_{i=0}^{K-1} U_0^i |0\rangle \otimes \sqrt{p_i} |i\rangle . \tag{53}$$

(More precisely, we can define the U_0 of $\mathcal{A}$ to be any unitary such that $U_0 |0\rangle = |\psi_0\rangle$.) Then for $j \in \{1, \ldots, d\}$, U_j of $\mathcal{A}$ is defined to be

$$U_j \coloneqq \sum_{i=0}^{K-1} U_j^i \otimes |i\rangle\langle i|. \tag{54}$$

□

The measurement of $\mathcal{A}$ is still $\{\Pi_0 \coloneqq |0\rangle\langle 0|, \Pi_1 \coloneqq |0\rangle\langle 0|\}$ (tensored with identities so that the Π_bs only act non-trivially on the single register that contains $\{T_i(x) \mid i \in \{0, \ldots, K-1\}\}$.

2. Principle of deferred measurement.

   In our definition of quantum query complexity, there is one measurement coming at the end. But in fact, could have also allowed "intermediate measurements". The principle of deferred measurement says that such measurements can always be simulated by a measurement at the end.

   Proof of deferred measurement. Suppose we make a measurement $\mathcal{M} \coloneqq \{\Pi_1, \dots, \Pi_k\}$ on a state $|\psi\rangle$ and if the measurement outcome is $i \in [k]$, we apply unitary U_i to another state $|\psi'\rangle$. Comment: In Simon's problem (later), need $|\psi'\rangle$ to be the postmeasurement state of $|\psi\rangle$—but the proof is the same. Then the effect of this procedure is that with probability $\|\Pi_i |\psi\rangle\|^2$, we end up with final state $U_i |\psi'\rangle$.

   Now consider the following simulation: we apply the unitary

$$U \coloneqq \sum_{i=1}^{n} \Pi_i \otimes U_i \tag{55}$$

   to the state $|\psi\rangle |\psi'\rangle$ and *then* measure the first register using $\mathcal{M}$. (Note U is unitary: $UU^\dagger = \sum_{i=1}^n \Pi_i \otimes U_i \cdot \sum_{j=1}^n \Pi_j \otimes U_j^\dagger = \sum_{i=1}^n \Pi_i \otimes I = I$; Likewise $U^\dagger U = I$.)
   Then the probability of observing outcome $i \in [k]$ is

$$\|(\Pi_i \otimes \mathbb{1})U |\psi\rangle |\psi'\rangle\|^2 = \|\Pi_i |\psi\rangle \otimes U_i |\psi'\rangle\|^2 = \|\Pi_i |\psi\rangle\|^2, \tag{56}$$

   where the last equality uses the fact that $\|u \otimes v\| = \|u\|\|v\|$ and $\|Vu\| = \|V\|$ for unitary V. And the state on the second register becomes $U_i |\psi'\rangle$. This is precisely the same effect as the original procedure where the measurement comes first. □

Using these facts (implicitly), can show the following.

Proposition 4. *There exists $c > 0$ such that for all $n \in \mathbb{N}$, we have*

$$Q(\mathrm{OR}_n) \le c\sqrt{n}. \tag{57}$$

Proof sketch. First, we may assume that $|x| \le 0.01n$. Else, if we randomly query 10000 indices of x, we'll not find a 1 (i.e., fail to distinguish the input from 0^n) with probability at most

$$\left(1 - \frac{0.01n}{n}\right)^{10000} \le e^{-100} = \text{negligible}^3 \tag{58}$$

where the inequality uses $1 - x \le e^{-x}$ for all $x \ge 0$.

From the analysis before, we see that, on input $x \in \{0,1\}^n$ using k queries we can get the probability of outputting 0 to be

$$p_x(k) = \cos^2(2\theta_x k) = \frac{1 + \cos(4\theta_x k)}{2}, \tag{59}$$

where $\theta_x = \arcsin(\sqrt{|x|/n})$. Plot the graph of $p_x(k)$as a function of k; note that its period T_x satisfies

$$15 \le \frac{\pi}{2 \arcsin\sqrt{0.01}} \le T_x \coloneqq \frac{\pi}{2\theta_x} \le \frac{\pi}{2}\sqrt{n}, \tag{60}$$

where the first second inequality uses the fact that $|x| \le 0.01n$ and the last inequality uses $|x| \ge 1$ (together with the monotonicity of $\arcsin(a)$ for $a \in [0,1]$ and $\arcsin(a) \ge a$ for $a \in [0,1]$).

Therefore, in the interval $[1, \lceil \frac{\pi}{2}\sqrt{n} \rceil]$, $p_x(k)$ runs over at least one period and each period must span over at least 15 positive integers (by the first inequality of eq. (60)).

The last step of the algorithm is:

> Repeat the following 10000 times: choose $k \in \mathbb{N}$ uniformly at random between 1 and $2\sqrt{n}$, run Grover's quantum query algorithm which has $p_k(x)$ probability of outputting 0 (i.e., the measurement outcome being 0). If the output is 1, return 1.
>
> If all repeats give output 0, return 0.

[3] Note that this is "negligible" since we only care about computing OR_n with bounded error 1/3 and *compared to 1/3*, $1 - e^{-100}$ is negligible. To argue this formally, we need to consider the probabilities of failure from all sources (there's another source later on), add them together (cf. the "union bound" or "Boole's inequality" on Wikipedia) and show that the sum is $\le 1/3$.

The intuition for why this works is that if I choose an integer k uniformly at random from $[1, \lceil \frac{\pi}{2}\sqrt{n} \rceil]$, eq. (60) shows that $p_x(k)$ will be constant away from 1 with constant probability (over the randomness of the choice of k). (Think pictorially.)

This means that the quantum query algorithm will output 1 with constant probability. (Recall $p_x(k)$ is the probability of the quantum algorithm outputting 0.) Since we would never see 1 when $x = 0^n$, we can just repeat this a large number of times and output 1 if and only if the quantum query algorithm outputs a 1 in any of those repeats. This allows us to suppress the error probability to be negligible.[4] □

Remark 4. *1. To see that the query algorithm described in the proof is a bonafide quantum query algorithm according to our definition, we need to use* both *facts that we established earlie, i.e., quantum can simulate randomized and principle of deferred measurement. The first fact allows us to convert the randomized query algorithm doing the preprocessing to a quantum query algorithm. But this quantum query algorithm could continue running if its output is not* 1*, and recall a quantum query algorithm's output always arises from a measurement. However, by the second fact, we can defer this measurement to the end. The second fact also allows us to defer the measurements made in each of the repeat loops to the end.*

2. The exposition here expands a little on Scott Aaronson's lecture notes on Grover search (top of page 8).

3. A somewhat different algorithm, along the lines of what Nick suggested in class of exponentially increasing k from 1 *to $O(\sqrt{n})$, is analyzed in detail in Section 4 of this paper.*

4. In fact, there's yet another algorithm for computing OR_n using a "fully quantum strategy" (i.e., very unlike the two algorithms mentioned above that are essentially Grover + classical ideas) called "fixed-point amplitude amplification". See this paper. Maybe we'll have time to discuss this when we talk about quantum signal processing.

Proposition 5 (Error suppression/Chernoff bound)**.** *Let $\epsilon \in (0, 1/3)$. Let $f\colon D \subseteq \{0, 1, \ldots, m-1\}^n \to \Gamma$. Then $R_\epsilon(f) \leq R(f)\lceil 18\ln(1/\epsilon)\rceil$ and $Q_\epsilon(f) \leq Q(f)\lceil 18\ln(1/\epsilon)\rceil$.*

Proof. Will prove the randomized case. Same idea also works in the quantum case since we can simulate randomized by quantum.

Suppose $\mathcal{T}$ is an RDT that computes f with bounded error 1/3. Take $k \in \mathbb{N}$ copies of $\mathcal{T}$ and output the modal output of the k copies. For a given $x \in D$, let X denote the number of copies that output the correct answer on x, the probability that each copy outputs the correct answer is $p = \frac{1}{2} + \delta$, where $\delta \geq 1/6$ and the probability that each copy outputs the incorrect answer is $q = 1 - p = \frac{1}{2} - \delta \leq 1/3$. Correct $\iff X > k/2$. So probability of incorrect is

$$\begin{aligned}
\Pr[X \leq k/2] =& \sum_{i=0}^{k/2} \Pr[X = i] = \sum_{i=0}^{k/2} \binom{k}{i} p^i q^{k-i} \\
\leq& \sum_{i=0}^{k/2} \binom{k}{i} p^{k/2} q^{k/2} \leq 2^k (pq)^{k/2} \\
=& 2^k \Big(\frac{1}{2} + \delta\Big)^{k/2} \Big(\frac{1}{2} - \delta\Big)^{k/2} = 2^k \Big(\frac{1}{4} - \delta^2\Big)^{k/2} \\
=& (1 - 4\delta^2)^{k/2} \leq e^{-2k\delta^2} \qquad \forall x \geq 0,\ 1 - x \leq e^{-x}.
\end{aligned}$$

So if we pick $k \geq \ln(1/\epsilon)/(2\delta^2)$, we have $\Pr[X \leq k/2] \leq \epsilon$. Since $\delta \geq 1/6$, it suffices to pick $k \geq 18\ln(1/\epsilon)$. Hence the proposition. □

Remark 5. *We have shown that given k i.i.d. random variables $X_1, \ldots, X_k$ taking values in $\{0, 1\}$ such that $\exists \delta \in [0, 1/2]$, $\forall i, \Pr[X_i = 1] = \frac{1}{2} + \delta$. Then $\Pr[\sum_{i=1}^k X_i \leq k/2] \leq e^{-2k\delta^2}$. This type of bound is known as a Chernoff bound, there are more sophisticated variants with more sophisticated proofs. The rough-and-ready proof given here is taken from Nielsen and Chuang, Box 3.4.*

[4] Same comment here as the first negligible.