July 24, 2023

Quantum Query Complexity Problem Session 1

Instructor: Yassine Hamoudi — Teaching Assistant: Angelos Pelecanos

Problem 1

Question 1

We will show the first statement: $R_\epsilon(f) \leq O(R(f)\log(1/\epsilon))$. The second statement follows using the same argument.

First note that the definition of $R(f)$ that we have seen in lecture corresponds to $R_{1/3}(f)$.

To prove the statement, consider a randomized algorithm $\mathcal{A}$ that makes $R(f)$ queries and is incorrect with probability at most $\frac{1}{3}$ for every input x. We will construct a randomized algorithm $\mathcal{B}$ that makes $O(R(f)\log(1/\epsilon))$ queries and makes an error with probability at most ϵ.

Our algorithm $\mathcal{B}$ is quite simple: Run $\mathcal{A}$ for k times independently, obtaining k binary values $y_1, \ldots, y_k$. Then output the majority of these values.

What is the probability that $\mathcal{B}$ makes an error? We know that

$$\Pr[y_i \neq f(x)] = p \leq \frac{1}{3}.$$

Thus for $\mathcal{B}$ to make an error, we want at least $\frac{k}{2}$ of the y_i's to be incorrect. This happens with probability at most

$$p^{k/2}(1-p)^{k/2} \cdot \binom{k}{k/2} \leq \frac{1}{3^{k/2}} \cdot \frac{2^{k/2}}{3^{k/2}} \cdot 2^k = \left(\frac{8}{9}\right)^{k/2}.$$

Thus choosing $k = O(\log 1/\epsilon)$ makes the probability that $\mathcal{B}$ makes an error at most ϵ. Since $\mathcal{B}$ runs $\mathcal{A}$ $O(\log 1/\epsilon)$ times, the total number of queries is at most $O(R(f)\log 1/\epsilon)$.

Question 2

Everything from the quantum query model transitions seamlessly to inputs $x \in \{0, 1, \ldots, m-1\}^n$ if we use qudits. In particular, we will need to modify the oracle gate to

$$\mathcal{O}_x |i, b\rangle = \left| i, \underbrace{b + x_i \bmod m}_{\{0,\ldots,m-1\}} \right\rangle$$

where $b \in \{0, \ldots, m-1\}$.

Problem 2

Question 1

We will show how to simulate the *phase query* operator using the query operator and Hadamard transforms as follows:

$$\mathcal{O}_x^{\pm} = (I \otimes H)\mathcal{O}_x(I \otimes H).$$

Recall that

$$\mathcal{O}_x |i, b\rangle = |i, b \oplus x_i\rangle, \quad \mathcal{O}_x^{\pm} |i, b\rangle = (-1)^{b \cdot x_i} |i, b\rangle.$$

$$\begin{aligned}
(I \otimes H)\mathcal{O}_x(I \otimes H) |i, b\rangle &= (I \otimes H)\mathcal{O}_x \frac{|i, 0\rangle + (-1)^b |i, 1\rangle}{\sqrt{2}} \\
&= (I \otimes H) \frac{|i, x_i\rangle + (-1)^b |i, x_i \oplus 1\rangle}{\sqrt{2}} \\
&= \frac{|i, 0\rangle + (-1)^{x_i} |i, 1\rangle}{2} + (-1)^b \frac{|i, 0\rangle + (-1)^{x_i \oplus 1} |i, 1\rangle}{2} \\
&= \frac{1 + (-1)^b}{2} |i, 0\rangle + \frac{(-1)^{x_i}(1 + (-1)^{b \oplus 1})}{2} |i, 1\rangle.
\end{aligned}$$

It is now easy to verify that

$$(I \otimes H)\mathcal{O}_x(I \otimes H) |i, b\rangle = \begin{cases} |i, 0\rangle & b = 0 \\ (-1)^{x_i} |i, 1\rangle & b = 1 \end{cases} = \mathcal{O}_x^{\pm} |i, b\rangle$$

Question 2

For ease of notation, we will denote the two inputs as x_0, x_1. Since we are only allowed one query, we will need to query both indices in superposition. Thus it makes sense to prepare the following state

$$\mathcal{O}_x^{\pm} \frac{|0, 1\rangle + |1, 1\rangle}{\sqrt{2}} = \frac{(-1)^{x_0} |0, 1\rangle + (-1)^{x_1} |1, 1\rangle}{\sqrt{2}}.$$

Now we want somehow the amplitudes to interfere, thus we will apply the Hadamard transform on the first register:

$$
\begin{aligned}
&(H \otimes I)\ \frac{(-1)^{x_0}|0,1\rangle + (-1)^{x_1}|1,1\rangle}{\sqrt{2}} \\
&= \frac{(-1)^{x_0}|0,1\rangle + (-1)^{x_0}|1,1\rangle}{2} + \frac{(-1)^{x_1}|0,1\rangle - (-1)^{x_1}|1,1\rangle}{2} \\
&= \frac{(-1)^{x_0} + (-1)^{x_1}}{2}|0,1\rangle + \frac{(-1)^{x_0} - (-1)^{x_1}}{2}|1,1\rangle \\
&= \begin{cases} (-1)^{x_0}|0,1\rangle & \text{if } x_0 \oplus x_1 = 0 \\ (-1)^{x_0}|1,1\rangle & \text{if } x_0 \oplus x_1 = 1 \end{cases}
\end{aligned}
$$

Thus we measure the first register, and the value we observe equals the value of $x_0 \oplus x_1$.

This algorithm trivially implies a $\frac{n}{2}$-query quantum algorithm to solve Parity: Split the input into $\frac{n}{2}$ pairs and use the 1-query quantum algorithm on each pair. The Parity is equal to the boolean sum of the parities of each pair.

Problem 3

Question 1

- $bs(\text{OR}) = n$. Choose subset $B_j = \{j\}$ for all $j \in [n]$. Then for $x = 0^n$, it holds that
$$\text{OR}(x^{B_j}) \neq \text{OR}(x).$$

- $bs(\text{AND}) = n$. Similar to the previous case, we choose subset $B_j = \{j\}$ for all $j \in [n]$. Then for $x = 1^n$, it holds that
$$\text{AND}(x^{B_j}) \neq \text{AND}(x).$$

- $bs(\textsc{Parity}) = n$. Again, choose each subset $B_j = \{j\}$ for all $j \in [n]$. Then for any $x \in \{0,1\}^n$, it holds that
$$\textsc{Parity}(x^{B_j}) \neq \textsc{Parity}(x),$$
since flipping any bit changes the parity of the input.

- $bs(\textsc{Majority}) = \lceil \frac{n}{2} \rceil$, for odd n. Consider $x = 0^{\lfloor n/2 \rfloor} 1^{\lceil n/2 \rceil}$. Choose each subset $B_j = \{j\}$ for all $j \in \{\lfloor n/2 \rfloor, \ldots, n-1\}$. Then, it holds that
$$\textsc{Majority}(x^{B_j}) \neq \textsc{Majority}(x).$$
It is easy to see that this is the largest number of disjoint subsets. If we choose an input with $2d+1 > 1$ 1's than 0's, then each B_j must satisfy $|B_j| \geq d+1$, which means that we will have at most $\frac{n}{d+1} \leq \frac{n}{2}$ such subsets.

Question 2

For simplicity, let $k = bs(f)$, let x be an input on which f attains its block sensitivity, and consider the respective disjoint subsets $B_1, \ldots, B_k$. For any randomized classical algorithm that makes $o(k)$ queries, there exists at least one subset B_j that has not been queried on any $i \in B_j$.

Thus the classical algorithm will not be able to distinguish x from x^{B_j}.

Question 3

In this question, we will generalize from the previous classical case. Note that the argument in the classical case is that a classical algorithm must query at least one coordinate in block B to be able to distinguish x from x^B. A quantum algorithm can query many coordinates in superposition, thus we first need to understand what it means for a quantum algorithm to 'query' a coordinate in a block.

Turns out that the right way to formalize the above intuition is by defining

$$m_i^t = \Pr[\text{measuring } t^{th} \text{ query register outputs coordinate } i].$$

For example, if the t^{th} query is over a superposition over all n coordinates, then $m_i^t = \frac{1}{n}$ for all i. A classical query at position i will satisfy $m_i^t = 1$. We will define $m_i = \sum_t m_i^t$, which can be interpreted as the expected number of times that coordinate i is queried by the quantum algorithm.

Then for a T-query quantum algorithm to distinguish between x from x^B with constant probability, the expected number of times that the algorithm queries a coordinate in block B must be at least $\Omega\left(\frac{1}{T}\right)$. Formally,

Claim. If $\mathcal{A}$ is a T-query quantum algorithm that distinguishes between inputs $x \in \{0,1\}^n$ and x^B with probability $\geq \frac{2}{3}$ (outputs 0 on x w.p. $\geq 2/3$ and 1 on x^B w.p. $\geq 2/3$), then

$$\sum_{i \in B} m_i \geq \Omega\left(\frac{1}{T}\right).$$

The claim then implies the desired result. Consider an input x that achieves the maximum number of disjoint sensitive blocks $k = bs(f)$. Then we know that for each such block B_j, it must hold that

$$\sum_{i \in B_j} m_i \geq \Omega\left(\frac{1}{T}\right).$$

Summing over all blocks gives

$$\sum_{j=1}^{k} \sum_{i \in B_j} m_i \geq \Omega\left(\frac{k}{T}\right).$$

Since the algorithm makes T queries, the total sum of all m_i is equal to T, and because the sensitive blocks are disjoint, this is an upper bound for the LHS. Thus

$$T \geq \Omega(k/T) \implies T^2 \geq \Omega(k) \implies T = \Omega(\sqrt{bs(f)}).$$

We now proceed with proving the claim.

PROOF. [Proof of Claim] We will define Π_i to be the operator that projects the query register to the subspace where the index is equal to i. Then the definition of m_i^t is equivalent to

$$m_i^t = \|\Pi_i \left|\psi_x^t\right\rangle\|^2.$$

We have seen from Lemma 1.2 of the lecture that for a T-query quantum algorithm to succeed with probability $\geq \frac{2}{3}$, it must hold that $\|\left|\psi_x^T\right\rangle - \left|\psi_{x^B}^T\right\rangle\| \geq \frac{1}{3}$.

We will upper bound $\|\left|\psi_x^T\right\rangle - \left|\psi_{x^B}^T\right\rangle\|$ by expressing it as a telescopic sum series

$$\begin{aligned}
\|\left|\psi_x^T\right\rangle - \left|\psi_{x^B}^T\right\rangle\| &= \sum_{t=1}^{T-1} \|\left|\psi_x^{t+1}\right\rangle - \left|\psi_{x^B}^{t+1}\right\rangle\| - \|\left|\psi_x^t\right\rangle - \left|\psi_{x^B}^t\right\rangle\| \\
&= \sum_{t=1}^{T-1} \|U_{t+1} O_x \left|\psi_x^t\right\rangle - U_{t+1} O_{x^B} \left|\psi_{x^B}^t\right\rangle\| - \|\left|\psi_x^t\right\rangle - \left|\psi_{x^B}^t\right\rangle\| \\
&= \sum_{t=1}^{T-1} \left\|O_{x^B}^\dagger O_x \left|\psi_x^t\right\rangle - \left|\psi_{x^B}^t\right\rangle\right\| - \|\left|\psi_x^t\right\rangle - \left|\psi_{x^B}^t\right\rangle\| \\
&\leq \sum_{t=1}^{T-1} \left\|O_{x^B}^\dagger O_x \left|\psi_x^t\right\rangle - \left|\psi_{x^B}^t\right\rangle - \left|\psi_x^t\right\rangle + \left|\psi_{x^B}^t\right\rangle\right\| \\
&= \sum_{t=1}^{T-1} \left\|O_{x^B}^\dagger O_x \left|\psi_x^t\right\rangle - \left|\psi_x^t\right\rangle\right\| \\
&= \sum_{t=1}^{T-1} \|(O_x - O_{x^B}) \left|\psi_x^t\right\rangle\|
\end{aligned}$$

Where we used the triangle inequality at the fourth line and the fact that unitary matrices preserve the 2-norm in multiple lines. Now we observe that $|\psi_x^t\rangle = \sum_i \Pi_i |\psi_x^t\rangle$. Additionally, since x and x^B are equal for any index $i \notin B$, it holds that $O_x - O_{x^B}$

map any vector with $i \notin B$ in the index register to 0. Now we proceed with

$$\begin{aligned}
&= \sum_{t=1}^{T-1} \left\| (O_x - O_{x^B}) \sum_i \Pi_i \left|\psi_x^t\right\rangle \right\| \\
&= \sum_{t=1}^{T-1} \left\| \sum_i (O_x - O_{x^B}) \Pi_i \left|\psi_x^t\right\rangle \right\| \\
&= \sum_{t=1}^{T-1} \left\| \sum_{i \in B} (O_x - O_{x^B}) \Pi_i \left|\psi_x^t\right\rangle \right\| \\
&= \sum_{t=1}^{T-1} \sqrt{\sum_{i \in B} \left\| (O_x - O_{x^B}) \Pi_i \left|\psi_x^t\right\rangle \right\|^2} \\
&\leq \sum_{t=1}^{T-1} \sqrt{\sum_{i \in B} (2 \left\| \Pi_i \left|\psi_x^t\right\rangle \right\|)^2} \\
&\leq \sqrt{T \sum_{t=1}^{T-1} \sum_{i \in B} 4 m_i^t} \\
&= \sqrt{4T \sum_{i \in B} m_i}
\end{aligned}$$

Where the fourth line follows from the fact that the $(O_x - O_{x^B})\Pi_i \left|\psi_x^t\right\rangle$ are orthogonal for different i.

We conclude that

$$\sqrt{4T \sum_{i \in B} m_i} \geq \frac{1}{3} \implies \sum_{i \in B} m_i \geq \Omega\left(\frac{1}{T}\right).$$

□

Question 4.1

We are given that $f(x^{B'}) = f(x)$ for all proper subsets $B' \subset B$. Consider the proper subsets of the form $B \setminus \{i\}$ for all $i \in B$. Then it holds that $1 - f(x) = f(x^B) \neq f(x^{B \setminus \{i\}}) = f(x)$ for all $i \in B$.

Thus, for input x^B, there exist $|B|$ disjoint subsets $\{i\}_{i \in B}$ that change the value of f when we flip their respective bits. Thus the block sensitivity of f is at least $|B|$.

Question 4.2

Fix some $x \in \{0,1\}^n$ consider any maximal set of $k \leq bs(f)$ disjoint block sensitivity subsets $B_1, \ldots, B_k$.

WLOG, we can assume that each B_j is minimal. Otherwise, if there exists $i \in B_j$ such that $f(x^{B_j}) = f(x^{B_j \setminus \{i\}}) \neq f(x)$, then we can remove i from B_j without changing the number of subsets k.

Now we will show that $B_1 \cup \ldots \cup B_k$ is a certificate for x. Note that we have $k \leq bs(f)$ subsets, each of size at most $bs(f)$. Thus the certificate above has size $bs(f)^2$, as desired.

Claim. $B_1 \cup \ldots \cup B_k$ is a certificate for x.

Proof. By contradiction. Assume that $B_1 \cup \ldots \cup B_k$ is not a certificate for x. This means that there exists some $y \in \{0,1\}^n$ that agrees with x on $B_1 \cup \ldots \cup B_k$, but $f(x) \neq f(y)$. Write $y = x^D$, i.e. D is the subset of the bits that x and y differ. Then D is disjoint from $B_1 \cup \ldots \cup B_k$, and thus it could be added to the block sensitive subsets to get a larger set! □

Question 4.3

Consider the first $k = bs(f)^2$ iterations of the algorithm, where the algorithm chose the certificates $C_{y_1}, \ldots, C_{y_k}$. If after these $bs(f)^2$ iterations $C^{(1)}$ is empty, the algorithm terminates as desired. Thus, let's consider the case when $C^{(1)}$ is not empty. In particular, let C_z, $f(z) = 1$ be a 1-certificate still in $C^{(1)}$.

Observation. All $C_{y_i} \in C^{(0)}$ and $C_z \in C^{(1)}$ must intersect in at least one index. Otherwise, one may fix the indices in C_{y_i} according to y_i and the indices of C_z according to z and obtain an input that maps to both 0 and 1.

We can strengthen the above observation by noting that C_{y_i} and C_z must intersect in at least one index that was not queried during the previous $i-1$ iterations. Again, this is because one can fix the indices that were queried before and then fix the indices of the 0- and 1-certificates and obtain the same contradiction.

Thus what it means is that every iteration intersects with C_z in at least one new index. However, the length of C_z is at most $bs(f)^2$, and thus after $bs(f)^2$ iterations, if C_z is still in the set, it means that we have found the entire 1-certificate, which means that $C^{(0)} = \emptyset$ and the algorithm returns $f(x) = 1$. Otherwise, the set of 1-certificates is empty, and the algorithm returns $f(x) = 0$.

Question 4.4

We have showed that the algorithm terminates after at most $bs(f)^2$ repetitions. Each repetition queries the entirety of a certificate C_y, whose size is at mot $bs(f)^2$. Thus the total number of queries of the above *deterministic* algorithm is $D(f) = bs(f)^4$.

The second expression follows because any quantum algorithm can simulate a deterministic algorithm trivially, thus $Q(f) \leq D(f)$. We have also showed that $Q(f) = \Omega(\sqrt{bs(f)}) \implies bs(f) = O(Q(f)^2)$. Combining this with the paragraph above we get that $D(f) \leq O(Q(f)^8)$.