Computational Power of Quantum vs Classical Oracles
SPUR Final Paper, Summer 2013

Hyun Sub Hwang
Mentor: Adam Bouland
Project suggested by Scott Aaronson

January 23, 2014

Abstract

Comparing the computational power of quantum computers vs classical computers has been extensively studied since the invention of quantum computing. A related question is whether a quantum oracle is more powerful than a classical oracle. In this paper, we examine two topics that give partial answers to this question. We show that we can imitate a polynomial number of columns of a quantum oracle with a classical oracle. We generalize the quantum oracle separation of QMA and QCMA to obtain a quantum oracle generated by a classical oracle separation between these classes.

1 Introduction

Computer scientists conceptualize proofs as a protocol between a prover and a verifier. The prover sends a statement to the verifier and then the verifier runs an efficient polynomial time algorithm to check if the statement is true. We want the verifier to accept true statements with a high probability and reject false statements with a high probability. This type of system is known as a Merlin Arthur(MA) protocol and has been extensively studied in classical computer science theory.
Recently, with the rise of quantum computing, researchers have considered Quantum Merlin Arthur (QMA) protocols, which allows the proof statement to be a quantum state and the verifier to be a quantum computer. Likewise, they have defined proof systems in which the proof is classical, but the verifier is quantum. These are called QCMA proof systems. Then the natural question arises: is QMA powerful than QCMA? Many believe this is true, but no separation has been proven. The best known result is a quantum oracle separation between QMA and QCMA by Aaronson and Kuperberg [1]. A quantum oracle O is a set of unitary operations, $\{U_i\}$ for each $i \in \mathbb{N}$. An algorithm A with access to O, A^O, can apply U_i at cost $O(1)$ on inputs of length i. The quantum oracle separation gives a quantum oracle O such that $\mathsf{QCMA}^O \subsetneq \mathsf{QMA}^O$.
Also, we define a classical oracle, O, as a set of permutations of computational basis qubits, $\{P_i\}$ for each $i \in \mathbb{N}$. An algorithm A with access to O, A^O, can apply P_i at cost $O(1)$ on inputs of length i.
If a separation by a quantum oracle is possible, then is a separation by a classical oracle possible?

This project examines a broader question first. Can we approximate a quantum oracle with a classical oracle? Specifically, we examine the following question:

Is there a polynomial time quantum oracle algorithm A such that for all unitary matrices U,
there exists an oracle O so that A^O approximately implements U?

This is basically a question about what subset of $U(2^n)$ can be generated by quantum circuits with $\mathsf{poly}(n)$ fixed 2-local quantum gates interleaved with $\mathsf{poly}(n)$ permutations in S_{2^n} which are variable. It is known that we can prepare one arbitrary state, one column of unitary matrix, with classical oracles. If the answer for the question is "yes", it would show that classical oracles have roughly the same computational power as quantum oracles on BQP machines. We also examine whether we can separate QMA and QCMA with a classical oracle.
In section 2, we give background knowledge about quantum computing and interactive proof systems. We also review how to prepare one arbitrary state with a classical oracle. In section 3, we discuss how to prepare a quantum oracle with a classical oracle. We show how to implement a polynomial number of columns of a target unitary matrix U with a classical oracle. Although, we do not have the scheme to prepare all columns of a unitary matrix, we describe two arguments why preparing all columns of a unitary is hard. In section 4, we examine a proposed roadmap for separating QMA and QCMA by a classical oracle. The basic idea is to first separate them by a quantum oracle which is generated by a classical oracle, and then generalize this to a classical oracle separation. We describe a decision problem which can prove a separation by a quantum oracle generated by a classical oracle and may be useful in proving the separation by a classical oracle.

2 Preliminaries

2.1 Background

In a classical computer, we represent information with a bit, 0 or 1. A quantum bit, a qubit, is a linear combination of $|0\rangle$ and $|1\rangle$ with norm 1. Simply, a qubit is an element of $\mathbb{C}^2$ with norm 1. The space of quantum states of dimension n is Hilbert space $(\mathbb{C}^2)^{\otimes n}$, a Hilbert space of dimension 2^n with computational basis states of form $|x\rangle$ for $x \in \{0,1\}^n$.
A quantum algorithm is a quantum circuit where inputs and outputs are qubits and operations are quantum gates. In the circuit, we get input qubits and ancilla qubits, and we solve a quantum decision problem by measuring output qubits. Because a quantum state evolves unitarily, all quantum gates perform unitary operations. We can freely use 2-local gates. In other words, we can apply any unitary operation acting on 1 or 2 qubits. A polynomial time quantum algorithm is a circuit that can be written down by a polynomial time classical algorithm. A classical oracle in the paper is a gate on $\mathsf{poly}(n)$ qubits that performs any permutation. Simply, a classical oracle is an element of $S_{\mathsf{poly(n)}}$.
In our definition of an oracle, we can only use a fixed oracle f_i in the algorithm A acting on i qubits. However, if we add extra qubits as a counter and increase a counter by one at each stage, we can apply a different permutation at each stage. Therefore, without loss of generality, we can

apply different oracles/permutations at each call in the algorithm.

We also discuss two interactive proof system, QMA and QCMA.

Definition 2.1 (QMA and QCMA)**.** *QMA is a class of language $\mathscr{L}$ that there exists a polynomial time quantum verifier $\mathscr{Q}$ and a polynomial $p(n)$ such that*
(i) If $x \in \mathscr{L}$, there exists a quantum proof $|\phi\rangle$ of length $p(n)$ such that $\mathscr{Q}$ accepts $|x\rangle|\phi\rangle$ with a probability at least $\frac{2}{3}$.
(ii) If $x \notin \mathscr{L}$, for all quantum proofs $|\phi\rangle$ of length $p(n)$ such that $\mathscr{Q}$ rejects $|x\rangle|\phi\rangle$ with a probability at least $\frac{2}{3}$.
In QCMA, we have a classical proof of length $p(n)$, $C \in \{0,1\}^{p(n)}$, instead of a quantum proof $|\phi\rangle$.

For some $n \times n$ unitary matrix U, if we can map input state $|x\rangle$ to $U|x\rangle$ for all $x \in \{0,1\}^n$, then we can apply U on an arbitrary state, which is superposition of some $|x\rangle$ for $x \in \{0,1\}^n$. We start with creating a circuit which maps $|0\rangle^{\otimes n}$ to $U|0\rangle^{\otimes n}$ and maps the other basis states arbitrarily. This implements one column of the target unitary matrix.

2.2 Prepare one arbitrary state

We will first explain the case of two qubits. We generalize the idea later.

Proposition 2.2. *There exists a classical oracle circuit that sends $|00\rangle$ to an arbitrary state $|\psi\rangle$.*

Proof. The basic idea of the algorithm is purely classical, and is based on the fact that there is a randomized algorithm to prepare an arbitrary probability distribution of binary strings of length n with a classical oracle. The algorithm first queries the oracle to get the probability of getting 1 on the first bit. It then sets the first bit to one with that probability. In the next step, it queries the oracle to find the probability of getting a 1 on the next bit, conditioned on the outcome of its first bit. In later stages, it queries the oracle for the probability the next bit is one, conditioned on its previous measurements. This allows one to query the oracle "in superposition" - asking different questions in different cases - to approximately sample from an arbitrary distribution.
The basic idea is same as the idea of a randomized algorithm to prepare an arbitrary probability distribution of binary strings of length n with a classical oracle. We query the oracle to get a probability of getting 0 or 1 at first. Then, at each point, we query the oracle to get a probability of getting 0 or 1 based on the result we have gotten so far. With a classical oracle, we can query in superposition so we can get the exponential amount of information in a polynomial number of queries.
Likewise, using a classical oracle, we prepare a target 2 qubit state $|\psi\rangle$ in 4 queries to an oracle.
Let $p_x = |\langle x|\psi\rangle|^2$ be the probability of getting x upon measurement for $x \in \{0,1\}^2$ in which $\sum_{x\in\{0,1\}^2} p_x = 1$. Then, we can write $|\psi\rangle = \sqrt{p_{00}}|00\rangle + e^{i\theta_1}\sqrt{p_{01}}|01\rangle + e^{i\theta_2}\sqrt{p_{10}}|10\rangle + e^{i\theta_3}\sqrt{p_{11}}|11\rangle$ up to global phase.
We prepare the state in two stages. We prepare $|\phi\rangle = \sqrt{p_0}|00\rangle + e^{i\theta_2}\sqrt{p_1}|10\rangle$, in which $p_0 = p_{00} + p_{01}$ and $p_1 = p_{10} + p_{11}$ at the first stage and send it to $|\psi\rangle$ at the second stage. At each stage, we query

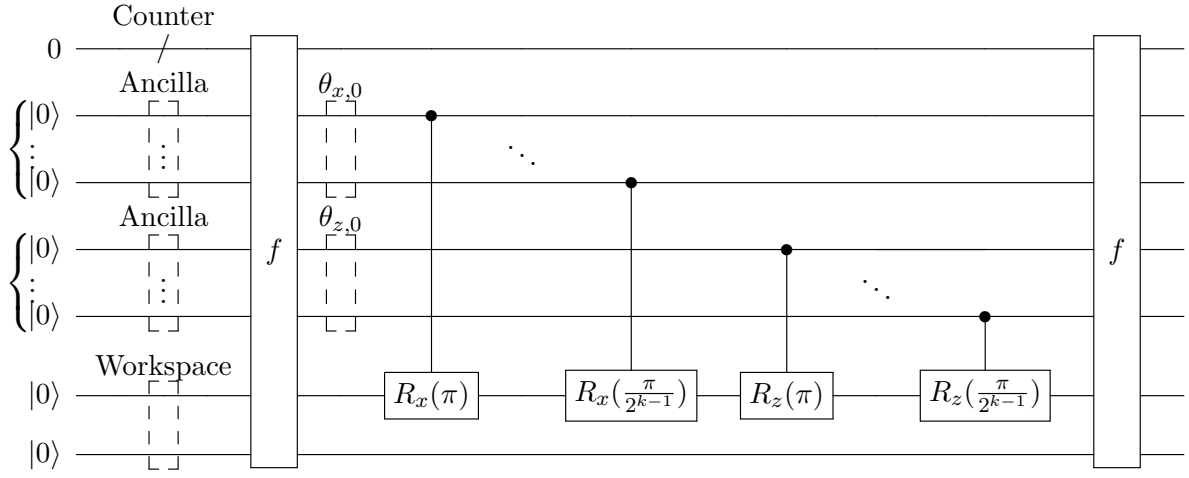
an oracle in superposition so that we can proceed each stage with 2 queries to an oracle because an oracle can output different results for different cases.

$$|00\rangle \longrightarrow |\phi\rangle = \sqrt{p_0}|00\rangle + e^{i\theta_2}\sqrt{p_1}|10\rangle \longrightarrow |\psi\rangle = \sqrt{p_{00}}|00\rangle + e^{i\theta_1}\sqrt{p_{01}}|01\rangle + e^{i\theta_2}\sqrt{p_{10}}|10\rangle + e^{i\theta_3}\sqrt{p_{11}}|11\rangle.$$

Note that there exist $\theta_{x,0}, \theta_{x,10}, \theta_{x,11}, \theta_{z,0}, \theta_{z,10}$ and $\theta_{z,11}$ such that

$$|\psi\rangle = \cos\theta_{x,0}\cos\theta_{x,10}|00\rangle + e^{i\theta_{z,10}}\cos\theta_{x,0}\sin\theta_{x,10}|01\rangle + e^{i\theta_{z,0}}\sin\theta_{x,0}\cos\theta_{x,11}|10\rangle + e^{i(\theta_{z,0}+\theta_{z,11})}\sin\theta_{x,0}\sin\theta_{x,11}|11\rangle.$$

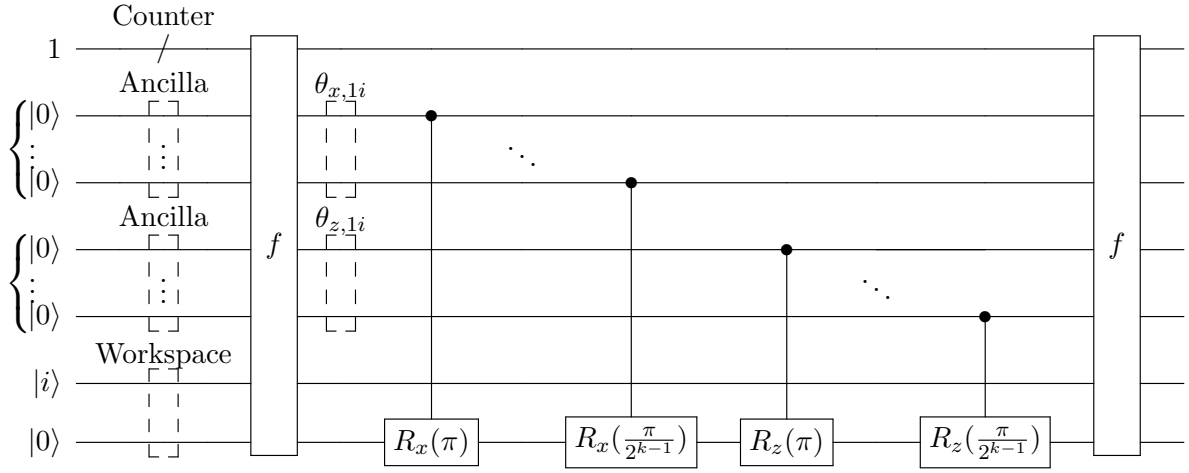
For the first step of our algorithm, we prepare $\cos\theta_{x,0}|0\rangle + e^{\theta_{z,0}}\sin\theta_{x,0}|1\rangle$ in the first qubit. Approximate $\theta_{x,0}$ and $\theta_{z,0}$ in two sets of ancilla qubits such that they are approximated as $\theta_{x,0} \approx \sum_{i=0}^{k-1} x_i \frac{\pi}{2^{i-1}}$ and $\theta_{z,0} \approx \sum_{i=0}^{k-1} y_i \frac{\pi}{2^{i-1}}$. (The error here is exponentially decreased by increasing the number of ancilla qubits.)
Set an oracle f with counter 0 to send first ancilla qubits $|0\rangle$ to $|0\rangle \oplus |x_0 x_1 \dots x_{k-1}\rangle$, second ancilla qubits $|0\rangle$ to $|0\rangle \oplus |y_0 y_1 \dots y_{k-1}\rangle$ and leave the other qubits as they were. Then, applying controlled R_x gates on first ancilla qubits, we can rotate $|0\rangle$ to approximate $\cos\theta_{x,0}|0\rangle + \sin\theta_{x,0}|1\rangle$. Also, applying controlled R_z, we can control the relative phase to $\cos\theta_{x,0}|0\rangle + e^{i\theta_{z,0}}\sin\theta_{x,0}|1\rangle$
After R_x and R_z gates, we get the states we want on workspace. However, the state we have gotten is the entangled state $|0\rangle|x_1 x_2 \dots x_{k-1}\rangle|y_1 y_2 \dots y_{k-1}\rangle|\phi\rangle$ where the x and y are the binary representation of $\theta_{x,0}$ and $\theta_{z,0}$. To disentangle these ancilla qubits from our workspace, we "uncompute" f by applying f again and get $|0\rangle|0\rangle^{\otimes n}|0\rangle^{\otimes n}|\phi\rangle$.



Let's see how this circuit works in detail.

$$\begin{aligned}
|0\rangle|0\rangle^{\otimes k}|0\rangle^{\otimes k}|0\rangle|0\rangle &\xrightarrow{f} |0\rangle|x_0 x_1 \dots x_{k-1}\rangle|y_0 y_1 \dots y_{k-1}\rangle|0\rangle|0\rangle \\
&\xrightarrow{R_x \text{ gates}} |0\rangle|x_0 x_1 \dots x_{k-1}\rangle|y_0 y_1 \dots y_{k-1}\rangle(\cos\theta_{x,0}|0\rangle + \sin\theta_{x,0}|1\rangle)|0\rangle \\
&\xrightarrow{R_z \text{ gates}} |0\rangle|x_0 x_1 \dots x_{k-1}\rangle|y_0 y_1 \dots y_{k-1}\rangle(\cos\theta_{x,0}|0\rangle + e^{i\theta_{z,0}}\sin\theta_{x,0}|1\rangle)|0\rangle \\
&\xrightarrow{f} |0\rangle|0\rangle^{\otimes k}(\cos\theta_{x,0}|0\rangle + e^{i\theta_{z,0}}\sin\theta_{x,0}|1\rangle)|0\rangle
\end{aligned}$$

Now let's apply operations on the second qubit. We rotate $|00\rangle$ and $|10\rangle$ by different angles, $\theta_{x,10}$ and $\theta_{x,11}$, and control phase also by different angles, $\theta_{z,10}$ and $\theta_{z,11}$. We use the state of the first qubit to determine which angle to apply.



The circuit above allows us to send $|00\rangle \to \cos\theta_{x,10}|00\rangle + e^{\theta_{z,10}} \sin\theta_{x,10}|10\rangle$ and $|10\rangle \to \cos\theta_{x,11}|10\rangle + e^{\theta_{z,11}} \sin\theta_{x,11}|11\rangle$ using a single query to the oracle. More precisely, the state evolves as follows:

$$
\begin{aligned}
|0\rangle|0\rangle^{\otimes k}|0\rangle^{\otimes k}(\cos\theta_{x,0}|00\rangle + e^{i\theta_{z,0}}\sin\theta_{x,0}|10\rangle) &\xrightarrow{f} \cos\theta_{x,0}|0\rangle|\theta_{x,10}\rangle|\theta_{z,10}\rangle|00\rangle + e^{i\theta_{z,0}}\sin\theta_{x,0}|0\rangle|\theta_{x,11}\rangle|\theta_{z,11}\rangle|10\rangle \\
&\xrightarrow{R_x\ gates} \cos\theta_{x,0}|0\rangle|\theta_{x,10}\rangle|\theta_{z,10}\rangle(\cos\theta_{x,10}|00\rangle + \sin\theta_{x,10}|01\rangle) \\
&+ e^{i\theta_{z,0}}\sin\theta_{x,0}|0\rangle|\theta_{x,11}\rangle|\theta_{z,11}\rangle(\cos\theta_{x,10}|10\rangle + \sin\theta_{x,11}|11\rangle) \\
&\xrightarrow{R_z\ gates} \cos\theta_{x,0}|0\rangle|\theta_{x,10}\rangle|\theta_{z,10}\rangle(\cos\theta_{x,10}|00\rangle + e^{i\theta_{z,10}}\sin\theta_{x,10}|01\rangle) \\
&+ e^{i\theta_{z,0}}\sin\theta_{x,0}|0\rangle|\theta_{x,11}\rangle|\theta_{z,11}\rangle(\cos\theta_{x,10}|10\rangle + e^{i\theta_{z,11}}\sin\theta_{x,11}|11\rangle) \\
&\xrightarrow{f} \cos\theta_{x,0}|0\rangle|0\rangle^{\otimes k}|0\rangle^{\otimes k}(\cos\theta_{x,10}|00\rangle + e^{i\theta_{z,10}}\sin\theta_{x,10}|01\rangle) \\
&+ e^{i\theta_{z,0}}\sin\theta_{x,0}|0\rangle|0\rangle^{\otimes k}|0\rangle^{\otimes k}(\cos\theta_{x,10}|10\rangle + e^{i\theta_{z,11}}\sin\theta_{x,11}|11\rangle) \\
&= |0\rangle|0\rangle^{\otimes k}|0\rangle^{\otimes k}|\psi\rangle
\end{aligned}
$$

Note that disentanglement is necessary because the workspace is entangled with ancilla qubits before we uncompute them by f. By entangled, we mean the state cannot be written as a tensor product. We can disentangle the state with the first qubit as an indicator. From the circuit above, we get the desired states

$$|\psi\rangle = \cos\theta_{x,0}\cos\theta_{x,10}|00\rangle + e^{i\theta_{z,10}}\cos\theta_{x,0}\sin\theta_{x,10}|01\rangle + e^{i\theta_{z,0}}\sin\theta_{x,0}\cos\theta_{x,11}|10\rangle + e^{i(\theta_{z,0}+\theta_{z,11})}\sin\theta_{x,0}\sin\theta_{x,11}|11\rangle.$$

Therefore, we can approximately prepare any state we want by choosing an appropriate classical oracle. □

This quantum circuit does not exactly prepare $|\psi\rangle$, but prepares $|\psi\rangle$ approximately. Here we say that two unitary matrices U and V are ϵ-close if they differ by ϵ in the operator norm that is $\max_{\psi} \|(U-V)|\psi\rangle\|_2$. Note that this type of error adds linearly from operation to operation. Thus, if we have error tolerance ϵ, a polynomial number of gates and ancilla qubits, we can have up to $\frac{\epsilon}{\mathsf{poly(n)}}$ error at each stage and, therefore only need $\log \frac{\epsilon}{\mathsf{poly(n)}}$ qubits of ancilla qubits to achieve this accuracy.
Now we look at the general case.

Proposition 2.3. *There exists a polynomial time classical oracle circuit C such that for any state $|\psi\rangle$ of length n, there exists a classical oracle such that sends $|0\rangle^{\otimes n}$ to $|\psi\rangle$ for all $n \in \mathbb{N}$.*

Proof. We can make an arbitrary state in n stages as we have done for two qubits. We generalize how to implement different operations to different states. Querying in superposition, we can specify 2^{k-1} states in the kth oracle query, which allows us to prepare an n-qubits state in $O(n)$ queries. Specifically, on kth stage, we have prepared $\sum_{x\in\{0,1\}^{k-1}} t_x|x\rangle|0\rangle^{\otimes n-k+1}$. Then, if first $k-1$ qubits are $|x\rangle$ then prepare appropriate angles θ_x and $\theta_{z,x}$, apply rotation and phase correction, and then apply the oracle again to disentangle the state with the ancilla qubits. We can disentangle ancilla qubits using first $k-1$ qubits as an indicator. As before, one can easily check that this performs the right operations and prepares the arbitrary states. □

We have shown that we can prepare an arbitrary state starting from $|0\rangle^{\otimes n}$ in a polynomial time. Note that we can also prepare an arbitrary state from any computational basis states. In the proof above, we have used some portion of workspace to indicate which case we are in to apply different angles in different cases. We can generalize this scheme to add extra n bits, copy our workspace on them, and use them as to create different states $U|x\rangle$ for each computational basis state $|x\rangle$. Thus, next corollary follows.

Corollary 2.4. *For an arbitrary unitary matrix U, there exists a classical oracle circuit that sends $|x\rangle|0\rangle^{\otimes n}$ to $U|x\rangle|x\rangle$ for all x in $\{0,1\}^n$.*

However, this is not the same as applying U to the workspace because the workspace is entangled with the ancilla qubits. This performs a more general super operator on the workspace which maps ρ to $\sum_x E_x \rho E_x^\dagger$ where $E_x = U|x\rangle\langle x|U^\dagger$. To preform U on the workspace, we would need to map $|x\rangle|0\rangle^{\otimes n}$ to $U|x\rangle|0\rangle^{\otimes n}$.
At least, we have seen that we can prepare $U|x\rangle$ somehow by entangling it with n extra workspace qubits. The open question is if it is feasible to produce $U|x\rangle$ using only n bits of workspace without entangling it with any other qubits.

3 Prepare unitary matrix with a classical oracle

In this section, we give a proof that there exists an algorithm to prepare a polynomial number of columns of a unitary matrix with a classical oracle. Then, we give a counting argument that

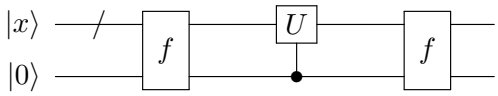
provides why the current scheme cannot be generalized to prepare all columns. We also describe another difficulty with our scheme.

3.1 Preparing a polynomial number of Columns

To prepare a polynomial number of columns, we prepare one desired state at a time. We stabilize the other basis states using the following lemma.

Lemma 3.1 (Stabilization Lemma)**.** *Let S be a set of computational basis states and U be a unitary gate we can perform. Then, there is a classical oracle circuit C that performs a unitary mapping $|x\rangle \to |x\rangle$ for $x \in S$ unitarilly and $|y\rangle \to U|y\rangle$ for $y \notin \mathsf{span}(S)$ as a super operator. If $\langle x|U|y\rangle = 0$, C maps $|y\rangle \to U|y\rangle$ unitarilly. Furthermore if U can be implemented in a polynomial time, then so can C.*

Proof. Add an extra qubit on the workspace. We use it as an indicator. Then, using a classical oracle, we can switch the extra qubit to 1 if and only if the basis state on the workspace is not in S. Then, add the extra bit to our controlled gates as a control bit. In other words, let f be a map that sends $|x\rangle|i\rangle \to |x\rangle|i\rangle$ for $x \in S$ and $|y\rangle|i\rangle \to |y\rangle|i \oplus 1\rangle$ for $y \notin \mathsf{span}(S)$.
Then, we can apply the operation on only the states we want in the following circuit.



The circuit above applies the operation on the basis states as follows:

$$\text{if } x \in S,\ |x\rangle|0\rangle \xrightarrow{f} |x\rangle|0\rangle \xrightarrow{U \text{ gate}} |x\rangle|0\rangle \xrightarrow{f} |x\rangle|0\rangle$$
$$\text{if } y \notin \mathsf{span}(S),\ |y\rangle|0\rangle \xrightarrow{f} |y\rangle|1\rangle \xrightarrow{U \text{ gate}} U|y\rangle|1\rangle \xrightarrow{f} U|y\rangle|0\rangle$$

Note that this operation is reversible if $\langle x|U|y\rangle$ because for $x \in S$ and $y \notin S$, we have $\langle x|y\rangle = 0$ and $\langle x|U|y\rangle = 0$. In either case, the ancilla qubit is left in the $|0\rangle$ state so is unentangled with the workspace. If this were not the case, then we would have that

$$U|y\rangle = \sum_{x \in S} \alpha_x |x\rangle + \sum_{y \notin S} \alpha_y |y\rangle$$

with $\alpha_x \neq 0$ for some $x \in S$. Then the circuit evolves as

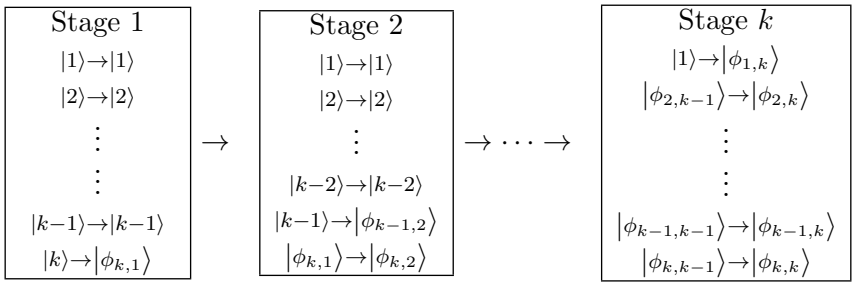
$$|y\rangle|0\rangle \xrightarrow{f} |y\rangle|1\rangle \xrightarrow{U \text{ gate}} U|y\rangle|1\rangle = \sum_{x \in S} \alpha_x |x\rangle|1\rangle + \sum_{y \notin S} \alpha_y |y\rangle|1\rangle \xrightarrow{f} \sum_{x \in S} \alpha_x |x\rangle|1\rangle + \sum_{y \notin S} \alpha_y |y\rangle|0\rangle$$

which leaves the workspace entangled with the ancilla qubits. Therefore, we need to have $\langle x|U|y\rangle = 0$ not to have entanglement generated by a circuit. □

Using the Stabilization Lemma, we can prepare a polynomial number of columns of a target unitary matrix in a polynomial time.

Theorem 3.2. *There exists a polynomial time quantum oracle algorithm such that for any set S of computational basis states of length n where $|S| = O(\mathsf{poly}(n))$, and any set of orthonormal states $\{\psi_i\}_{|i\rangle \in S}$, there exists a classical oracle such that the algorithm maps $|i\rangle$ to ψ_i for all $|i\rangle \in S$.*

Proof. For simplicity, we consider S to be $\{|1\rangle, |2\rangle, \ldots, |k\rangle\}$.
The basic idea is to send a state $|i\rangle$ to its target state $|\psi_i\rangle$ at each stage while stabilizing the other computational basis states. The difficulty with doing this, however, is that once one has prepared $|\psi_i\rangle$, there is no way to stabilize $|\psi_i\rangle$ since the Stabilization Lemma gives us the power to stabilize only computational basis states.
To fix this, we make use of the orthogonality of the $|\psi_i\rangle$. Consider the case that k is 2. First, we implement the circuit which maps $|1\rangle$ to $|\psi_1\rangle$. Since this map is unitary, there exists some state $|\phi_{2,1}\rangle$ which is mapped to $|\psi_2\rangle$ under this transformation where $\langle\phi_{2,1}|1\rangle = 0$. Thus using the Stabilization Lemma, we can create a circuit which maps $|1\rangle \to |1\rangle$ and $|2\rangle \to |\phi_{2,1}\rangle$ due to this orthogonality. Composing these circuits map $|1\rangle \to |1\rangle \to |\psi_1\rangle$ and $|2\rangle \to |\phi_{2,1}\rangle \to |\psi_2\rangle$.
We can generalize this to map k states to $\psi_1, \psi_2, \ldots, \psi_k$ in k stages. The basic flow of the algorithm is below.



For all $i \in S$, $\phi_{i,k} = \psi_i$ and we will define $\phi_{a,b}$ inductively in reverse direction.
At stage t, we implement the circuit which maps $|k-t+1\rangle$ to $|\phi_{k-t+1,t}\rangle$. Then, there exists $|\phi_{j,t-1}\rangle$ which is mapped to $|\phi_{j,t}\rangle$ for $j \geq k-t+2$. Since we have $\langle i|\phi_{j,t}\rangle = 0$ for $1 \leq i \leq k-t$ and $k-t+2 \leq j \leq k$ in stage $t+1$, there exists a circuit that send $|i\rangle$ to $|i\rangle$ for $1 \leq i \leq k-t$, $|\phi_{j,t-1}\rangle$ to $|\phi_{j,t}\rangle$ for $k-t+2 \leq j \leq k$, and $|k-t+1\rangle$ to $|\phi_{k-t+1,t}\rangle$ by the Stabilization Lemma.
Thus, there exists a circuit that implements the operation at stage t as in the diagram, so we can implement a polynomial number of columns of a target unitary matrix. □

Note that the circuit in Theorem 3.2 does not act unitarilly on the other computational basis states. The circuit acts as a more general super operator on these states. At stage t, the state $|x\rangle$ is mapped to $|y\rangle$ where $\langle y|i\rangle$ is not guaranteed for $i \in S$. Then, as we have seen in Lemma 3.1, our state is entangled with an ancilla qubit.
However, we can prepare a unitary matrix if all the other computational basis states are stabilized.

Corollary 3.3. *There exists a polynomial time quantum oracle algorithm such that for any set S of computational basis states of length n where $|S| = O(\mathsf{poly}(n))$, and any basis $\{|\psi_i\rangle\}_{|i\rangle \in S}$ of $\mathsf{span}(S)$, there exists a classical oracle such that the algorithm maps $|i\rangle$ to $|\psi_i\rangle$ for all $|i\rangle \in S$ and $|j\rangle$ to $|j\rangle$ for all the other computation basis states $|j\rangle \notin S$*

We have shown that there exists a classical oracle circuit that prepares a polynomial number of columns of a unitary matrix. However, the current scheme cannot be generalized to prepare all columns of a unitary matrix. We give a counting argument which shows this is not possible in the next section.

3.2 Counting Argument

We now show our current approach to make a polynomial number of arbitrary qubits cannot be generalized to prepare an arbitrary unitary matrix. Note that we want to prepare all 2^n columns of a unitary matrix. Our current scheme is as follows:
Prepare angles in ancilla qubits with total error ϵ and apply a prepared angle on workspace using quantum gates.
Note that quantum gates do not help in preparing multiple columns: quantum gates do fixed operations. Thus, we only consider our classical oracles to look at how many states we can prepare. Ancilla qubits prepare $\frac{\epsilon}{\mathsf{poly(n)}}$ error in each oracles. Then, we have $\log \frac{\mathsf{poly(n)}}{\epsilon}$ bits in the oracle and can make $\frac{\mathsf{poly(n)}}{\epsilon}$ number of angles. Also, we prepare each angle for each basis vector $|x\rangle$ for $x \in \{0,1\}^n$.
Therefore, we can prepare $\left(\frac{\mathsf{poly(n)}}{\epsilon}\right)^{2^n}$ states in one oracle. We have $\mathsf{poly(n)}$ number of oracles, so we can prepare $\left((\frac{\mathsf{poly(n)}}{\epsilon})^{2^n}\right)^{\mathsf{poly(n)}} = \left(\frac{\mathsf{poly(n)}}{\epsilon}\right)^{2^n \mathsf{poly(n)}}$ states in total.

The volume of an ϵ-ball around a state is ϵ^{2^n}. Then, we need a $(\frac{1}{\epsilon})^{2^n}$ states to represent a single state. Then, to represent a unitary matrix which has 2^n columns, we need $\left((\frac{1}{\epsilon})^{2^n}\right)^{2^n} = (\frac{1}{\epsilon})^{2^{2n}}$ states.

Let's compare two cases. Our scheme can prepare roughly $\left(\frac{\mathsf{poly(n)}}{\epsilon}\right)^{2^n \mathsf{poly(n)}}$ different unitary matrices, but the number of distinct unitary matrices up to error tolerance ϵ is roughly $(\frac{1}{\epsilon})^{2^{2n}}$. Taking log on each side, we need

$$2^n \mathsf{poly}(n) \log \frac{\mathsf{poly}(n)}{\epsilon} \geq 2^{2n} \log \frac{1}{\epsilon}.$$

However, it is easy to see that $(\frac{1}{\epsilon})^{2^{2n}}$ grows much faster as $n \to \infty$. Thus, our scheme cannot implement an arbitrary unitary matrix for large n.
Note that in our previous scheme, we had only used a small fraction of the number of classical oracles available. In particular, we only consider oracles that act as an identity on the workspace rather than perform $2^n!$ different permutations. Using the Stirling's formula, we have $2^n! \approx (2^n)^{2^n} = 2^{n2^n}$ different oracles at each stage. Thus, in total we have $2^{n2^n \mathsf{poly}(n)}$ number of states we can prepare. However, the inequality

$$n2^n \mathsf{poly}(n) \geq 2^{2n} \log \frac{1}{\epsilon}$$

also does not hold for large n.

The above counting argument shows that we cannot prepare an arbitrary unitary matrix using our current scheme. Also, it shows that we need to use at least 2^{2n} different oracles setting in our algorithm if we want to generate all columns of a unitary matrix. For instance, enlarging the workspace to size $2n$ would suffice.

Also, Corollary 2.4 shows that there are no information theoretic barriers in preparing a unitary in this fashion. This suggests that there might be another scheme to prepare a unitary matrix with a classical oracle.

3.3 Difficulties in extending our current scheme

We attempted to extend our scheme to implement a super polynomial number of columns of a target unitary matrix while evading the counting argument shown above. However, we ran into several difficulties when working with this. For instance, we tried to extend our scheme to implement more than one quantum state at each stage. The problem we ran into is that it is difficult to disentangle the ancilla qubits when preparing multiple states. For instance, suppose we want to map $|00\rangle$ to $|\psi\rangle$ and $|01\rangle$ to $|\phi\rangle$ using the same number of queries as is required to map $|00\rangle$ to $|\psi\rangle$. We can easily prepare a unitary to map

$$|00\rangle \to \sqrt{p_0}|00\rangle + \sqrt{p_1}|10\rangle \text{ and } |01\rangle \to \sqrt{p'_0}|01\rangle + \sqrt{p'_1}|11\rangle.$$

Furthermore, we can easily specify the rotation angles θ_{ij} to rotate $|i\rangle|j\rangle$ to map

$$\sqrt{p_0}|00\rangle + \sqrt{p_1}|10\rangle \to |\psi\rangle \text{ and } \sqrt{p'_0}|01\rangle + \sqrt{p'_1}|11\rangle \to |\phi\rangle.$$

However, after this operation the total state of the system evolves as

$$|0\rangle|00\rangle \to |\theta_{00}\rangle\sqrt{p_{00}}|00\rangle + |\theta_{00}\rangle\sqrt{p_{01}}|01\rangle + |\theta_{01}\rangle\sqrt{p_{10}}|10\rangle + |\theta_{01}\rangle\sqrt{p_{11}}|11\rangle$$
$$|0\rangle|10\rangle \to |\theta_{10}\rangle\sqrt{p'_{00}}|00\rangle + |\theta_{10}\rangle\sqrt{p'_{01}}|01\rangle + |\theta_{11}\rangle\sqrt{p'_{10}}|10\rangle + |\theta_{11}\rangle\sqrt{p'_{11}}|11\rangle$$

which is highly entangled with ancilla qubits. It does not seem possible to remove this entanglement with the ancilla qubits.

4 Classical Oracle Separation

In this section we denote A to be a classical oracle and $U(A)$ to be a unitary matrix that is generated by a classical oracle A. To separate QMA and QCMA with a classical oracle, we first try to prove a quantum oracle separation where the quantum oracle is generated by a classical oracle A. We later explore separating QMA and QCMA using the classical oracle underlying this construction directly. We implement the hybrid argument by Aaronson and Kuperberg[1], which is a generalization of the argument of Bennett et al[2].
We propose a problem that is in $QMA^{U(A)}$ but not in $QCMA^{U(A)}$ that leads to prove $\mathsf{QCMA}^{U(A)} \subsetneq$

$\mathsf{QMA}^{U(A)}$. We also conjecture that $\mathsf{QMA}^A \neq \mathsf{QCMA}^A$ can be proven with the same problem. We take the definition of p-uniform and Lemma 3.2 that bounds the difference between oracles the paper by Aaronson and Kuperberg[1]

Definition 4.1. *For all $p \in [0,1]$, a probability measure σ over $\mathbb{CP}^{N-1}$ is called p-uniform if $p\sigma \leq \mu$. Equivalently, σ is p-uniform if it can be obtained by starting from μ, and then conditioning on an event that occurs with probability at least p.*

Lemma 4.2. *Let σ be a p-uniform probability measure over $\mathbb{CP}^{N-1}$. Then for all ρ,*

$$\mathop{\mathbb{E}}_{|\psi\rangle\in\sigma} \langle\psi|\rho|\psi\rangle = O\left(\frac{1+\log\frac{1}{p}}{N}\right)$$

We define the ball around $|\psi\rangle$ with radius ϵ on $\mathbb{CP}^{N-1}$, $B_\epsilon(|\psi\rangle)$ as

$$\{|\phi\rangle \| \||\phi\rangle - |\psi\rangle\| \leq \epsilon\}$$

Also, we define $\mu(K)$ as a uniform measure on K.

Dealing with a classical oracle, we do not have continuous measure on quantum sapce. Instead, we have a distribution on quantum space. We will see how to apply Lemma 4.2 to a classical distribution. We define a similar concept on distribution as p-uniform measure on measure space.

Definition 4.3 (ϵ error p-pseudo uniform distribution)**.** *A distribution X on $\mathbb{CP}^{N-1}$ is ϵ error p-pseudo uniform if probability measure $\frac{1}{|X|}\sum_{|\phi\rangle\in X} \mu(B_\epsilon(|\psi\rangle))$ is p - uniform measure.*

To consider multiplicity when summing up measure, we need following lemma.

Lemma 4.4. *For a distribution X on $\mathbb{CP}^{N-1}$ such that $\mathop{\mathbb{P}}_{|\phi\rangle\in\mathbb{CP}^{N-1}}[|\phi\rangle \in \bigcup_{|\psi\rangle\in X} B_\epsilon(|\psi\rangle)] \geq p$ for $|\psi\rangle \in \mathbb{CP}^{N-1}$, $\frac{1}{|X|}\sum_{|\phi\rangle\in X} \mu(B_\epsilon(|\psi\rangle))$ is $\frac{p}{|X|}$-uniform measure.*

Proof. $\mu(\bigcup_{|\psi\rangle\in X} B_\epsilon(|\psi\rangle))$ is p-uniform measure. Then, each $x \in \bigcup_{|\psi\rangle\in X} B_\epsilon(|\psi\rangle)$ has at most $|X|$ multiplicity when summing up measures and $\frac{1}{|X|}\sum_{|\phi\rangle\in X} \mu(B_\epsilon(|\psi\rangle))$ is $\frac{p}{|X|}$-uniform. □

The following lemma gives the connection between inequality on ϵ error p-pseudo uniform distribution and inequality on p-uniform measure.

Lemma 4.5. *For a density matrix ρ,*

$$\langle\psi|\rho|\psi\rangle \leq 2 \mathop{\mathbb{E}}_{|\phi\rangle\in B_\epsilon(|\phi\rangle)} \langle\phi|\rho|\phi\rangle$$

Proof. It is enough to show that

$$\langle\psi|\alpha\rangle^2 \leq 2 \mathop{\mathbb{E}}_{|\phi\rangle\in B_\epsilon(|\phi\rangle)} \langle\phi|\alpha\rangle^2.$$

We know,

$$\langle\phi|\alpha\rangle^2 = (\langle\phi-\psi|\alpha\rangle + \langle\psi|\alpha\rangle)^2 \geq \langle\psi|\alpha\rangle^2 + 2\langle\psi|\alpha\rangle\langle\phi-\psi|\alpha\rangle$$

Then, for hemi-sphere of $B_\epsilon(\psi)$, $\langle\psi|\alpha\rangle\langle\phi-\psi|\alpha\rangle$ is positive and $\langle\phi|\alpha\rangle^2 \geq \langle\psi|\alpha\rangle^2$. Thus, we get

$$\langle\psi|\alpha\rangle^2 \leq 2 \mathop{\mathbb{E}}_{|\phi\rangle\in B_\epsilon(|\phi\rangle)} \langle\phi|\alpha\rangle^2.$$

□

Corollary 4.6. *For ϵ error p-pseudo uniform distribution X,*

$$\mathop{\mathbb{E}}_{|\psi\rangle\in X} \langle\psi|\rho|\psi\rangle = O\left(\frac{1+\log\frac{1}{p}}{N}\right)$$

Proof. By definition, $\frac{1}{|X|}\sum_{|\phi\rangle\in X} \mu(B_\epsilon(|\psi\rangle))$ is p-uniform. Then, by Lemma 4.2,

$$\begin{aligned}
\mathop{\mathbb{E}}_{|\psi\rangle\in X} \langle\psi|\rho|\psi\rangle &\leq 2 \mathop{\mathbb{E}}_{|\psi\rangle\in X} \mathop{\mathbb{E}}_{|\phi\rangle\in B_\epsilon(|\psi\rangle)} \langle\phi|\rho|\phi\rangle \\
&= 2 \mathop{\mathbb{E}}_{|\phi\rangle\in B_\epsilon(|\psi\rangle)} \langle\phi|\rho|\phi\rangle \\
&= 2O\left(\frac{1+\log\frac{1}{p}}{N}\right) = O\left(\frac{1+\log\frac{1}{p}}{N}\right)
\end{aligned}$$

□

By Corollary 2.4, we know how to implement a target unitary matrix if we entangle the results with ancilla qubits. In fact, we will implement any unitary matrix that can be prepared with a classical oracle on the workspace using ancilla bits as indicators. Also, using the last half of the workspace as indicators, we can implement a unitary operation on the first half of the workspace. In other words, let O_c to be a set of unitary matrix that is implemented by a classical oracle and O_q to be a set of unitary matrix that consists of O_c. Then, we define O_c and O_q as follows: for simplicity let a^n denotes a succesion of a for n times for $a \in \{0,1\}$.

Definition 4.7. *Let $U_{\psi,x}$ be a unitary matrix performed by an algorithm in Proposition 2.3 when preparing $|\psi\rangle$ starting from $|x\rangle$. In other words, U consists of n oracles in which kth oracle encodes angles to apply on kth qubit on the workspace.*
Let $U(\{\psi_x\}_{x\in\{0,1\}^n})$ to be a unitary matrix which performs the unitary operation which maps $|y\rangle|x\rangle$ to $(U_{\psi_x,x}|y\rangle)|x\rangle$ for all $x \in \{0,1\}^{\frac{n}{2}}$. In other words, $U(\{\psi_x\}_{x\in\{0,1\}^n})$ performs $U_{\psi_x,x}$ on the first $\frac{n}{2}$ qubits of the workspace if the last $\frac{n}{2}$ qubits is $|x\rangle$.
O_c is a set of $U_{\psi,x}$ and O_q is a set of $U(\{\psi_x\}_{x\in\{0,1\}^n})$ for all $\psi \in \mathbb{CP}^{n-1}$, $\psi_x \in \mathbb{CP}^{n-1}$ and $x \in \{0,1\}^n$.

O_{qq} is a set of U such that $U = (I \otimes U^{\dagger}_{\phi,0^{\frac{n}{2}}})U(\{\psi_x\}_{x\in\{0,1\}^n})(I \otimes U_{\phi,0^{\frac{n}{2}}})$ for some $\phi \in \mathbb{CP}^{\frac{n}{2}-1}$, $\psi_x \in \mathbb{CP}^{n-1}$ and $x \in \{0,1\}^n$. In other words, U performs $U_{\psi_x,x}$ on the first $\frac{n}{2}$ qubits of the workspace if the last $\frac{n}{2}$ qubits is $|\psi_x\rangle$ and performs I otherwise.

Note that the oracle $U \in O_q$ is a unitary matrix of length $n+k$ where k is a constant because we have a counter and ancilla bits to encode angles. Now we want to prove the following conjecture.

Theorem 4.8. *Given access to a quantum oracle U of length $n+k$, we want to decide which case U is in between the following two cases:*

(i) U is an identity (drawn from O_{qq}).

(ii) U is drawn uniformly at random from O_{qq} conditioned on that there exist two quantum states of length $\frac{n}{2}$ $|\psi\rangle$ & $|\phi\rangle$ and a state of length $\frac{n}{2}$ $|\xi\rangle$ such that $U|\psi\rangle|\xi\rangle = |\phi\rangle|\xi\rangle$ and $U|\gamma\rangle|\delta\rangle = |\gamma\rangle|\delta\rangle$ for all $|\delta\rangle$ orthogonal to $|\xi\rangle$.

Then, even with a classical proof of length m, we need at least $\Omega(\sqrt{\frac{1+m+n\log\frac{1}{\epsilon}}{2^{\frac{n}{2}}}})$ queries to an oracle U.

Proof. If $m = O(2^n)$, it is obvious. We consider the case where $m = o(2^n)$. Let ω be a classical proof of length m we get. Let $U_f = I$ and U_s be a quantum oracle drawn with first condition and second condition respectively. Suppose we have an algorithm A and A queries the oracle T times. A want to accept if $U = U_s$. Fix ψ and ϕ. Then, for each ξ, there exists $\omega \in \{0,1\}^m$ such that probability of accepting U_s is largest for ω. Let X be a uniform probability distribution of states represented in O_c. By definition, $\bigcup_{|\psi\rangle\in X} B_\epsilon(\psi) = \mathbb{CP}^{N-1}$. Also, we know that $|X| = O\left((\frac{1}{\epsilon})^n\right)$ as we have seen in Section 3.2. Then denote $S(\omega)$ be a set of ξ such that probability of accepting U_s is largest with a classical proof ω. Then, there exists ω^* such that

$$\mathbb{P}_{|\xi\rangle\in X}[|\xi\rangle \in S(w^*)] \geq \frac{1}{2^m}$$

Then, it is enough to prove that when we hardwire ω^* into a circuit and draw $|\xi\rangle$ uniformly at random from $S(\omega^*)$ with $o\left(\sqrt{\frac{1+m+n\log\frac{1}{\epsilon}}{2^{\frac{n}{2}}}}\right)$ quries, we cannot distinguish U_s and U_f with a high probablilty. Let $|\Phi_t\rangle$ be a state after applying U_f for first t queries and U_s for last $T-t$ queries. We want the difference between $|\Phi_0\rangle$ and $|\Phi_T\rangle$ to be $\Omega(1)$. Let $\rho_t = \sum_{i=1}^{k} p_i|\phi_i\rangle\langle\phi_i|$ be a density matrix of $|\Phi_t\rangle$. Note that U_f and U_s differ only in states that end with ξ. Let $|\alpha\rangle = \sum_{x\in\{0,1\}^{\frac{n}{2}}} |x\rangle|\xi\rangle$. Then, $|\Phi_{t+1}\rangle$ and $|\Phi_t\rangle$ differ by at most $2\sum p_i\langle\alpha|\phi_i\rangle$. Denote $\rho'_t = \sum_{i=1}^{k} p_i \sum_{j=1}^{2^{\frac{n}{2}}} q_{i,j}|j\rangle\langle j| = \sum_{i=1}^{s} r_i|\delta_i\rangle\langle\delta_i|$

where $q_{i,j}$ is the probability of getting j from $|\phi_i\rangle$ when j is represented in binary form. Thus,

$$\||\Phi_{t+1}\rangle - |\Phi_t\rangle\|_2 = \|(U_f U_s^\dagger - I)|\Phi_t\rangle\|_2 \leq \sum_{i=1}^{s} 2r_i\langle\delta_i|\xi\rangle = 2\sum_{i=1}^{s} r_i\sqrt{\langle\delta_i|\xi\rangle\langle\delta_i|\xi\rangle} \leq 2\sqrt{\sum_{i=1}^{s} r_i\langle\delta_i|\xi\rangle\langle\delta_i|\xi\rangle} =$$

$2\sqrt{\langle\xi|\rho'|\xi\rangle}$.

Let σ be a unifrom probability measure from $S(w^*)$. Note that $S(w^*)$ is $\frac{1}{2^m|X|}$-uniform measure. By Lemma 4.6,

$$\mathop{\mathbb{E}}_{\xi\in\sigma}\langle\xi|\rho'|\xi\rangle \leq \sqrt{\frac{1+\log\frac{1}{2^{-m}}}{2^{\frac{n}{2}}}} = \sqrt{\frac{1+m+\log|X|}{2^{\frac{n}{2}}}} =$$

. Therefore,

$$\mathop{\mathbb{E}}_{\xi\in\sigma}\||\Phi_{t+1}\rangle - |\Phi_t\rangle\|_2 \leq \mathop{\mathbb{E}}_{\xi\in\sigma}\langle\xi|\rho|\xi\rangle \leq \sqrt{\frac{1+m+\log|X|}{2^{\frac{n}{2}}}}$$

and

$$\mathop{\mathbb{E}}_{\xi\in\sigma}\||\Phi_T\rangle - |\Phi_0\rangle\|_2 \leq \sum_{t=0}^{T-1}\mathop{\mathbb{E}}_{\xi\in\sigma}\||\Phi_{t+1}\rangle - |\Phi_t\rangle\|_2 \leq \sum_{t=0}^{T-1}\mathop{\mathbb{E}}_{\xi\in\sigma}\langle\xi|\rho|\xi\rangle \leq \sum_{t=0}^{T-1}\sqrt{\frac{1+m+\log|X|}{2^{\frac{n}{2}}}} = O\left(T\sqrt{\frac{1+m+n\log\frac{1}{\epsilon}}{2^{\frac{n}{2}}}}\right)$$

Because we want $\||\Phi_T\rangle - |\Phi_0\rangle\|_2$ to be $\Omega(1)$, we need at least $T = \Omega\left(\sqrt{\frac{1+m+n\log\frac{1}{\epsilon}}{2^{\frac{n}{2}}}}\right)$ queries to the oracle.

□

Thus, the separation between QMA and QCMA follows as in [1].

Theorem 4.9. *There exists $U(A)$ such that $\mathsf{QMA}^{U(A)} \neq \mathsf{QCMA}^{U(A)}$.*

We extend Theorem 4.9 to the following conjecture:

Conjecture 4.10. *There exists a classical oracle A that consists of a classical oracle A such that $\mathsf{QMA}^A \neq \mathsf{QCMA}^A$.*

5 Acknowledgements

I would like to thank the SPUR program for providing an opportunity to work on this project and to Professor Jacob Fox, Pavel Etingof and Scott Aaronson for their helpful conversations and suggestions. I especially would like to thank my mentor, Adam Bouland, for his wholehearted help in guidance of research, studying and writing.

References

[1] Scott Aaronson and Greg Kuperberg. Quantum Versus Classical Proofs and Advice. http://theoryofcomputing.org/articles/v003a007/v003a007.pdf

[2] C. Bennett, E.Bernstein, G.Brassard, and U. Vazirani: Strengths and weaknesses of quantum computing. SIAM J. Computing, 26(5):15101523, 1997. quant-ph/9701001. [SICOMP:10.1137/S0097539796300933, arXiv:quant-ph/9701001].

[3] Michael Nielsen and Isaac Chuang. Quantum Computation and Quantum Information. Cambridge University Press.

[4] Sanjeev Arora and Boaz Barak. Computational Complexity: A Moden Approach. Cambridge University Press.