Quantum Computing: Foundations to Frontier Fall 2018

Lecture 7

Lecturer: Henry Yuen *Scribes: Phillip W.K. Jensen, Adrian She, Karan Grewal*

1 Quantum Complexity Theory

Computational complexity theory is the study of the power of various computational resources:

- time
- space
- randomness
- interactivity
- communication
- quantumness
- non-determinism

and how these resources relate to each other. The types of questions you ask about these include:

- Does non-determinism help speed up computations?
- Can any problem that you solve using a small amount of space also be solved using a small amount of time?
- Are quantum computers more powerful than classical computers?

Complexity theory gives precise and formal ways of asking these questions. An important conceptual tool is the idea of *complexity classes*. Complexity classes organize computational problems into groups based on the resources required to solve those problems. For example,

P = polynomial time = {A problem is solvable in polynomial time if it can be solved in n^c time (for some constant $c > 0$ and n is the length of your input) by a deterministic Turing machine (think computer program such as one written in Python)}.

Examples: Sorting, multiplying 2 integers, finding shortest paths between two points in a network, primality testing.

NP = non-deterministic polynomial time = {set of all problems whose solutions you can check in deterministic polynomial time using a classical computer. That is, it is easy to check if the answer is correct, however, it may be nontrivial to get there.}

Examples: Traveling Salesman Problem (TSP), 3-SAT, factoring integers, graph isomorphism, generalized Pokemon.

BPP = bounded error probabilistic polynomial time = {Set of problems you can solve using a randomized computer program in polynomial time, where you have to get the right answer with probability $\geq 2/3$}.

Examples: Any problem in **P**, polynomial identity testing.

PSPACE = polynomial space = {set of all problems solvable using a deterministic computer program that only uses polynomial bits of memory}.

Examples: Any problem in **P**, any problem in **NP**, generalized Donkey Kong, generalized Tic-tac-toe.

EXP = exponential time.

Examples: Any problem in **P**, any problem in **NP**, generalized chess, Go.

Classical complexity landscape:

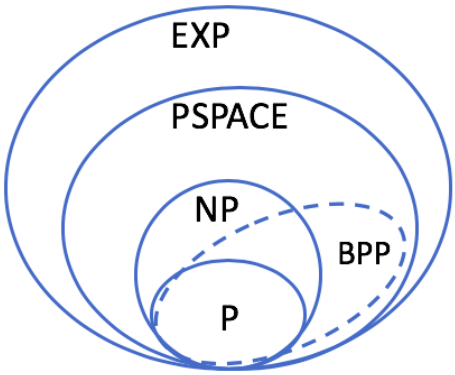


Figure 1: **Classical complexity landscape.** It is conjectured that **BPP** = **P** (in other words, randomization does not allow one to solve problems significantly faster). However, the only proven inclusion among the classes here is that **BPP** is contained in **PSPACE**.

BQP = bounded error quantum polynomial time = {problems that can be solved efficiently on a quantum computer in poly time, and gets correct answer with probability $\geq 2/3$}. Examples: Factoring, simulating quantum systems, computing certain Knot invariants.

The biggest question in quantum complexity theory: Where does **BQP** sit relative to the classical complexity classes?

- Does **BQP** $\subseteq$ **BPP**? We know that **BPP** $\subseteq$ **BQP**.
- Does **NP** $\subseteq$ **BQP**?
- What is the best classical upper bound on **BQP**?

2 Classical Upper Bounds on BQP

We would like to know if how quantum models of computation relate to classical models of computation. One of these questions is the following: are problems solvable using a quantum computer in polynomial time also solvable using a classical computer with access to randomness? As numerous people believe that quantum computers are more powerful than classical computers, this can be expressed in complexity-theoretic terms as the conjecture that **BQP** $\neq$ **BPP** since the whole premise of quantum computation is that it cannot be efficiently simulated by classical computers.

As a step towards relating **BQP** to known classical complexity classes, numerous upper bounds on **BQP** have been proven. This relates to the problem of simulating quantum systems on classical computers.

2.1 BQP is in EXP

Firstly, we will prove that we can simulate any polynomial-time quantum circuit in classical exponential time. That is **BQP** $\subseteq$ **EXP**.

Consider the following computational problem, which we will call CIRCUIT-PROB.

Definition 1 (CIRCUIT-PROB problem)**.** *Suppose we have a description of a quantum circuit $\mathcal{C}$ accepting n qubits as input and m gates, with m a polynomial in n and each acting on one or two qubits. What is the probability that measuring the first qubit in the state $\mathcal{C}|0\rangle^{\otimes n}$ gives the result 1?*

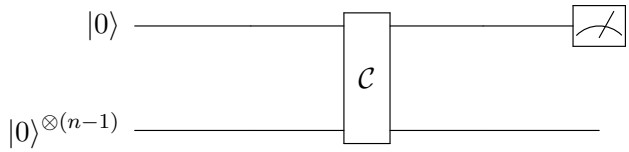


Figure 2: Schematic for the CIRCUIT-PROB problem

To show that **BQP** $\subseteq$ **EXP**, it suffices to show that CIRCUIT-PROB is in **EXP** since every problem in **BQP** may be reduced to CIRCUIT-PROB.

Theorem 2. **BQP** $\subseteq$ **EXP**.

Proof. Let $\mathcal{C}$ be a polynomial sized quantum circuit on n qubits and m gates. Let $|\psi_0\rangle = |0\rangle^{\otimes n}$ and $|\psi_i\rangle$ be the state after the i^{th} gate in the circuit is applied to $|\psi_{i-1}\rangle$. Each state $|\psi_i\rangle$ can be represented in a classical computer as a unit vector in $\mathbb{C}^{2^n}$. Furthermore, $|\psi_i\rangle$ can be computed from $|\psi_{i-1}\rangle$ by matrix-vector multiplication taking $2^{O(n)}$ time. Hence, the final state $|\psi_m\rangle$ can be computed in $m2^{O(n)}$ time, and therefore all together, we have an $2^{O(n)}$ time algorithm for calculating the probability that the first qubit is measured to be one. This implies that **BQP** $\subseteq$ **EXP** by **BQP**-completeness of CIRCUIT-PROB. □

Notice in the above proof that the algorithm used to simulate the given quantum circuit used $O(2^n)$ space. Can we be more space efficient? In fact, we can simulate quantum circuits in polynomial space.

2.2 BQP is in PSPACE

To simulate polynomial sized quantum circuits in polynomial space, we will use physicist Richard Feynman's "sum over histories" construction which he introduced in his study of quantum mechanics. This result was proved by Bernstein and Vazirani in [3].

Sum Over Histories Example. To illustrate this construction, we will revisit one of the first quantum circuit that we studied in this class, illustrated in Figure 3. Since H is a Hadamard gate, we can compute the probability that we measure 0 to be 1 by computing the components of the vector $H^2|0\rangle$.

Figure 3: A quantum circuit

We can view the computation performed by the circuit in Figure 3 as a tree as follows. The first Hadamard gate H takes $|0\rangle$ to the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, so we can represent this as a labelled tree with root label $|0\rangle$, and edges $(|0\rangle, |i\rangle)$ labelled by $\langle i|H|0\rangle$. Call each label an **amplitude**. This is illustrated in Figure 4.

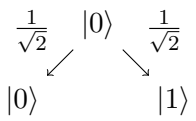


Figure 4: One level of the sum over histories tree

The action of the Hadamard gate H on the states $|0\rangle$ and $|1\rangle$ can then be represented by second level of the tree. This is illustrated in Figure 5.

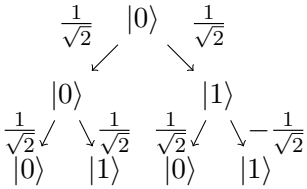


Figure 5: The full sum over histories tree

Using the tree illustrated in Figure 5, we can then calculate the final state $a_0|0\rangle + a_1|1\rangle$ of the quantum circuit. Firstly, consider the two paths from the root to a leaf labelled $|0\rangle$, namely $|0\rangle \to |0\rangle \to |0\rangle$ and $|0\rangle \to |1\rangle \to |0\rangle$. Along either path, we can compute a path amplitude by multiplying the amplitudes along each path. Summing up the amplitudes over both paths yields $a_0 = (\frac{1}{\sqrt{2}})^2 + (\frac{1}{\sqrt{2}})^2 = 1$. Summing up the path amplitudes over all paths leading to $|1\rangle$ yields $a_1 = (\frac{1}{\sqrt{2}})^2 - (\frac{1}{\sqrt{2}})^2 = 0$. Hence, we have recovered that the final state is $|0\rangle$ from this calculation.

Sum Over Histories in General. In general, if $\mathcal{C}$ is a quantum circuit acting on n qubits with m gates, the sum over histories tree will be a tree of depth m, with one level for each gate g_i in addition to the root, and with branching factor 2^n.

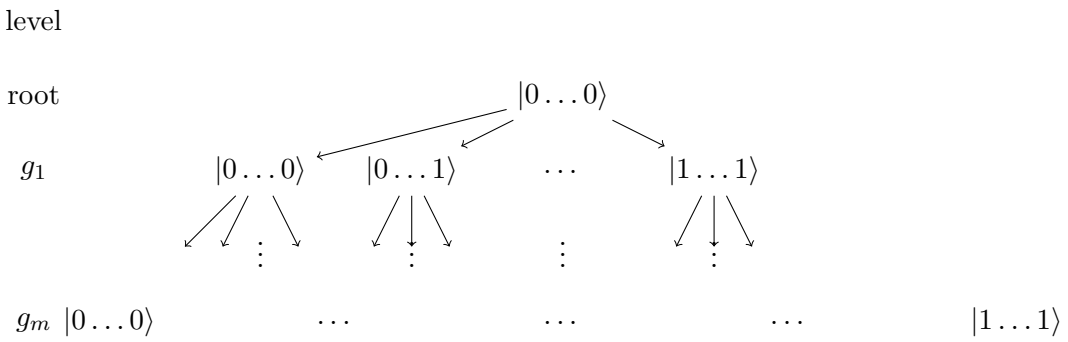


Figure 6: The full sum over histories tree

A **history** is a path in the sum of histories tree. We will denote a history by a sequence $u_0 = |0\rangle^{\otimes n} \to u_1 \to \cdots \to u_{m-1} \to u_m = x$ for some final state x.

Let $u, v \in \{0,1\}^n$. Observe amplitude of the edge $(|u\rangle, |v\rangle)$ in the j^{th} level of the sum over histories tree is $\alpha_j(u \to v) = \langle v|g_j|u\rangle$. Next, for any history, the **transition amplitude of the history** is the product

$$\alpha_1(|0\rangle^n \to u_1)\alpha_2(u_1 \to u_2)\ldots\alpha_m(u_{m-1} \to x). \tag{1}$$

Lemma 3. *Fix a history $u_0 \to \cdots \to u_m$. The transition amplitude of the history is computable in polynomial time.*

Proof. Each gate g_j can be decomposed into $g_j = I \otimes \tilde{g_j}$ for some unitary operator $\tilde{g_j}$ acting on two qubits, which without loss of generality can taken to be the first two. Hence,

$$\langle v|g_j|u\rangle = \langle v_1v_2|\tilde{g_j}|u_1u_2\rangle\langle v_3\ldots v_n|u_3\ldots u_n\rangle,$$

which can be computed in polynomial time in n. Since m is polynomial in n, the transition amplitude of the history can be computed in polynomial time. □

From equation 1, we can then deduce the following formula.

Lemma 4. *Let $\tilde{x} \in \{0,1\}^m$ and $\mathcal{C}|0\rangle^{\otimes n} = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$ be the final state of the quantum circuit. The amplitude $\alpha_{\tilde{x}}$ can be computed by*

$$\alpha_{\tilde{x}} = \sum_{\text{histories } u_0\to\cdots\to u_m} \prod_{i=1}^{m} \alpha_j(u_{j-1} \to u_j)$$

where $u_0 = |0\rangle^{\otimes n}$ and $u_m = |\tilde{x}\rangle$.

Pseudo-code for computing $\alpha_{\tilde{x}}$ for implementing the formula given in Lemma 4 is given in Algorithm 1.

We are now ready to prove that **BQP** $\subseteq$ **PSPACE**.

Algorithm 1 Sum of histories algorithm for computing the amplitude α_x in $\mathcal{C}|0\rangle^{\otimes n} = \sum_x \alpha_x |x\rangle$.

```
amp ← 0
for every history u_0 = |0⟩^{⊗n}, u_1, ..., u_{m-1}, u_m = |x⟩ do
    β ← 1
    for j = 1 to m do
        β ← β × α_j(u_{j-1} → u_j)
    end for
    amp ← amp + β
end for
return amp
```

Theorem 5. BQP $\subseteq$ PSPACE.

Proof. Let $\mathcal{C}$ be a polynomial sized quantum circuit on n qubits and m gates. Suppose $\mathcal{C}|0\rangle^{\otimes n} = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.

Notice in the sum over histories algorithm to compute some amplitude α_x, only one history is stored at any point in the computation. Hence, the sum over histories algorithm uses $O(nm)$ space to compute α_x for any x since $O(nm)$ bits are needed to store the histories in addition to the workspace variables α_j, β, amp to some precision.

Therefore, in polynomial space, we may compute $\sum_x |\alpha_x|^2$ over all x with the first qubit being 1 to compute the probability that the first qubit is measured to be 1 by the end of the circuit. Since we have constructed a polynomial space algorithm for computing the probability of measuring 1 for the first qubit of $\mathcal{C}|0\rangle^{\otimes n}$ (i.e. the CIRCUIT-PROB problem), which we stated earlier that every problem in **BQP** reduces to, **BQP** $\subseteq$ **PSPACE**. □

Notice that compared with the simulation given for the proof that **BQP** $\subseteq$ **EXP**, our algorithm here takes far less space but far more time instead. In fact it takes $O(m2^{mn})$ time to calculate a single amplitude! Finding the right trade-offs between time and space for classical simulation of quantum circuits is a current research topic especially as larger quantum computers get built and need to be verified.

A similar sum-over-histories argument can be used to show that **BQP** $\subseteq$ **PP** (probabilistic polynomial time). The proof of this result can be found in [1]. The *best* upper bound one is that **BQP** $\subseteq$ **AWPP**, which is an (obscure) variant of **PP**.

3 Lower Bounds: evidence that NP is not in BQP

Despite these useful results, we are still interested in the relation between **BQP** and more commonly studied complexity classes such as **P**, **NP**, and **BPP**. To show **BQP** $\neq$ **BPP**, we must prove that there is no fast, clever classical algorithm that for simulating quantum systems that doesn't require exponential overhead. As of present, extremely smart physicists such as Richard Feynman have thought long and hard about this problem, but their inability to reach a conclusion doesn't mean that no such algorithm exists.

As many hypotheses in complexity theory aren't formally proven, we can try to gather evidence that supports our arguments. A widely-held belief is that **NP** is not a subset of **BQP**, that is, quantum computers can't efficiently solve **NP**-complete problems. If this hypothesis is true, it would consequently imply $\mathbf{P} \neq \mathbf{NP}$, since **P** is contained in **BQP**. More concretely, the $\mathbf{NP} \nsubseteq \mathbf{BQP}$ hypothesis states that quantum algorithm can't solve 3 SAT efficiently despite that 3 SAT is **NP**-complete.

To give evidence in favour of our hypothesis, we will show there is an *oracle problem* whose solution can be verified in polynomial time, however the problem itself cannot be solved efficiently. Given black-box access to a function $f : \{0,1\}^n \to \{0,1\}$, the goal is to find $x \in \{0,1\}^n$ such that $f(x) = 1$ if such an x exists. We can only make queries to f. This problem can be solved non-deterministically: if a proposed solution y is provided, one can easily verify if $f(y) = 1$ with a simple query, hence this problem is in $\mathbf{NP}^f$ (i.e., **NP** given oracle access to f). In contrast, exponentially many queries to f are required in the worst case to find a solution using a quantum algorithm.

A quantum algorithm accesses f via the oracle U_f as

$$U_f|x\rangle = (-1)^{f(x)}|x\rangle \qquad \forall x \in \{0,1\}^n.$$

By the results obtained from Grover's search algorithm, $\sqrt{2^n}$ is an upper bound on the number of queries required to solve the problem. We will show that this is the optimal number of queries and any quantum query algorithm that makes fewer than $\sqrt{2^n}$ queries cannot solve the problem. This was also shown to be a lower bound by Bennett, Bernstein, Brassard, and Vazirani [2]. A quantum query algorithm $\mathcal{A}$ that makes T queries is as follows:

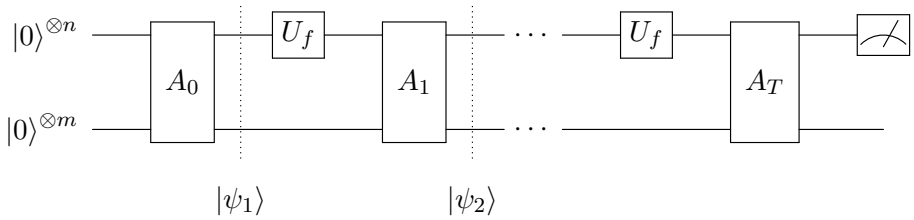


Figure 7: Quantum query algorithm $\mathcal{A}$

where each A_i is a unitary that acts on a $(n+m)$-dimensional qubit and is independent of f. There are a total of $T+1$ A_i gates. Assume $T \ll \sqrt{2^n}$ and the only way $\mathcal{A}$ finds information about f is through a query performed on this quantum circuit.

We now analyze this quantum circuit using a *hybrid argument*. From a high level, we run the algorithm on an input f such that $f(x) = 0$ for all x, and then find $x^* \in \{0,1\}^n$ that causes some other function g (which we will soon define) to output 1 on x^* and 0 everywhere else, but $\mathcal{A}$ can't notice the change. Hence from $\mathcal{A}$'s perspective, $U_f = U_g$, but they are actually quite different.

Let $|\psi_t\rangle$ be the state of the circuit just before the t^{th} query (or directly after applying the A_{t-1} unitary, which is the same). Then,

$$|\psi_t\rangle = \sum_{x,w} \alpha_{x,w,t}|x,w\rangle$$

where the sum is over all $x \in \{0,1\}^n, w \in \{0,1\}^m$. Define *query magnitude* of $x \in \{0,1\}^n$ as

$$M_x = \sum_{t=1}^{T} \sum_{w} |\alpha_{x,w,t}|^2.$$

Intuitively, this is a sum over all queries of the probability that x is queried.

Claim 6. *The sum of query magnitudes over all possible configurations of $x \in \{0,1\}^n$ is T, i.e.,*

$$\sum_{x\in\{0,1\}^n} M_x = T.$$

Proof. Just by using the definition of query magnitude,

$$\sum_{x\in\{0,1\}^n} M_x = \sum_{x\in\{0,1\}^n} \sum_{t=1}^{T} \sum_{x,w} |\alpha_{x,w,t}|^2 = \sum_{t=1}^{T} 1 = T.$$

□

This result thus implies that, on average over x, $M_x \approx T/2^n$. Since it's an average, we can infer $\exists \tilde{x}$ such that $M_{\tilde{x}} \leq T/2^n$. Consequently, the total weight of the queries on $\tilde{x}$ is $T/2^n$. Also, since $T \ll \sqrt{2^n}$, we get that $\tilde{x} \ll 1/\sqrt{2^n}$ and $\mathcal{A}$ doesn't really "pay attention" to $\tilde{x}$. We can use this fact and change the U_f oracles in our quantum algorithm one-by-one until they all become U_g. Define $g : \{0,1\}^n \to \{0,1\}$ as

$$g(x) = \begin{cases} 1 & \text{if } x = \tilde{x} \\ 0 & \text{otherwise} \end{cases}.$$

After this change, the state of $\mathcal{A}$ using all U_g oracles instead of U_f will be close. The hybrid approach structure proceeds as follows:

$\mathcal{H}_0$: Let $|\psi_T^{(0)}\rangle$ be the state of $\mathcal{A}$ after the T^{th} query

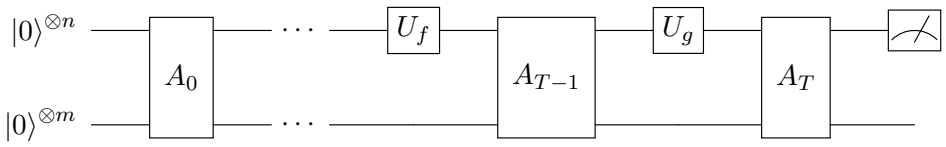
$\mathcal{H}_1$: Let $|\psi_T^{(1)}\rangle$ be the state of $\mathcal{A}$ after the T^{th} query where we replace the last U_f query with U_g

$\mathcal{H}_2$: Let $|\psi_T^{(2)}\rangle$ be the state of $\mathcal{A}$ after the T^{th} query where we replace the last two U_f queries with U_g

$\vdots$

$\mathcal{H}_T$: Let $|\psi_T^{(T)}\rangle$ be the state of $\mathcal{A}$ after the T^{th} query where we replace all U_f queries with U_g

For example, the quantum circuit corresponding to $\mathcal{H}_1$ is illustrated here:



Claim 7.

$$\sum_{t=1}^{T} \sqrt{\sum_{w} |\alpha_{\tilde{x},w,t}|^2} \leq \frac{T}{\sqrt{2^n}}.$$

Proof. Making use of the Cauchy-Schwarz inequality,

$$\sum_{t=1}^{T}\sqrt{\sum_{w}|\alpha_{\tilde{x},w,t}|^2} \leq \sqrt{\left(\sum_{t=1}^{T}1\right)\left(\sum_{t}\sum_{w}|\alpha_{\tilde{x},w,t}|^2\right)} = \sqrt{TM_x} \leq \frac{T}{2^n}.$$

□

Our goal now is to show that $|\psi_{T+1}^{(0)}\rangle$ and $|\psi_{T+1}^{(T)}\rangle$ are very "close", which will ultimately show $\mathcal{A}$ treats the U_f and U_g oracles the same since their outputs are close. First, we establish that

$$\begin{aligned}
|\psi_T^{(0)}\rangle &= \sum_{x,w}\alpha_{x,w,T}|x,w\rangle, \\
|\psi_T^{(1)}\rangle &= \sum_{\tilde{x}\neq x,w}\alpha_{x,w,T}|x,w\rangle - \sum_{x,w}\alpha_{\tilde{x},w,T}|\tilde{x},w\rangle,
\end{aligned}$$

and therefore

$$\begin{aligned}
\left\||\psi_T^{(0)}\rangle - |\psi_T^{(1)}\rangle\right\| &= \left\|A_T U_f|\psi_{T-1}^{(0)}\rangle - A_T U_g|\psi_{T-1}^{(0)}\rangle\right\| \\
&= \left\|U_f|\psi_{T-1}^{(0)}\rangle - U_g|\psi_{T-1}^{(0)}\rangle\right\| \qquad \text{(since unitaries preserve distance)} \\
&= 2\left\|\sum_{w}\alpha_{\tilde{x},w,T-1}|\tilde{x},w\rangle\right\| \\
&= 2\sqrt{\sum_{w}|\alpha_{\tilde{x},w,T-1}|^2}.
\end{aligned}$$

This argument can be extended via induction to show that

$$\begin{aligned}
\left\||\psi_T^{(k)}\rangle - |\psi_T^{(k+1)}\rangle\right\| &= \left\||\psi_{T-k}^{(k)}\rangle - |\psi_{T-k}^{(k+1)}\rangle\right\| \\
&= 2\sqrt{\sum_{w}|\alpha_{\tilde{x},w,T-k}|^2}.
\end{aligned}$$

for all k. We are now ready to prove our final result. Using claims 6 and 7,

$$\begin{aligned}
\left\||\psi_T^{(0)}\rangle - |\psi_T^{(T)}\rangle\right\| &\leq \sum_{t=0}^{T-1}\left\||\psi_T^{(t)}\rangle - |\psi_T^{(t+1)}\rangle\right\| \qquad \text{(by triangle inequality)} \\
&\leq 2\sum_{t=0}^{T-1}\sqrt{\sum_{w}|\alpha_{\tilde{x},w,T-t}|^2} \\
&\leq 2\frac{T}{\sqrt{2^n}}.
\end{aligned}$$

What is happening here? Intuitively, the angle between $|\psi_T^{(0)}\rangle$ and $|\psi_T^{(T)}\rangle$ is so small that $\mathcal{A}$ cannot tell the difference between f and g unless $T \approx \sqrt{2^n}$. We have thus shown that $\sqrt{2^n}$ is a lower bound on the number of queries required. Although this is not a formal proof that $\mathbf{NP} \not\subseteq \mathbf{BQP}$, it is nevertheless evidence in favour of the hypothesis.

References

[1] L. Adleman, J. DeMarrais, and M. Huang. *Quantum computability*, SIAM Journal on Computing 26:1524-1540, 1997.

[2] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. *Strengths and weaknesses of quantum computation*, SIAM Journal on Computing, 1997.

[3] E. Bernstein and U. Vazirani. *Quantum complexity theory*, SIAM Journal on Computing, 26(5):1411-1473, 1997.