Week 7

Other Methods

This week, we will briefly go through some more obscure lower bound techniques for quantum query complexity. For some of them, only references will be provided without much explanation; students may pursue whichever of these papers seem interesting.

7.1 Lower Bounds by Upper Bounds

One way of proving lower bounds is by proving upper bounds, that is, by providing algorithms for other problems. This sounds surprising at first, but it's actually a familiar idea: it usually goes by the name "reduction". If you already know that problem A is hard, then in order to show that problem B is hard, it suffices to give an algorithm for A that uses a solver (usually called an *oracle*) for B as a subroutine; since we know A cannot be solved quickly, we can conclude B cannot be solved quickly either.

In computer science, reductions are usually extremely simple: they involve modifying the input so that yes-inputs to A become yes-inputs to B, and no-inputs to A become no-inputs to B. This style of reductions (sometimes called Karp reductions) are the ones often encountered in NP-hardness proofs.

In this section we will talk about reductions that are extremely involved: the reductions themselves are complex quantum algorithms. They feel more like "algorithms" than like "reductions", and hence the lower bounds have a weird feeling where all the hard work came from coming up with an algorithm. Such proofs are sometimes called "ironic" lower bounds.

There are two main examples of ironic quantum lower bounds. One, first used by [CVN+98], uses the quantum algorithm of Bernstein-Vazirani [BV97], and the other, first used by [BBG+18], uses a quantum algorithm of Belovs [Bel15].

7.1.1 The first ironic lower bound

We will start by describing the Bernstein-Vazirani problem. For each $x \in \{0,1\}^n$, define $y_x \in \{0,1\}^{2^n}$ to be the string where $(y_x)_S = \textsc{Parity}(x_S)$ for each $S \subseteq [n]$. That is to say, the exponentially long string y_x will be indexed by subsets $S \subseteq [n]$ (there are 2^n of them). At the index corresponding to S, the bit $(y_x)_S$ will equal the parity bits x_i of x for $i \in S$, or in other words, the sum modulo 2 of x_i over $i \in S$. The Bernstein-Vazirani problem is the function f defined on the promise set $P = \{y_x : x \in \{0,1\}^n\} \subseteq \{0,1\}^{2^n}$, which is defined by $f(y_x) = x$. That is, $f\colon P \to \{0,1\}^n$ has Boolean input alphabet but has non-Boolean outputs (it outputs an entire n-bit string when given a 2^n-bit string as input).

Classically, one could compute f by querying $(y_x)_{\{i\}}$ for each $i \in [n]$, which gives the bits x_i one at a time; this requires n queries to produce x (note that this is logarithmic in the input size, since the input size is 2^n). Surprisingly, Bernstein and Vazirani showed that $\mathrm{Q}(f) = 1$; that is, a quantum query algorithm can extract n bits of information using a single query.

The observation of [CVN+98] was that the Bernstein-Vazirani algorithm can be used to show a lower bound on parity. This works as follows: Suppose we had a fast quantum algorithm A for computing parity. We will build a quantum algorithm B which takes in an input $x \in \{0,1\}^n$ and outputs $x \in B^n$ (i.e. it extracts all the bits of the input). B will work by running Bernstein-Vazirani, and whenever the Bernstein-Vazirani algorithm tries to query $(y_x)_S$, B will call the algorithm A as a subroutine to compute the parity of the bits of x that are indexed by S. B will do all of this in superposition.

Now, since Bernstein-Vazirani makes only 1 query in superposition, the number of queries B makes turns out to be the maximum number of the number of queries the subroutine A makes for computing the parity of the bits of x indexed by some subset S. If A computed parity too fast, say $o(n)$, then B would use $o(n)$ quantum queries for the task of reading the whole input (that is, for computing the identity function $\textsc{Id}$). In other words, we have shown $\mathrm{Q}(\textsc{Id}_n) = O(\mathrm{Q}(\textsc{Parity}_n))$. To give a lower bound on $\mathrm{Q}(\textsc{Parity})$, it therefore suffices to lower bound $\mathrm{Q}(\textsc{Id})$.

Now, admittedly, lower bounding $\mathrm{Q}(\textsc{Parity})$ directly is not too hard (it can be done using a simple symmetrization argument using the polynomial method, without even the need for the Markov brothers' inequality or any other fancy tools; it can also be done using the simplest form of the positive-weight adversary method). And lower bounding $\mathrm{Q}(\textsc{Id})$ is something that itself requires some argumentation. So it's not clear what has been gained.

However, the algorithmic reduction from identity to parity turns out to work in a wide variety of models. First, one could take the quantum algorithm for Bernstein-Vazirani and convert it into a polynomial; then, assuming one had a low-degree polynomial for parity, one would get a low-degree system of polynomials for $\textsc{Id}$. Therefore this approach can give an $\widetilde{\deg}(\textsc{Parity})$ lower bound, not merely a $\mathrm{Q}(\textsc{Parity})$ lower bound.

But this is just the beginning. We can also consider computing the function $\textsc{Parity} \circ g$ for some Boolean function g. Using the Bernstein-Vazirani algorithm, if we can compute $\textsc{Parity} \circ g$ quickly, we can also compute $\textsc{Id} \circ g$ quickly, which means we can compute n independent copies of g. This approach (with some additional work to lower bound the quantum query complexity of computing n copies of g) can be used to show that $\mathrm{Q}(\textsc{Parity}_n \circ g) = \Omega(n\,\mathrm{Q}(g))$. Of course, we already knew that from the negative-weight adversary bound, but that was a highly nontrivial technique.

We can even do this for polynomial degree: we can reduce the task of showing $\widetilde{\deg}(\textsc{Parity}_n \circ g) = \Omega(n\,\widetilde{\deg}(g))$ to the task of showing a so-called "direct sum" theorem for polynomial degree (though that itself is not trivial).

Next, the whole approach even works in communication complexity, where lower bounds are generally much harder to prove. As before, the Bernstein-Vazirani algorithm only provides a *reduction*, so to get a lower bound for different versions of parity we must start with lower bounds for idenitity, but in communication complexity the latter are easier.

7.1.2 The second ironic lower bound

The second ironic lower bound works similarly to the first, but uses a much more complicated quantum algorithm. The setup of the algorithm is similar, except with OR instead of $\textsc{Parity}$. That is, for each $x \in \{0,1\}^n$, let $z_x \in \{0,1\}^{2^n}$ be the string defined by $(z_x)_S = \bigvee_{i \in S} x_i$ (the OR of the bits of x indexed by S, instead of the $\textsc{Parity}$ as in the string y_x). The task called "combinatorial group testing" is the function $f \colon P \to \{0,1\}^n$ where $P = \{z_x : x \in \{0,1\}^n\}$ and $f(z_x) = x$. In other words, the goal is to extract the string x using subset-OR queries.

Belovs [Bel15] showed that $\mathrm{Q}(f) = \Theta(\sqrt{n})$, while classically this task requires n queries. This can be used to prove a lower bound on OR_n: Using Belov's algorithm, if we had a fast algorithm A for OR, then we could get a fast algorithm B for $\textsc{Id}$ by running Belov's algorithm, and whenever it wants to query a subset S, run A to compute $\mathrm{OR}(\{x_i : i \in S\})$. Since Belov's algorithm uses $O(\sqrt{n})$ queries, we get that $\mathrm{Q}(\textsc{Id}_n) = O(\sqrt{n}\,\mathrm{Q}(\mathrm{OR}_n))$, or in other words, $\mathrm{Q}(\mathrm{OR}_n) = \Omega(\mathrm{Q}(\textsc{Id}_n)/\sqrt{n})$.

As before, this generalizes to other settings. The same argument gives $\mathrm{Q}(\mathrm{OR}_n \circ g) = \Omega(\mathrm{Q}(\textsc{Id}_n \circ g)/\sqrt{n})$. By converting Belov's algorithm into a polynomial, we can also get $\widetilde{\deg}(\mathrm{OR}_n \circ g) = \Omega(\widetilde{\deg}(\textsc{Id}_n \circ g)/\sqrt{n})$, which has been used to show $\widetilde{\deg}(\mathrm{OR}_n \circ g) = \Omega(\sqrt{n}\,\widetilde{\deg}(g))$ (this is the only known proof of the latter statement – proving a lower bound on polynomials required using a quantum algorithm!) We also get similar results in communication complexity.

As a final note, it is worth remarking on how Belovs constructed his algorithm. Instead of designing a quantum algorithm from scratch, he used the primal form of the negative-weight adversary to upper bound $\mathrm{Adv}^{\pm}(f)$ by explicitly giving a primal solution to the minimization problem. This means that not only are these lower bounds on the approximate degree non-constructive, they rely on a quantum algorithm which is itself non-constructive!

7.2 The Multiplicative Adversary Method

A strange variant of the quantum adversary is something called the *multiplicative* quantum adversary. This technique is primarily useful for proving lower bounds on quantum algorithms that make small (but not necessarily exponentially small) bias. For example, if you want a lower bound on $\mathrm{Q}_{1/2-1/n}(f)$ for some function f, the multiplicative quantum adversary method would be a reasonable technique to try (though it may make sense to try the polynomial method first).

The multiplicative quantum adversary method is generally hard to use. There are only a handful of papers that ever used or studied it: [Špa08; AMR+11; LR13; MR15]. It is substantially uglier than even the negative-weight adversary method.

(Recall that the negative-weight adversary method is only tight up to constant factors for *bounded error* quantum algorithms; it does not give good bounds for small-bias quantum algorithms. The multiplicative weight adversary can sometimes give good lower bounds for small-bias or small-success-probability algorithms, though it is not known whether this method is always tight.)

The main claim to fame of the multiplicative adversary method is that it was necessary in the proof of the direct product theorem for quantum query complexity [LR13]. This is the following theorem.

Theorem 7.1. *Let f be a (possibly partial) Boolean function. Then for all k, we have*

$$\mathrm{Q}_{1-0.9^k}(\textsc{Id}_k \circ f) = \Omega(k\,\mathrm{Q}(f)).$$

In this theorem, $\mathrm{Q}_{1-0.9^k}$ denotes the quantum query complexity with error $1 - 0.9^k$, or in other words success probability 0.9^k. This is a little strange, because usually we require the error probability to be less than $1/2$, since guessing randomly without reading the input is already sufficient to achieve error $1/2$. However, that's only true for functions that have Boolean outputs; here, we are analyzing $\mathrm{Q}_{1-0.9^k}(\textsc{Id}_k \circ f)$, which is the task of solving k copies of the function f on k different copies of the input. In this task, guessing randomly only gives success probability of $0.5^k < 0.9^k$.

What this theorem is saying is that not only does computing k copies of f require $\Omega(k\,\mathrm{Q}(f))$ quantum queries, but even getting the answer to k copies of f right with tiny probability (0.9^k instead of the 0.5^k of random guessing) already requires $\Omega(k\,\mathrm{Q}(f))$ quantum queries. Such "direct product" theorems are quite tricky to prove, even for randomized algorithms, though a version of Theorem 7.1 is known for randomized algorithms too (using completely different techniques; see [Dru12]). Note that Theorem 7.1 cannot be shown using the negative-weight adversary bound, since that bound is only tight for bounded error quantum algorithms; it will only give the weaker statement $\mathrm{Q}(\textsc{Id}_k \circ f) = \Omega(k\,\mathrm{Q}(f))$.

7.3 Zhandry's Quantum Query Lower Bounds

Quantum computing researcher Mark Zhandry developed some lower bound techniques that have not been widely adopted. He has successfully used them to prove several nontrivial quantum lower bounds.

In [Zha13], Zhandry used a tool he developed in [Zha12] to prove a lower bound on a problem called Set-Equality. This problem is defined on a subset of $[n]^n$. The n bits of the input are divided into two parts of size $n/2$ each (we assume n is even), and it is promised that each part contains no duplicate symbols. It is further promised that either these two "sets" of symbols (specified by the parts of size $n/2$ of the input) are either identical or disjoint. The task is to determine which: are the sets identical, or are they disjoint?

It is not hard to see that Set-Equality is a special case of Collision; indeed, it is the same function as Collision, just restricted to the promise that if the input contains $n/2$ symbols twice each, then each symbol occurring in the input occurs once in the first $n/2$ bits of the input and once in the last $n/2$ bits of the input.

Since restricting to a promise can only make a problem easier, we know that $\mathrm{Q}(\textsc{Set-Equality}) = O(n^{1/3})$, since there is an $O(n^{1/3})$-query quantum algorithm for Collision. However, we do not immediately know anything about lower bounds for Set-Equality, since it may be easier than Collision. It turns out that such lower bounds are difficult to prove.

Zhandry proved the first lower bound on Set-Equality using his techniques, which involve a different way of getting a polynomial out of a quantum algorithm. I do not know whether Zhandry's proof can be converted to a true polynomial method lower bound, that is, to a lower bound on $\widetilde{\deg}(\textsc{Set-Equality})$.

A different method of Zhandry's was developed in [Zha19], and used in [LZ19] to give lower bounds for problems such as k-sum, which is a problem that has only been previously lower bounded by the negative-weight adversary method. Once again, it is not clear whether Zhandry's techniques can be used to settle the approximate degree of k-sum, which has been a long-standing open problem.

7.4 Lower bounds in other models

Apart from the query lower bounds we have studied, there are also quantum lower bounds in other models. We will focus the rest of the course on one such model, called communication complexity. Here are a few others we won't cover:

1. Quantum information results. Quantum information is an entire field, on which a book has recently been written [Wat18]. It deals with how much information can be stored in qubits or transmitted across quantum channels.

2. Some lower bounds in quantum cryptography are unconditional "information theoretic" lower bounds, or lower bounds relative to an oracle. These often combine quantum information techniques, properties such as no cloning, and quantum query lower bounds. See [BS16] for a survey of quantum cryptography.

3. Some lower bounds work in the a modified query setting where in addition to making queries to an input x, we are also given some side information about x in the form of a quantum state. Two examples of such lower bounds are [AC12; AKK+19].

4. Finally, there are some remaining quantum query lower bounds we won't cover, especially ones concerning average-case lower bounds rather than worst-case ones. Proving that computing f is hard even when the inputs are generated from a specific, known distribution (such as the uniform distribution) is more difficult than proving a lower bound on the worst-case complexity $\mathrm{Q}(f)$ of f, but it often involves ad hoc techniques that depend on the distribution in question, so is somewhat less interesting to us in this course. Still, such lower bounds are fairly common, especially in cryptography; see some of Zhandry's work and references therein for examples. See also [BBH+17] for a version of the negative-weight adversary bound that can be used to prove average-case lower bounds.

References

[AC12] Scott Aaronson and Paul Christiano. "Quantum money from hidden subspaces". In: *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. 2012 (p. 5).

[AKK+19] Scott Aaronson, Robin Kothari, William Kretschmer, and Justin Thaler. "Quantum lower bounds for approximate counting via Laurent polynomials". In: *arXiv preprint arXiv:1904.08914* (2019) (p. 5).

[AMR+11] Andris Ambainis, Loïck Magnin, Martin Roetteler, and Jérémie Roland. "Symmetry-assisted adversaries for quantum state generation". In: *2011 IEEE 26th Annual Conference on Computational Complexity*. 2011 (p. 3).

[BBG+18] Shalev Ben-David, Adam Bouland, Ankit Garg, and Robin Kothari. "Classical lower bounds from quantum upper bounds". In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. 2018 (p. 1).

[BBH+17] Aleksandrs Belovs, Gilles Brassard, Peter Hoyer, Marc Kaplan, Sophie Laplante, and Louis Salvail. "Provably secure key establishment against quantum adversaries". In: *arXiv preprint arXiv:1704.08182* (2017) (p. 5).

[Bel15] Aleksandrs Belovs. "Quantum algorithms for learning symmetric juntas via the adversary bound". In: *computational complexity* 24.2 (2015) (pp. 1, 3).

[BS16] Anne Broadbent and Christian Schaffner. "Quantum cryptography beyond quantum key distribution". In: *Designs, Codes and Cryptography* 78.1 (2016) (p. 4).

[BV97] Ethan Bernstein and Umesh Vazirani. "Quantum complexity theory". In: *SIAM Journal on computing* 26.5 (1997) (p. 1).

[CVN+98] Richard Cleve, Wim Van Dam, Michael Nielsen, and Alain Tapp. “Quantum entanglement and the communication complexity of the inner product function”. In: *NASA International Conference on Quantum Computing and Quantum Communications*. 1998 (pp. 1, 2).

[Dru12] Andrew Drucker. “Improved direct product theorems for randomized query complexity”. In: *Computational Complexity* 21.2 (2012). DOI: 10.1007/s00037-012-0043-7 (p. 4).

[LR13] Troy Lee and Jérémie Roland. “A strong direct product theorem for quantum query complexity”. English. In: *computational complexity* 22.2 (2013). DOI: 10.1007/s00037-013-0066-8. URL: http://dx.doi.org/10.1007/s00037-013-0066-8 (p. 3).

[LZ19] Qipeng Liu and Mark Zhandry. “On finding quantum multi-collisions”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. 2019 (p. 4).

[MR15] Loïck Magnin and Jérémie Roland. “Explicit relation between all lower bound techniques for quantum query complexity”. In: *International Journal of Quantum Information* 13.04 (2015) (p. 3).

[Špa08] Robert Špalek. “The multiplicative quantum adversary”. In: *2008 23rd Annual IEEE Conference on Computational Complexity*. 2008 (p. 3).

[Wat18] John Watrous. *The theory of quantum information*. Cambridge University Press, 2018 (p. 4).

[Zha12] Mark Zhandry. “How to construct quantum random functions”. In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. 2012 (p. 4).

[Zha13] Mark Zhandry. “A note on the quantum collision and set equality problems”. In: *arXiv preprint arXiv:1312.1027* (2013) (p. 4).

[Zha19] Mark Zhandry. “How to record quantum queries, and applications to quantum indifferentiability”. In: *Annual International Cryptology Conference*. 2019 (p. 4).