LECTURE NOTES ON APPLICATIONS OF GROTHENDIECK'S INEQUALITY

QUANTUM QUERY ALGORITHMS

JOP BRIËT

Abstract. In this lecture we shall cover a result of Aaronson et al. [AAI+16] that chacterizes one-query quantum algorithms in terms of quadratic polynomials. In particular, we shall see an extremely short proof of this result from [ABP18] based on the factorization version of Grothendieck's inequality.

1. Quantum query complexity

In the black-box model of quantum computation one has access to a unitary operation, referred to as an oracle, that allows one to probe the bits of an unknown string $x \in \{-1,1\}^n$ in superposition. The goal in this model is to learn some property of x given by a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, when only given access to x through the oracle. An application of the oracle is referred to as a *query*.

A quantum algorithm works by evolving the state of a quantum register with queries interlaced with input-independent unitary operations. To produce an output after making some number of queries, the algorithm performs a two-outcome measurement and returns the result. (We review a few more details in the next section). Such algorithms are inherently random for two reasons. First, the algorithm may use randomness in selecting which input-independent unitaries to use in between queries. Second, the measurement outcomes be random, depending on which state is measured.

Definition 1.1. Let $\varepsilon \in (0,1/2)$. The bounded-error quantum query complexity of $f : \{-1,1\}^n \to \{-1,1\}$, denoted $Q_\varepsilon(f)$, is the minimal number of queries a quantum algorithm must make on the worst-case input $x \in \{-1,1\}^n$ to compute $f(x)$ with probability at least $1-\varepsilon$.

More generally, this notion is defined for *partial functions*, for which one only cares about inputs from a known subset of $\{0,1\}^n$. For simplicity we will only consider "total" functions here. Determining the query complexity of a function is unfortunately notoriously hard in general.

2. The polynomial method

A useful tool for proving lower bounds comes from a beautiful connection with polynomials, shown by Beals et al. [BBC+01]. In the following all polynomial are assumed to be real and n-variate. A polynomial p is *bounded* if it satisfies $p(x) \in [-1,1]$ for all $x \in \{-1,1\}^n$.

Lemma 2.1 (Beals et al.). *For every t-query quantum algorithm $\mathcal{A}$ that on input $x \in \{-1,1\}^n$ returns a random sign $\mathcal{A}(x)$, there exists a bounded degree-$(2t)$ polynomial p such that $p(x) = \mathbb{E}[\mathcal{A}(x)]$ for every x (where the expectation is taken over the randomness of the output and the algorithm).*

It follows that if $\mathcal{A}$ computes $f : \{-1,1\}^n \to \{-1,1\}$ with probability at least $1-\varepsilon$, then p satisfies $|p(x) - f(x)| \leq 2\varepsilon$ for every x. Indeed, if $\mathcal{A}$ computes f with error δ on x, then

$$p(x) = \mathbb{E}[\mathcal{A}(x)] = (1-\delta)f(x) - \delta f(x) = (1-2\delta)f(x).$$

To prove lower bounds on $Q_\varepsilon(f)$, one can thus instead try lower bounding the smallest degree of a bounded polynomial p that for all inputs x satisfies $|p(x) - f(x)| \leq 2\varepsilon$. The minimal degree of such a polynomial, denoted $\deg_\varepsilon(f)$, is called the *approximate (polynomial) degree* of f. This approach, known as the *polynomial method*, has been used with great success for a large number of functions.

3. A converse to the polynomial method

If Lemma 2.1 had a converse, we would get a succinct characterization of quantum algorithms in terms of polynomials. However, Ambainis [Amb06] proved that this not the case, showing that for infinitely many n, there is a function $f : \{-1,1\}^n \to \{-1,1\}$ with $\deg_{1/3}(f) \leq n^\alpha$ and $Q_{1/3}(f) \geq n^\beta$ for some constants $\beta > \alpha > 0$. Too bad. But this result does leave open the possibility that *constant-degree* polynomials characterize constant-query quantum algorithms. And indeed, Aaronson et al. [AAI+16] showed the following surprising result.

Theorem 3.1 (Aaronson et al.)**.** *For every bounded quadratic polynomial p, there exists a one-query quantum algorithm that, on input $x \in \{-1,1\}^n$, returns a random sign with expectation $Cp(x)$, where $C \in (0,1]$ is an absolute constant.*

In this lecture we will see a very short of this fact based on the factorization version of Grothendieck's inequality.[1] Before going into this proof, we first we review some basics of quantum query algorithms.

4. The quantum query model

A quantum query algorithm is given by a triple of complex vector spaces $(\mathsf{A}, \mathsf{Q}, \mathsf{W})$ and a set of unitary operators on $\mathsf{V} = \mathsf{A} \otimes \mathsf{Q} \otimes \mathsf{W}$. The three spaces have the following specifications and interpretations:

- $\mathsf{A} = \mathbb{C}^2$ represents an *auxiliary register* enabling "controlled" operations (defined below).
- $\mathsf{Q} = \mathbb{C}^n$ represents a *query register* on which input-dependent operations are performed.
- $\mathsf{W} = \mathbb{C}^d$ (for some $d \in \mathbb{N}$) is a *workspace register*, which is affected nontrivially only by input-independent operations.

The space V represents the physical system on which the algorithm does its computations. The set of (pure) states that the system can be in is formed by the set of unit vectors in V. Computations are done via unitary operations on V and $\{-1,1\}$-valued measurements on A.

Such a measurement does the following. Suppose the system is in state

$$\psi = e_0 \otimes \phi_0 + e_1 \otimes \phi_1 \ \in \mathsf{V}, \tag{1}$$

where $e_0, e_1 \in \mathsf{A}$ denote the standard basis vectors for $\mathbb{C}^2$ and ϕ_0, ϕ_1 belong to $\mathsf{Q} \otimes \mathsf{W}$. Then, a $\{-1,1\}$-valued measurement on A produces a random sign whose expected value equals $\|\phi_0\|_2^2 - \|\phi_1\|_2^2$.

Given a unitary operator U on $\mathsf{Q} \otimes \mathsf{W}$, let its *controlled* version be the unitary operator on V defined by

$$\begin{aligned} e_0 \otimes \phi &\mapsto e_0 \otimes \phi \\ e_1 \otimes \phi &\mapsto e_1 \otimes (U\phi). \end{aligned}$$

[1] The original proof presented in [AAI+16] also used this, but only as a lemma in a more intricate argument.

An input $x \in \{-1,1\}^n$ is represented by the unitary operator on Q given by the $n \times n$ diagonal matrix with diagonal x, denoted $\mathrm{Diag}(x)$. A *query* to x is then made by either performing $I_{\mathsf{A}} \otimes \mathrm{Diag}(x) \otimes I_{\mathsf{W}}$ or the controlled version of $\mathrm{Diag}(x) \otimes I_{\mathsf{W}}$ on V, where $I_{\mathsf{A}}, I_{\mathsf{W}}$ are identity operators on A, W, respectively.[2]

A t-query quantum algorithm begins by initializing V in the first standard basis vector (a.k.a. the all-zero state) and continues by interleaving a sequence of unitaries $U_0, \dots, U_t$ on V with queries on (A, Q). Finally, the algorithm performs a measurement on A (see figure 1).

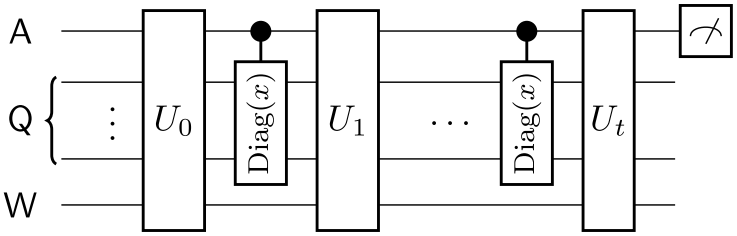

Figure 1. A t-query quantum algorithm that starts with the all-zero state, interlaces controlled queries to $x \in \{-1,1\}^n$ with fixed unitaries $U_0, \dots, U_t$ and concludes by measuring the register A.

Important unitary operations on $\mathbb{C}^2$ are the *Hadamard* and *bit-flip*:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \text{and} \qquad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

A measurement is said to be performed in the Hadamard basis if, prior to the measurement, a Hadamard is performed on A. A moment's reflection shows that if the state (1) is measured in the Hadamard basis, then the expected outcome equals $2\Re\langle \phi_0, \phi_1 \rangle$.

5. Proof of Theorem 3.1

Let us recall the factorization version of Grothendieck's inequality.

Theorem 5.1 (Grothendieck)**.** *For every positive integer n and matrix $A \in \mathbb{R}^{n\times n}$, there exist positive unit vectors $u, v \in \mathbb{R}_{>0} \cap S^{n-1}$ such that the matrix*

$$B = \frac{1}{K_G}\,\mathrm{Diag}(u)^{-1} A\,\mathrm{Diag}(v)^{-1} \tag{2}$$

satisfies $\|B\| \le \|A\|_{\ell_\infty \to \ell_1}$.

[2] Since these act trivially on W, queries are usually said to act only on (A, Q).

It turns out that it is sufficient to prove Theorem 3.1 for the case where the polynomial p is a quadratic form (see [AAI+16]).

Lemma 5.2. *There exists an absolute constant $C \in (0,1]$ such that the following holds. For any bounded quadratic polynomial p, there exists a matrix $A \in \mathbb{R}^{(n+1)\times(n+1)}$ with $\|A\|_{\ell_\infty\to\ell_1} \leq 1$, such that the quadratic form $q(y) = y^\mathsf{T} Ay$ satisfies $q((x,1)) = Cp(x)$ for all $x \in \{-1,1\}^n$.*

Proof of Theorem 3.1 (sketch): By Lemma 5.2 it suffices to prove the statement for a quadratic form $p(x) = x^\mathsf{T} Ax$ given by some matrix $A \in \mathbb{R}^{n\times n}$ such that $\|A\|_{\ell_\infty\to\ell_1} \leq 1$. Theorem 5.1 gives unit vectors u, v such that the matrix B as in (2) has operator norm at most 1. Unitary matrices have norm exactly 1 and of course represent the type of operation a quantum algorithm can implement. Moreover, since u, v are unit vectors, they represent valid states of a quantum system.

Observe that for $w, z \in \mathbb{R}^n$, we have $\mathrm{Diag}(w)z = \mathrm{Diag}(z)w$. It follows that we get the following "factorization formula":

$$\frac{x^\mathsf{T} Ax}{K_G} \overset{(2)}{=} x^\mathsf{T}\,\mathrm{Diag}(u)B\,\mathrm{Diag}(v)x = u^\mathsf{T}\,\mathrm{Diag}(x)B\,\mathrm{Diag}(x)v. \tag{3}$$

If we assume for the moment that the matrix B actually is unitary, then the right-hand side of (3) suggests the simple one-query quantum algorithm described in Figure 2.

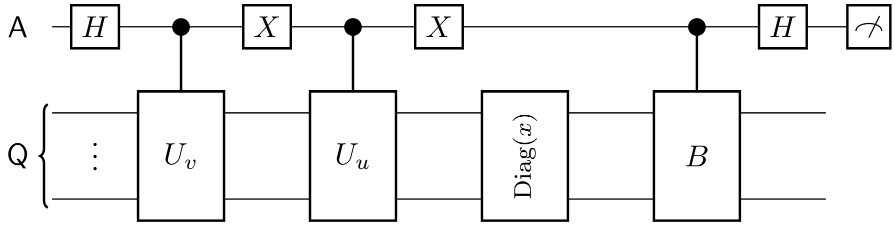

Figure 2. Let $U_u, U_v \in \mathbb{C}^{n\times n}$ be unitaries that have u, v as their first columns, respectively. The algorithm initializes a $2n$-dimensional register (A, Q) in the all-zero state, transforms this state into the superposition $\frac{1}{\sqrt{2}}(e_0 \otimes u + e_1 \otimes v)$, queries the input x via the (uncontrolled) unitary $\mathrm{Diag}(x)$ applied to Q, applies a controlled-B, and finishes by measuring A in the Hadamard basis.

Using (3), it can be shown that expected output of the algorithm is

$$\big\langle \mathrm{Diag}(u)x, B\,\mathrm{Diag}(v)x\big\rangle = \frac{x^\mathsf{T} Ax}{K_G} = p(x)/K_G, \tag{4}$$

giving Theorem 3.1 with $C = 1/K_G$.

In the general case where B is not necessarily unitary, we can use the fact that, by the Russo–Dye Theorem and Carathéodory's Theorem, B is a convex combination of at most $n^2 + 1$ unitaries. The algorithm can thus use randomness to effect B on expectation. □

6. Exercises

Exercise 6.1. Show that (4) is indeed the expected output.

References

[AAI+16] Scott Aaronson, Andris Ambainis, Janis Iraids, Martins Kokainis, and Juris Smotrovs. Polynomials, quantum query complexity, and Grothendieck's inequality. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[ABP18] S. Arunachalam, J. Briët, and C. Palazuelos. Quantum query algorithms are completely bounded forms. In *9th Innovations in Theoretical Computer Science Conference ITCS, 2018*, volume 94, pages 3:1–3:21, 2018.

[Amb06] Andris Ambainis. Polynomial degree vs. quantum query complexity. *J. Comput. System Sci.*, 72(2):220–238, 2006.

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001.