Quantum computing, norms and polynomials

Escudero Gutiérrez, F.

Part I

Quantum query complexity via polynomials

Chapter 3

The quantum polynomial method is complete

3.1 Introduction

In this chapter, we will review the evolution of the polynomial method in quantum query complexity. Initially, it was proposed by Beals, Buhrman, Cleve, Mosca and de Wolf as a tool to lower bound quantum query complexity [BBC$^+$01], who were inspired by the *classical* polynomial method of Nisan and Szegedy to lower bound the randomized query complexity [NS94]. This technique has been proven useful in many problems, often providing optimal lower bounds (see e.g., [BKT20] and references therein). More than 15 years after its birth, Arunachalam, Briët and Palazuelos refined the method using completely bounded polynomials. This way, it became a tool that potentially allows one to prove upper bounds to quantum query complexity [ABP19]. In this chapter, based on unpublished joint work with Jop Briët, we show how to use completely bounded polynomials to prove several previously known upper bounds to quantum query complexity. In particular, we reprove the upper bounds by Grover, by Deutsch and Jozsa, and by Bernstein and Vazirani [DJ92, BV93, Gro96], and we show that k-fold forrelation can be computed by k quantum queries [AA15, BS21]. Following the result of Arunachalam et al., Gribling and Laurent proposed a hierarchy of semidefinite programs to compute quantum query complexity [GL19]. However, these semidefinite programs do not give any information about how optimal quantum algorithms look like. Finally, we proposed an alternative hierarchy of semidefinite programs, also based on completely bounded polynomials, that not only compute quantum query complexity, but also output the description of optimal quantum query algorithms [Esc25]. Putting everything together, we can say that the polynomial method is *complete*, in the sense that it has all the capabilities desirable from a method to understand quantum query complexity; it can be used to show lower bounds and upper bounds, to compute quantum query complexity, and to extract optimal quantum algorithms.

A novelty of this chapter is that the exposition of all the results is elementary and almost self-contained. In particular, we follow [Esc25] and reprove the Christensen and Sinclair factorization theorem of operator spaces via semidifenite programming [CS87]. This result is the key in the refinement of the polynomial method by Arunachalam et al., but it does not belong to the usual toolbox of the theoretical computer scientist [ABP19]. Thus, this chapter offers to the computer scientist a way to fully understand the method of Arunachalam et al. without requiring a background in operator spaces.

3.2 Quantum lower bounds by polynomials

The key observation by Beals et al. that linked quantum query algorithms to polynomials is that the bias of a quantum algorithm that makes t queries is a multilinear polynomial of degree at most $2t$ [BBC$^+$01].

Theorem 3.1. *Let $\mathcal{A} : \{-1, 1\}^n \to [-1, 1]$ be the bias of t-query quantum algorithm. Then, $\mathcal{A}$ is a polynomial of degree at most $2t$.*

Proof. Before the measurement, on input x, the algorithm prepares a pure quantum state that can be written as

$$|\psi_t(x)\rangle = U_t(O_x \otimes \mathrm{Id}_d)U_{t-1} \dots U_1(O_x \otimes \mathrm{Id}_d)U_0|\psi_0\rangle$$

for some fixed unitary matrices $U_0, \dots U_t$ and some fixed pure state $|\psi_0\rangle$. Note that by definition of matrix multiplication, the coefficients of $|\psi_t(x)\rangle$ in the computational basis are multilinear polynomials of degree at most t. Hence, if $\{M_{-1}, M_1\}$ is the binary measurement performed by the algorithm, then the bias, $\langle\psi_t(x)|M_1|\psi_t(x)\rangle - \langle\psi_t(x)|M_{-1}|\psi_t(x)\rangle$, is a polynomial of degree at most $2t$. □

A direct consequence of Theorem 3.1 is that to lower bound the quantum query complexity of a Boolean function f, it suffices to show that it cannot be approximated by polynomials of low degree. More formally, we have the following.

Definition 3.2. Let $f : D \subseteq \{-1,1\}^n \to \{-1,1\}$ and $\varepsilon \geq 0$. The *ε-approximate degree of f* is the minimum degree of a *bounded* polynomial $p : \{-1,1\}^n \to [-1,1]$ such that $|p(x) - f(x)| \leq \varepsilon$ for every $x \in D$. We use $\widetilde{\deg}_\varepsilon(f)$ to refer to this quantity. We also use $\widetilde{\deg}(f)$ to refer to $\widetilde{\deg}_{2/3}(f)$ and $\deg(f)$ to refer to $\widetilde{\deg}_0(f)$.

Corollary 3.3. *Let $f : D \subseteq \{-1,1\}^n \to \{-1,1\}$ be a Boolean function and let $\varepsilon \geq 0$. Then, $\widetilde{\deg}_\varepsilon(f)/2 \leq Q_\varepsilon(f)$.*

As an example of an application of Corollary 3.3, we will show that the quantum query complexity of the OR_n function is $\Omega(\sqrt{n})$, which implies that Grover's algorithm is optimal [Gro96]. To do that, we prove that $\widehat{\deg}(\mathrm{OR}_n) = \Omega(\sqrt{n})$, originally shown in [NS94], and then apply Corollary 3.7. We define the OR_n function as $\mathrm{OR}_n(x) = 1$ if $x = 1^n$ and $\mathrm{OR}_n(x) = -1$ otherwise.

Proposition 3.4. $Q(\mathrm{OR}_n) = \Omega(\sqrt{n})$.

Proof. By Corollary 3.3 it suffices to show that $\widetilde{\deg}(\mathrm{OR}_n) = \Omega(\sqrt{n})$. Let $p : \{-1,1\}^n \to [-1,1]$ be a degree-t polynomial that satisfies

$$|p(x) - \mathrm{OR}_n(x)| \leq 2/3$$

for every $x \in \{-1,1\}^n$. Consider the symmetrization p' of p, given by $p'(x) := \sum_{\pi \in \mathcal{S}_n} p(\pi \circ x)/n!$. The symmetric polynomial $p' : \{-1,1\}^n \to \mathbb{R}$ also has degree t, takes values between -1 and 1 and satisfies that

$$|p'(x) - \mathrm{OR}_n(x)| \leq 2/3.$$

By the Minsky-Papert symmetrization technique, Proposition 2.13, there is a univariate polynomial q of degree t such that $q(x) = p'(\sum_i x_i/n)$ for every $x \in \{-1,1\}^n$ and $q([-1,1]) \subseteq [-1,1]$. In particular, $|q((n-2)/n) - (-1)| \leq 2/3$ and $|q(1) - 1| \leq 2/3$. Hence, $|q((n-2)/n) - q(1)| \geq 2/3$. By Markov brothers' inequality, Proposition 2.11, this implies that

$$t = \sqrt{\frac{2/3}{1 - (n-2)/n}} = \Omega(\sqrt{n}),$$

as desired. □

3.2.1 Quantum upper bounds by polynomials

The two main techniques to prove lower bounds for quantum query complexity are the polynomial and the adversary method. The latter was proposed in 2000 by Ambainis [Amb00], and it was quickly refined to also serve as a tool to prove quantum query upper bounds [HLv07]. However, since 2003 it is known that there are functions f such that $Q(f) > (\widetilde{\deg}(f))^c$ for some constant $c > 1$ [Amb03], so the polynomial method does not provide upper bounds to quantum query complexity. A natural question was whether a refinement of the polynomial method would allow it to serve as a tool to prove quantum upper bounds. An attempt of this refinement was proposed by Aaronson, Ambainis, Iraids, Kokainis, Smotrovs [AAI$^+$16]. They strengthened Theorem 3.1 by noticing that the bias of every quantum t-query algorithm is not only a multilinear polynomial of degree at most $2t$, but also the amplitudes of such algorithms are multilinear forms of degree t. This is true because if one looks at the state prepared by the quantum algorithm after t queries it has the form of

$$U_t(O_x \otimes \mathrm{Id}_d)U_{t-1} \dots U_1(O_x \otimes \mathrm{Id}_d)U_0|\psi_0\rangle.$$

In particular, if one queried different inputs $x_1, \dots, x_t$ on every query,

$$U_t(O_{x_t} \otimes \mathrm{Id}_d)U_{t-1} \dots U_1(O_{x_1} \otimes \mathrm{Id}_d)U_0|\psi_0\rangle,$$

then the amplitudes of the resulting state would be linear in every input. Hence, the polynomials representing the bias of quantum query algorithms are more structured than initially noted by Beals et al. [BBC$^+$01]. Unfortunately, as shown in the work by Aaronson et al., the corresponding notion of polynomial degree also fails to provide upper bounds to quantum query complexity. However, the idea of Aaronson et al. was in the correct direction. Shortly after, Arunachalam, Briët and Palazuelos realized that if instead of querying binary strings the algorithms queried any contractions (matrices with operator norm at most 1) $X_1 \dots, X_t$ the amplitudes of the resulting vector,

$$U_t X_t U_{t-1} \dots U_1 X_1 U_0 |\psi_0\rangle,$$

would still be linear in $X_1, \dots, X_t$ and bounded by 1 in absolute value [ABP19]. Furthermore, the same is true if one takes *tensor products with identity,* meaning that for every $m \in \mathbb{N}$, every m-dimensional vector $|\phi\rangle$ and contractions $X_1, \dots, X_t$ we have that the amplitudes of

$$(U_t \otimes \mathrm{Id}_m) X_t (U_{t-1} \otimes \mathrm{Id}_m) \dots (U_1 \otimes \mathrm{Id}_m) X_1 (U_0 \otimes \mathrm{Id}_m)(|\psi_0\rangle \otimes |\phi\rangle)$$

are linear in every $X_1, \dots, X_t$ and bounded by 1. As this is true for every $m \in \mathbb{N}$, the bias of quantum query algorithms are, in some sense that we specify below, *completely bounded* polynomials. Surprisingly, Arunachalam et al. showed that the corresponding notion of degree fully characterizes quantum query complexity, enabling the polynomial method to be a potential tool to prove quantum upper bounds. In the rest of the section, we will make this idea rigorous, and give examples of quantum upper bounds by polynomials.

3.3 The completely bounded polynomial method

We start by defining a notion of completely bounded degree, and we will later prove that it characterizes quantum query complexity.

Definition 3.5. Let $f : D \subseteq \{-1,1\}^n \to \{-1,1\}$ and $\varepsilon \geq 0$. The *ε-approximate completely bounded degree of f* is the minimum $t \in \mathbb{N}$ such that there exists a t-linear form $T : \mathbb{R}^{2n} \times \dots \times \mathbb{R}^{2n} \to \mathbb{R}$ such that

- $\|T\|_{\mathrm{cb}} \leq 1$,
- and $|T((x, 1^n), \dots, (x, 1^n)) - p(x)| \leq \varepsilon \quad \forall\, x \in D$.

We use $\widetilde{\mathrm{cbdeg}}_\varepsilon(f)$ to refer to this quantity and $\widetilde{\mathrm{cbdeg}}(f)$ to refer to $\widetilde{\mathrm{cbdeg}}_{2/3}(f)$.

As we argued at the beginning of this section, every t-query quantum algorithm determines a completely bounded form T, so we have that $Q_\varepsilon(f) \geq \widetilde{\mathrm{cbdeg}}_\varepsilon(f)/2$. This strengthens the original polynomial method, because $\|T\|_\infty \leq \|T\|_{\mathrm{cb}}$. Given that there exist separations between the infinity and the completely bounded norms, see for instance [BP19], it is expected that this refinement of the polynomial method allows one to prove stronger quantum lower bounds. Additionally, Arunachalam et al. showed that $Q_\varepsilon(f) = \widetilde{\mathrm{cbdeg}}_\varepsilon(f)/2$, turning the polynomial method into a tool to prove quantum upper bounds.

Theorem 3.6 (Quantum query algorithms are completely bounded forms [ABP19])**.** *Let $p : \{-1,1\}^n \to \mathbb{R}$. Then, the following are equivalent;*

(a) p is the bias of a t-query quantum algorithm.

(b) There exists a $2t$-linear form $T : \mathbb{R}^{2n} \times \cdots \times \mathbb{R}^{2n} \to \mathbb{R}$ such that

$$\|T\|_{\mathrm{cb}} \leq 1 \quad \text{and} \quad T((x, 1^n), \ldots, (x, 1^n)) = p(x) \quad \forall\, x \in \{-1,1\}^n.$$

Corollary 3.7 (The completely bounded polynomial method)**.** *Let $f : D \subseteq \{-1,1\}^n \to \{-1,1\}$ and $\varepsilon \geq 0$. Then, $Q_\varepsilon(f) = \widetilde{\mathrm{cbdeg}}_\varepsilon(f)72$.*

In order to prove Theorem 3.6, Arunachalam et al. established a relation between operator spaces, where the completely bounded norm has been widely studied [Pau03], and quantum algorithms. In particular, they realized that a seminal result by Christensen and Sinclair, which asserts that multilinear forms are completely bounded if and only if they factor in a way resembling the structure of quantum algorithms, allows one to determine which polynomials can be produced by quantum query algorithms.

Theorem 3.8 (Christensen and Sinclair factorization [CS87])**.** *Let $T : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \to \mathbb{R}$ be a t-linear form. Then, $\|T\|_{\mathrm{cb}} \leq 1$ if and only if there exist $d \in \mathbb{N}$, $(n+d)$-dimensional contractions $A_0, \ldots, A_t$, an $(n+d)$-dimensional unit vector v such that*

$$T(x_1, \ldots, x_t) = \langle v, A_t(\mathrm{Diag}(x_t) \otimes \mathrm{Id}_d)A_{t-1} \ldots A_1(\mathrm{Diag}(x_1) \otimes \mathrm{Id}_d)A_0 v\rangle,$$

for every $x_1, \ldots, x_t \in \mathbb{R}^n$.

The original statement of Theorem 3.8 works for any operator space, and the one we use corresponds to the particular case of the natural operator space defined by ℓ_∞. Also, the usual formulation of Theorem 3.8 is for complex operator spaces, which was the one applied by Arunachalam et al. [ABP19]. However, Theorem 3.1 is sufficient to prove Theorem 3.6, provided that we assume, without loss of generality, that we use real numbers for quantum query algorithms (see Remark 2.5). In Section 3.4 we will give a new proof of Theorem 3.8, based on [Esc25], via semidefinite programming. Now, we are ready to prove Theorem 3.6.

Proof of Theorem 3.6. We first prove that $a) \implies b)$. By Remark 2.5, we have that the bias of a t-query quantum algorithm can be written as

$$\begin{aligned}\mathcal{A}(x) =& \langle v, A_0^\mathsf{T}(\mathrm{Diag}(1^n, x) \otimes \mathrm{Id}_d)A_1^\mathsf{T} \ldots A_{t-1}^\mathsf{T}(\mathrm{Diag}(1^n, x) \otimes \mathrm{Id}_d)A_t^\mathsf{T} \\ &\cdot (M_1 - M_{-1})A_t(\mathrm{Diag}(1^n, x) \otimes \mathrm{Id}_d)A_{t-1} \ldots A_1(\mathrm{Diag}(1^n, x) \otimes \mathrm{Id}_d)A_0 v\rangle,\end{aligned}$$

where $A_0, \ldots, A_T$ are $(n+d)$-dimensional contractions, v is an $(n+d)$-dimensional unit vector and $\{M_{-1}, M_1\}$ is a $(n+d)$ POVM. If we define the $(2t)$-linear form

$T : \mathbb{R}^{2n} \times \cdots \times \mathbb{R}^{2n} \to \mathbb{R}$ given by

$$\begin{aligned} T(y_1, \ldots, y_{2t}) =& \langle v, A_0^\intercal (\mathrm{Diag}(y_{2t}) \otimes \mathrm{Id}_d) A_1^\intercal \ldots A_{t-1}^\intercal (\mathrm{Diag}(y_{t+1}) \otimes \mathrm{Id}_d) A_t^\intercal \\ & \cdot (M_1 - M_{-1}) A_t (\mathrm{Diag}(y_t) \otimes \mathrm{Id}_d) A_{t-1} \ldots A_1 (\mathrm{Diag}(y_1) \otimes \mathrm{Id}_d) A_0 v \rangle, \end{aligned}$$

we have that $T((1^n, x), \ldots, (1^n, x)) = \mathcal{A}(x)$. Furthermore, as $\|M_1 - M_{-1}\|_{\mathrm{op}} \leq 1$, by Theorem 3.8 it follows that $\|T\|_{\mathrm{cb}} \leq 1$. Hence, we have showed that $a) \implies b)$.

We now prove that $b) \implies a)$. Let $p : \{-1,1\}^n \to \mathbb{R}$ be such that there exists a $2t$-linear form $T : \mathbb{R}^{2n} \times \cdots \times \mathbb{R}^{2n} \to \mathbb{R}$ satisfying that

$$\|T\|_{\mathrm{cb}} \leq 1 \quad \text{and} \quad T((x, 1^n), \ldots, (x, 1^n)) = p(x) \quad \forall\, x \in \{-1,1\}^n.$$

By Theorem 3.8, there exist $d \in \mathbb{N}$, $(n+d)$-dimensional contractions $A_0, \ldots, A_{2t}$ and $(n+d)$-dimensional unit vectors u, v such that

$$T(y_1, \ldots, y_{2t}) = \langle v, A_{2t} (\mathrm{Diag}(y_{2t}) \otimes \mathrm{Id}_d) A_{2t-1} \ldots A_1 (\mathrm{Diag}(y_1) \otimes \mathrm{Id}_d) A_0 v \rangle,$$

for every $y_1, \ldots, y_{2t} \in \mathbb{R}^{2n}$. For every $x \in \{-1,1\}^n$ we define

$$\begin{aligned} v_1(x) &= A_t (\mathrm{Diag}(x, 1^n) \otimes \mathrm{Id}_d) A_{t-1} \ldots A_1 (\mathrm{Diag}(x, 1^n) \otimes \mathrm{Id}_d) A_0 v, \\ v_2(x) &= (\mathrm{Diag}(x, 1^n) \otimes \mathrm{Id}_d) A_{t+1}^\intercal \ldots A_{2t-1}^\intercal (\mathrm{Diag}(x, 1^n) \otimes \mathrm{Id}_d) A_{2t}^\intercal v. \end{aligned}$$

Note that $\langle v_2(x), v_1(x) \rangle = T((x, 1^n), \ldots, (x, 1^n))$. Hence, it just remains to define a t-query quantum algorithm whose bias is $\langle v_2(x), v_1(x) \rangle$. To do that, we define $2(n+d)$-dimensional contractions

$$\begin{aligned} \widetilde{A}_0 &= (X \otimes \mathrm{Id}_{n+d}) c\text{-}A_0 (X \otimes \mathrm{Id}_{n+d}) c\text{-}A_{2t}^\intercal (H \otimes \mathrm{Id}_{n+d}), \\ \widetilde{A}_i &= (X \otimes \mathrm{Id}_{n+d}) c\text{-}A_i (X \otimes \mathrm{Id}_{n+d}) c\text{-}A_{2t-i}^\intercal, \quad \text{for } i \in [t-1], \\ \widetilde{A}_t &= (H \otimes \mathrm{Id}_{n+d}) c\text{-}A_t (X \otimes \mathrm{Id}_{n+d}), \end{aligned}$$

where c-A is the controlled version of A. Then, we have that the vector prepared by the corresponding quantum query algorithm is

$$\begin{aligned} |\psi(x)\rangle &= \widetilde{A}_t (\mathrm{Id}_2 \otimes \mathrm{Diag}(x, 1^n) \otimes \mathrm{Id}_d) \widetilde{A}_{t-1} \ldots \widetilde{A}_1 (\mathrm{Id}_2 \otimes \mathrm{Diag}(x, 1^n) \otimes \mathrm{Id}_d) \widetilde{A}_0 (|0\rangle \otimes |v\rangle) \\ &= \frac{1}{2} (|0\rangle \otimes (|v_1(x)\rangle + |v_2(x)\rangle) + (|1\rangle \otimes (|v_1(x)\rangle - |v_2(x)\rangle)). \end{aligned}$$

Finally, if we choose the measurement $\{M_{-1}, M_1\}$ to be $M_1 = |0\rangle\langle 0| \otimes \mathrm{Id}_{n+d}$ and

$M_{-1} = |1\rangle\langle 1| \otimes \mathrm{Id}_{n+d}$, then we have that the bias of the quantum algorithm is

$$\mathcal{A}(x) = \langle \psi(x)|(M_1 - M_{-1})|\psi(x)\rangle = \langle v_1(x), v_2(x)\rangle,$$

as desired. □

3.3.1 Examples of quantum upper bounds by polynomials

In this section, we will reprove several quantum upper bounds via the polynomial method. We will show that certain functions are completely bounded polynomials of degree $2t$, and we will invoke Theorem 3.6, which ensures that they are the bias of a t-query quantum algorithm.

Interestingly, for all of the examples of this section, the following non-commutative version of the Cauchy-Schwarz inequality will play a key role.

Lemma 3.9. *Let $X_1, \dots, X_n \in M_m$ and let $Y_1, \dots, Y_n \in M_m$. Then,*

$$\Big\| \sum_{i=1}^n X_i Y_i \Big\|_{\mathrm{op}}^2 \leq \Big\| \sum_{i=1}^n X_i X_i^{\mathsf{T}} \Big\|_{\mathrm{op}} \Big\| \sum_{i=1}^n Y_i^{\mathsf{T}} Y_i \Big\|_{\mathrm{op}}.$$

Proof. Consider the following matrices

$$X = \begin{pmatrix} X_1 & \dots & X_n \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{pmatrix} \quad \text{and} \quad Y = \begin{pmatrix} Y_1 & \dots & 0 \\ Y_2 & \dots & 0 \\ \vdots & & \vdots \\ Y_n & \dots & 0 \end{pmatrix}.$$

First, we have that $\|XY\|_{\mathrm{op}}^2 \leq \|XX^{\mathsf{T}}\|_{\mathrm{op}} \|Y^{\mathsf{T}}Y\|_{\mathrm{op}}$. Finally, we have that $\|XY\|_{\mathrm{op}}^2 = \|\sum_{i=1}^n X_i Y_i\|_{\mathrm{op}}^2$, $\|XX^{\mathsf{T}}\|_{\mathrm{op}} = \|\sum_{i=1}^n X_i X_i^{\mathsf{T}}\|_{\mathrm{op}}$ and $\|Y^{\mathsf{T}}Y\|_{\mathrm{op}} = \|\sum_{i=1}^n Y_i^{\mathsf{T}} Y_i\|_{\mathrm{op}}$. □

Reproving Deutsch-Jozsa

Deutsch and Jozsa gave a 1-query quantum algorithm whose bias is a Boolean function $f : D \subseteq \{-1,1\}^n \to \{-1,1\}$ whose classical query complexity is $\Omega(n)$ [DJ92]. Here,

$$D = \{x \in \{-1^n, 1^n\} : x \text{ is balanced}\} \cup \{-1^n, 1^n\},$$

where x is balanced if it has the same number of -1's and 1's, and f is given by

$$f(x) = \begin{cases} 1 & \text{if } x \in \{-1^n, 1^n\}, \\ -1 & \text{if } x \text{ is balanced,} \end{cases}$$

Here, we reprove the result by Deutsch and Jozsa showing that there exists a bilinear form $T : \mathbb{R}^{2n} \times \mathbb{R}^{2n} \to \mathbb{R}$ such that

$$\|T\|_{\mathrm{cb}} \leq 1 \quad \text{and} \quad T((x, 1^n), (x, 1^n)) = f(x) \quad \forall\, x \in D.$$

This bilinear form is given by

$$T((x, x'), (y, y')) = 2\mathbb{E}_{i\in[n]} x_i \mathbb{E}_{j\in[n]} y_j - \mathbb{E}_{i\in[n]} x_i y_i,$$

where $x, x', y, y' \in \{-1, 1\}^n$ and the expectation is taken with respect to the uniform distribution on $[n]$. (The form T does not depend on the variables x' and y', but we write it like that for consistency with Theorem 3.6). It is routine to check that $T((x, 1^n), (x, 1^n)) = f(x)$ if $x \in \{-1^n, 1^n\}$ or x is balanced. To show that $\|T\|_{\mathrm{cb}} \leq 1$, note that for any contractions $X_1, \dots, X_n, Y_1, \dots, Y_n$ it follows from Lemma 3.9 that

$$\begin{aligned} \|\mathbb{E}_i X_i (2\mathbb{E}_j Y_j - Y_i)\|_{\mathrm{op}}^2 &\leq \|\mathbb{E}_i X_i X_i^{\mathsf{T}}\|_{\mathrm{op}} \|\mathbb{E}_i (2\mathbb{E}_j Y_j - Y_i)^{\mathsf{T}} (2\mathbb{E}_k Y_k - Y_i)\|_{\mathrm{op}} \\ &\leq \|4\mathbb{E}_{j,k} Y_j^{\mathsf{T}} Y_k - 2\mathbb{E}_{i,j} Y_j^{\mathsf{T}} Y_i - 2\mathbb{E}_{i,k} Y_i^{\mathsf{T}} Y_k + \mathbb{E}_i Y_i^{\mathsf{T}} Y_i\|_{\mathrm{op}} \\ &= \|\mathbb{E}_i Y_i^{\mathsf{T}} Y_i\|_{\mathrm{op}} \\ &\leq 1. \end{aligned}$$

Reproving k-fold forrelation

We now consider the problem where, given k-Boolean functions $f_1, \dots, f_k : \{0, 1\}^n \to \{-1, 1\}$, the goal is to compute its k-fold forrelation (standing for *Fourier correlation*)$\mathrm{forr_k} : \{-1, 1\}^{2^n} \times \cdots \times \{-1, 1\}^{2^n} \to \mathbb{R}$, which is given by

$$\mathrm{forr_k}(f_1, \dots, f_k) = \frac{1}{2^{\frac{n(k-1)}{2}}} \sum_{x_1, \dots, x_{k-1} \in \{0,1\}^n} f_1(x_1)(-1)^{\langle x_1, x_2\rangle} f_2(x_2) \dots \cdot (-1)^{\langle x_{k-2}, x_{k-1}\rangle} f_{k-1}(x_{k-1}) \widehat{f_k}(x_{k-1}),$$

where $\langle x, y\rangle = \sum_i x_i y_i$. Here, the queries are made to the truth tables of $f_1, \dots, f_k$. Aaronson and Ambainis introduced this problem as a candidate to witness the largest possible separation between quantum and query complexities [AA15], which was later confirmed by Bansal and Sinha [BS21]. Here, we reprove that f can be computed as the bias of a quantum algorithm that makes k queries, one to each $f_1, \ldots, f_k$. Note that this is not the model that we have considered so far, where all the queries where made to the same input. However, a simple modification of Theorem 3.6 ensures that such an algorithm exists if $\mathrm{forr_k}$, which is a k-linear form, satisfies $\|\mathrm{forr_k}\|_{\mathrm{cb}} \leq 1$. Thus, it suffices to check the latter. Indeed, for m-dimensional orthogonal matrices $F_1(x_1), \ldots, F_k(x_k)$ we have that

$$\|(\mathrm{forr_k})_m(F_1, \ldots, F_k)\|_{\mathrm{op}}^2 = \frac{1}{2^{n(k-1)}} \| \sum_{x_1} F_1(x_1) \sum_{x_2 \ldots x_n} (-1)^{\langle x_1, x_2 \rangle} F_2(x_2) \ldots \widehat{F_k}(x_{k-1}) \|_{\mathrm{op}}^2,$$

where $\widehat{F}_k(x_{k-1}) = \mathbb{E}_{x_k} (-1)^{\langle x_{k-1}, x_k \rangle} F_k(x_k)$ is the matrix-valued Fourier coefficient. Next,

$$\begin{aligned}
\|(\mathrm{forr_k})_m(F_1, \ldots, F_k)\|_{\mathrm{op}}^2 &\leq \frac{1}{2^n} \| \sum_{x_1} F_1(x_1) F_1^{\mathsf{T}}(x_1) \|_{\mathrm{op}} \\
&\quad \cdot \underbrace{\frac{1}{2^{n(k-2)}} \| \sum_{x_2, \ldots, x_n'} \ldots F_2^{\mathsf{T}}(x_2) \underbrace{\left(\sum_{x_1} (-1)^{\langle x_1, x_2 \rangle} (-1)^{\langle x_1, x_2' \rangle} \right)}_{2^n \delta_{x_2, x_2'}} F_2(x_2') \ldots \|_{\mathrm{op}}}_{(*)} \\
&\leq \frac{1}{2^{n(k-3)}} \| \sum_{x_2} \left(\sum_{x_3, \ldots, x_n} (-1)^{\langle x_2, x_3 \rangle} F_3(x_3) \ldots \right)^{\mathsf{T}} F_2^{\mathsf{T}}(x_2) F_2(x_2) \\
&\quad \cdot \left(\sum_{x_3, \ldots, x_n} (-1)^{\langle x_2, x_3 \rangle} F_3(x_3) \ldots \right) \|_{\mathrm{op}},
\end{aligned}$$

where in the first line we have applied Lemma 3.9, and in the third line that $F_1(x_1)$ are orthogonal matrices. Now, as $F_2^{\mathsf{T}}(x_2) F_2(x_2) = \mathrm{Id}_m$, we have that

$$\begin{aligned}
&\|(\mathrm{forr_k})_m(F_1, \ldots, F_k)\|_{\mathrm{op}}^2 \\
&\leq \frac{1}{2^{n(k-3)}} \underbrace{\| \sum_{x_3, x_3' \ldots, x_n', x_n'} \ldots F_3^{\mathsf{T}}(x_3) \left(\sum_{x_2} (-1)^{\langle x_2, x_3 \rangle} (-1)^{\langle x_2', x_3' \rangle} \right) F_3(x_3') \ldots \|_{\mathrm{op}}}_{(**)}.
\end{aligned}$$

Now, $(**)$ is essentially the same as $(*)$, so iterating the argument that led us from $(*)$ to $(**)$ we arrive at

$$\|(\mathrm{forr_k})_m(F_1,\dots,F_k)\|_{\mathrm{op}}^2 \leq \|\sum_{x_{k-1}} \widehat{F}_k^{\mathsf{T}}(x_{k-1})\widehat{F}_k(x_{k-1})\|_{\mathrm{op}} = \|\mathbb{E}_x F_k^{\mathsf{T}}(x)F_k(x)\|_{\mathrm{op}} = 1,$$

where in the first equality we have used Parseval identity and in the second that $F_k(x)$ are orthogonal. Thus, $\mathrm{forr_k}$ is completely bounded, as desired.

Other examples

One can also reprove other well-known quantum upper bounds using polynomials. Briët reproved Grover's upper bound of $O(\sqrt{n})$ quantum queries to compute the OR_n function by showing that the polynomials constructed by Nisan and Szegedy to approximate OR_n are completely bounded [Bri19, NS94]. Also, using a modification of Theorem 3.6, we could show that there exists an algorithm that with one quantum query to the truth table of a Boolean function can sample from its Fourier distribution, reproving Bernstein-Vazirani's celebrated result [BV93]. We will not prove the latter claim because it would require introducing more notation and would not add conceptual value, as we have already accomplished the purpose of this section: demonstrating that quantum upper bounds can follow from the polynomial method.

3.4 From polynomials to quantum algorithms

In this section, we will start by giving an alternative proof of the Christensen-Sinclair factorization theorem, Theorem 3.8, via semidefinite programming. Contrary to the original proof, ours is elementary, constructive and does not need to use the Hahn-Banach theorem (just a finite-dimensional separation result). We will follow [Esc25], where a more general version of Christensen and Sinclair's result is proven. After, we will use the fact that this proof is based on semidefinite programming and is constructive to give a hierarchy of semidefinite programs that computes quantum query complexity and outputs optimal quantum query algorithms.

3.4.1 Christensen-Sinclair factorization via SDPs

We will prove an equivalent version of Theorem 3.8. To state it, we should introduce the representation norm of a t-linear form $T : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \to \mathbb{R}$, which is given by

$$\begin{aligned} \|T\|_{\mathrm{rep}} = \inf \quad & w \\ \text{s.t.} \quad & T(x_1, \ldots, x_t) = \langle u, A_0(\mathrm{Diag}(x_1) \otimes \mathrm{Id}_d) A_1 \ldots A_{t-1}(\mathrm{Diag}(x_t) \otimes \mathrm{Id}_d) A_t v \rangle \,, \\ & \forall\, x_1, \ldots, x_t \in \mathbb{R}^n, \\ & d \in \mathbb{N},\ u, v \in \mathbb{R}^d,\ \|u\|_2^2 = \|v\|_2^2 = w, \\ & A_0 \in M_{d,nd},\ A_1, \ldots, A_{t-1} \in M_{nd,nd},\ A_t \in M_{nd,d} \text{ contractions.} \end{aligned} \tag{3.1}$$

Now, we can rewrite Theorem 3.10 in the following way.

Theorem 3.10 (Christensen and Sinclair factorization [CS87]). *Let $T : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \to \mathbb{R}$ be a t-linear form. Then, $\|T\|_{\mathrm{cb}} = \|T\|_{\mathrm{rep}}$.*

We will prove the following result, which is stronger than Theorem 3.10.

Theorem 3.11. *Given a t-linear form $T : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \to \mathbb{R}$, there is a pair of semidefinite programs (P_{CS}) and (D_{CS}) such that*

(i) (P_{CS}) optimal value equals $\|T\|_{\mathrm{rep}}$,

(ii) (D_{CS}) optimal value equals $\|T\|_{\mathrm{cb}}$,

(iii) (D_{CS}) is the dual of (P_{CS}) and their optimal values are equal.

Theorem 3.11 has three important consequences. The first one is already clear from the statement, and the other two will become clear later (see Remark 3.12). These consequences are:

(a) Theorem 3.11 implies Theorem 3.10;

(b) (P_{CS}) and (D_{CS}) have $O(\mathrm{poly}(n)^t)$ variables, so the known algorithms to approximate semidefinite programs can be used to efficiently compute the completely bounded norm. This will imply that there is a hierarchy of SDPs to compute quantum query complexity.

(c) From the solution returned by these algorithms one can extract a description of the vectors and matrices appearing in a factorization as in Eq. (3.1). This will imply that optimal quantum query algorithms can be extracted from the hierarchy of SDPs mentioned in Item (b).

We divide the proof of Theorem 3.11 in 3 parts. In the first, we introduce (P_{CS}) and prove Theorem 3.11 (*i*), in the second we introduce (D_{CS}) and prove Theorem 3.11 (*ii*), and in the third we show that (P_{CS}) and (D_{CS}) are semidefinite programs and prove Theorem 3.11 (*iii*).

The primal semidefinite program

In this section, we introduce (P_{CS}) and prove Theorem 3.11 Item (i). Before doing that, we give some intuition for why $\|T\|_{\mathrm{rep}}$ can be formulated as a semidefinite program. Assume that T factors as in Eq. (3.1). Then, we consider the following block structure for the contractions A_s:

$$A_0 = \begin{pmatrix} A_0(1) & \dots & A_0(n) \end{pmatrix}, \; A_s = \begin{pmatrix} A_s(1,1) & \dots & A_s(1,n) \\ \vdots & \ddots & \vdots \\ A_s(n,1) & \dots & A_s(n,n) \end{pmatrix}, \; A_t = \begin{pmatrix} A_t(1) \\ \dots \\ A_t(n) \end{pmatrix}, \tag{3.2}$$

for $s \in [t-1]$. We also define the following vectors,

$$v_i = A_t(i)v, \text{ for } i \in [n], \tag{3.3}$$

$$v_{\mathbf{i}} = A_{t-s}((i_1, i_2)) \dots A_{t-1}((i_s, i_{s+1}))A_t(i_{s+1})v, \text{ for } \mathbf{i} \in [n]^{s+1}, \; s \in [t-1], \tag{3.4}$$

$$v'_{\mathbf{i}} = A_0(i_1)A_1((i_1, i_2)) \dots A_t(i_t)v, \text{ for } \mathbf{i} \in [n]^t. \tag{3.5}$$

We note that $T_{\mathbf{i}} = \langle u, v'_{\mathbf{i}} \rangle$. Hence, $T_{\mathbf{i}}$ is encoded in the entries of Y =Gram$\{u, v_{\mathbf{i}}, v'_{\mathbf{i}}\}$ (which corresponds to (3.7) below). In addition, the fact that the A_i are contractions can be encoded in the entries of this Gram matrix (which gives rise to Eqs. (3.9) to (3.11) below). With these intuitions, we are ready to state (P_{CS}):

$$
\begin{aligned}
\inf \quad & w && (P_{\mathrm{CS}})\\
\text{s.t.} \quad & w \geq 0,\ Y, Y' \succeq 0, && (3.6)\\
& Y'_{0,\mathbf{i}} = T_{\mathbf{i}},\ \mathbf{i} \in [n]^t, && (3.7)\\
& Y'_{0,0} = w, && (3.8)\\
& \sum_{i\in[n]} Y_{i,i} \leq w, && (3.9)\\
& \sum_{i\in[n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^s} \preceq \oplus_{i\in[n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^{s-1}},\ s \in [t-1], && (3.10)\\
& (Y'_{\mathbf{j},\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^t} \preceq \oplus_{i\in[n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^{t-1}}, && (3.11)
\end{aligned}
$$

where $Y \in M_{n+\cdots+n^t}$ and $Y' \in M_{1+n^t}$. The rows and columns of Y are labeled by the elements of $[n] \cup \cdots \cup [n]^t$, and for Y' they are labeled by the elements of $\{0\} \cup [n]^t$. [1]

Proof of Theorem 3.11. Assume first that T factors as in Eq. (3.1) for some vectors with $\|u\|^2 = \|v\|^2 = w$. Consider the block structure for the contractions A_s given in Eq. (3.2), and define the vectors $v_{\mathbf{i}}$ and $v'_{\mathbf{i}}$ as in Eqs. (3.3) to (3.5). Then, $T_{\mathbf{i}} = \langle u, v_{\mathbf{i}} \rangle$, for every $\mathbf{i} \in [n]^t$. Consider the positive semidefinite matrices

$$Y' := \mathrm{Gram}\{u, v'_{\mathbf{i}} :\ \mathbf{i} \in [n]^t\} \quad \text{and} \quad Y := \mathrm{Gram}\{v_{\mathbf{i}} :\ \mathbf{i} \in [n] \cup \cdots \cup [n]^t\},$$

and label the rows and columns corresponding to u with 0 and the ones corresponding to $v_{\mathbf{i}}$ and $v'_{\mathbf{i}}$ with $\mathbf{i}$. First, we have that $T_{\mathbf{i}} = Y'_{0,\mathbf{i}}$, so Eq. (3.7) is satisfied. Eq. (3.8) follows from the fact that $\|u\|^2 = w$. From the fact that A_t is a contraction, Eq. (3.9) follows:

$$\sum_{i\in[n]} Y_{i,i} = \sum_{i\in[n]} \langle v_i, v_i \rangle = \left\langle v, \sum_{i\in[n]} A_t(i)^{\mathsf{T}} A_t(i) v \right\rangle = \langle v, A_t^{\mathsf{T}} A_t v \rangle \leq \langle v, v \rangle = w.$$

From the fact that A_s are contractions for $s \in [t-1]$ Eq. (3.10) follows. Indeed, let

[1] Here, given $i \in [n]$ and $\mathbf{j} \in [n]^s$, $i\mathbf{j}$ should be interpreted as the concatenation of i and $\mathbf{j}$, i.e., $i\mathbf{j} = (i, j_1, \ldots, j_s)$.

$\lambda \in \mathbb{R}^{n^s}$. Then,

$$\begin{aligned}
&\left\langle \lambda, \sum_{i\in[n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^s} \lambda \right\rangle \\
&= \sum_{i\in[n],\mathbf{j},\mathbf{j}'\in[n]^s} \lambda_{\mathbf{j}} \langle v_{i\mathbf{j}}, v_{i\mathbf{j}'}\rangle \lambda_{\mathbf{j}'} \\
&= \sum_{i\in[n],\mathbf{j},\mathbf{j}'\in[n]^s} \lambda_{\mathbf{j}} \langle A_{t-s}(i,j_1)v_{\mathbf{j}}, A_{t-s}(i,j_1')v_{\mathbf{j}'}\rangle \lambda_{\mathbf{j}'} \\
&= \sum_{i\in[n],\mathbf{j},\mathbf{j}'\in[n]^s} \lambda_{\mathbf{j}} \left\langle v_{\mathbf{j}}, A_{t-s}^{\mathsf{T}}(j_1,i)A_{t-s}(i,j_1')v_{\mathbf{j}'}\right\rangle \lambda_{\mathbf{j}'} \\
&= \underbrace{\sum_{\mathbf{j},\mathbf{j}'\in[n]^s} \lambda_{\mathbf{j}} \left\langle v_{\mathbf{j}}, (A_{t-s}^{\mathsf{T}}A_{t-s})(j_1,j_1')v_{\mathbf{j}'}\right\rangle \lambda_{\mathbf{j}'},}_{(*)}
\end{aligned}$$

where in the second equality we have used that $v_{i\mathbf{j}} = A(i,j_1)v_{\mathbf{j}}$, and in the third line that $A_{t-s}(i,j)^{\mathsf{T}} = A_{t-s}^{\mathsf{T}}(j,i)$. Now, if we define $w_{\mathbf{j}} = (\lambda_{1\mathbf{j}}v_{1\mathbf{j}},\dots,\lambda_{n\mathbf{j}}v_{n\mathbf{j}})$, it follows that

$$(*) = \sum_{\mathbf{j},\mathbf{j}'\in[n]^{s-1}} \langle w_{\mathbf{j}}, A_{t-s}^{\mathsf{T}}A_{t-s}w_{\mathbf{j}'}\rangle = \left\langle \left(\sum_{\mathbf{j}} w_{\mathbf{j}}\right), A_{t-s}^{\mathsf{T}}A_{t-s}\left(\sum_{\mathbf{j}'} w_{\mathbf{j}'}\right)\right\rangle.$$

Hence, as $A_{t-s}^{\mathsf{T}}A_{t-s} \preceq \mathrm{Id}$, it is satisfied that

$$\begin{aligned}
(*) &\le \left\langle \left(\sum_{\mathbf{j}\in[n]^{s-1}} w_{\mathbf{j}}\right), \left(\sum_{\mathbf{j}'\in[n]^{s-1}} w_{\mathbf{j}'}\right)\right\rangle = \sum_{i\in[n],\mathbf{j},\mathbf{j}'\in[n]^{s-1}} \lambda_{i\mathbf{j}}\langle v_{i\mathbf{j}}, v_{i\mathbf{j}'}\rangle\lambda_{i\mathbf{j}'} \\
&= \langle \lambda, \oplus_{i\in[n]}(Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]\times[n])^{s-1}\times[n]}\lambda\rangle,
\end{aligned}$$

as desired. The fact that A_0 is a contraction implies Eq. (3.11), and this can be shown similarly to how we just showed that Eq. (3.10) holds.

Now, assume that there exist $Y, Y' \succeq 0$, satisfying equations Eqs. (3.7) to (3.11). Consider $d \in \mathbb{N}$ and vectors $\{u, v_{\mathbf{i}}, v_{\mathbf{i}}\} \in \mathbb{R}^d$ such that

$$Y = \mathrm{Gram}\{v_{\mathbf{i}}\} \quad \text{and} \quad Y' = \mathrm{Gram}\{u, v_{\mathbf{i}}'\}.$$

Eq. (3.8) implies that $\|u\|^2 = w$. We define A_t through its blocks. Let $v \in \mathbb{R}^d$ be a vector with $\|v\|^2 = w$. We define $A_t(i) \in M_d$ as the matrix that maps v to v_i and extend by 0 to the orthogonal complement of $\mathrm{span}\{v\}$. This way, A_t is a contraction, because

$$\|A_t\|_{\mathrm{op}}^2 = \frac{\langle A_t v, A_t v\rangle}{\langle v, v\rangle} = \frac{1}{w}\sum_{i\in[n]}\langle A_t(i)v, A_t(i)v\rangle = \frac{1}{w}\sum_{i\in[n]}\langle v_i, v_i\rangle = \frac{1}{w}\sum_{i\in[n]} Y_{i,i} \leq 1,$$

where in the inequality we have used Eq. (3.9). The definition of A_{t-s} for $s \in [t-1]$ is slightly more complicated. Given $(i,j) \in [n]\times[n]$, the block $A_{t-s}(i,j)$ is defined as the linear map on $\operatorname{span}\{v_{j\mathbf{j}} : \mathbf{j} \in [n]^{s-1}\}$ by

$$A_{t-s}(i,j)v_{j\mathbf{j}} = v_{ij\mathbf{j}}$$

and extended by 0 to the orthogonal complement. First, as $\{v_{j\mathbf{j}} : \mathbf{j} \in [n]^{s-1}\}$ may not be linearly independent, we have to check that this a good definition, namely that for every $\lambda \in \mathbb{R}^{n^{s-1}}$

$$\sum_{\mathbf{j}\in[n]^{s-1}} \lambda_{j\mathbf{j}} v_{j\mathbf{j}} = 0 \implies \sum_{\mathbf{j}\in[n]^{s-1}} \lambda_{j\mathbf{j}} v_{ij\mathbf{j}} = 0. \tag{3.12}$$

Indeed, we can prove something stronger. For any $\lambda \in \mathbb{R}^{n^{s-1}}$, we define $\tilde{\lambda} \in \mathbb{R}^{n^s}$ by $\tilde{\lambda}_{j'\mathbf{j}} := \delta_{j,j'}\lambda_{\mathbf{j}}$, where j is the second index in the pair (i,j) that indexes the block $A_{t-s}(i,j)$. Then,

$$\begin{aligned}
&\left\langle \sum_{\mathbf{j}\in[n]^{s-1}} \lambda_{j\mathbf{j}} v_{ij\mathbf{j}}, \sum_{\mathbf{j}'\in[n]^{s-1}} \lambda_{j\mathbf{j}'} v_{ij\mathbf{j}'} \right\rangle \\
&= \left\langle \lambda, (Y_{(ij\mathbf{j},ij\mathbf{j}')})_{\mathbf{j},\mathbf{j}'\in[n]^{s-1}} \lambda \right\rangle \\
&= \left\langle \tilde{\lambda}, (Y_{(i\mathbf{j},i\mathbf{j}')})_{\mathbf{j},\mathbf{j}'\in[n]^{s}} \tilde{\lambda} \right\rangle \\
&\leq \left\langle \tilde{\lambda}, \sum_{k\in[n]} (Y_{k\mathbf{j},k\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^{s}} \tilde{\lambda} \right\rangle \\
&\leq \left\langle \tilde{\lambda}, \oplus_{k\in[n]} (Y_{k\mathbf{j},k\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^{s-1}} \tilde{\lambda} \right\rangle \\
&= \left\langle \lambda, (Y_{j\mathbf{j},j\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^{s-1}} \lambda \right\rangle \\
&= \left\langle \sum_{\mathbf{j}\in[n]^{s-1}} \lambda_{j\mathbf{j}} v_{j\mathbf{j}}, \sum_{\mathbf{j}'\in[n]^{s-1}} \lambda_{j\mathbf{j}'} v_{j\mathbf{j}'} \right\rangle,
\end{aligned}$$

where in the first inequality we have used that $(Y_{k\mathbf{j},k\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^s} \succeq 0$ for every $k \in [n]$, and in the second inequality we have used (3.10). Thus, Eq. (3.12) holds. Now, we have to check that A_{t-s} is a contraction. By the definition of A_{t-s}, we just have to check that for every $\lambda \in \mathbb{R}^{n^s}$,

$$\lambda v := \begin{pmatrix} \sum_{\mathbf{j}\in[n]^{s-1}} \lambda_{1\mathbf{j}} v_{1\mathbf{j}} \\ \vdots \\ \sum_{\mathbf{j}\in[n]^{s-1}} \lambda_{n\mathbf{j}} v_{n\mathbf{j}} \end{pmatrix}$$

is mapped to a vector with smaller or equal norm. Indeed,

$$\begin{aligned} \langle A_{t-s}\lambda v, A_{t-s}\lambda v\rangle &= \sum_{i,\mathbf{j},\mathbf{j}'\in[n]^s} \lambda_{\mathbf{j}} \langle v_{i\mathbf{j}}, v_{i\mathbf{j}'}\rangle \lambda_{\mathbf{j}'} \\ &= \left\langle \lambda, \sum_{i\in[n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^s} \lambda \right\rangle \\ &\leq \left\langle \lambda, \oplus_{i\in[n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^{s-1}} \lambda \right\rangle \\ &= \langle \lambda v, \lambda v\rangle , \end{aligned}$$

where in the inequality we have used Eq. (3.10). Finally, we define A_0 through its blocks. $A_0(i)$ is defined by $A_0(i)v_{i\mathbf{j}} = v'_{i\mathbf{j}}$ for $\mathbf{j} \in [n]^{t-1}$ and extended by 0 to the orthogonal complement of $\text{span}\{v_{i\mathbf{j}} : \mathbf{j} \in [n]^{t-1}\}$. Using Eq. (3.11), we can check that these blocks are well-defined and that A_0 is a contraction using a similar argument to the one that we have just used to verify the same properties of A_{t-s}. It just remains to show that (u, v, A_i) defines a factorization for T as in (3.1). Eq. (3.1) holds if and only it holds for a basis of $\mathbb{R}^n$. We verify it for the canonical basis $\{e_i\}_{i\in[n]}$. On the one hand, by definition, we have that $T(e_{i_1}, \ldots, e_{i_t}) = T_{\mathbf{i}}$. On the other hand, a simple calculation shows that

$$\begin{aligned} Y'_{0,\mathbf{i}} &= \langle u, A_0(i_1)A_1((i_1, i_2)) \ldots A_{t-1}((i_{t-1}, i_t))A_t(i_t)v\rangle \\ &= \langle u, A_0(\text{Diag}(e_{i_1}) \otimes \text{Id}_d)A_1 \ldots A_{t-1}(\text{Diag}(e_{i_t}) \otimes \text{Id}_d)A_t v\rangle. \end{aligned}$$

Hence, by Eq. (3.7) follows that

$$T(e_{i_1}, \ldots, e_{i_t}) = \langle u, A_0(\text{Diag}(e_{i_1}) \otimes \text{Id}_d)A_1 \ldots A_{t-1}(\text{Diag}(e_{i_t}) \otimes \text{Id}_d)A_t v\rangle,$$

as desired. □

Remark 3.12. (P_{CS}) has $\text{poly}(n)^t$ variables, so Item (b) holds. Item (c) can be inferred from the second part of the proof of Theorem 3.11 Item (i), where a recipe to extract a factorization as in Eq. (3.1) for $(Y'_{0,\mathbf{i}})_{\mathbf{i}}$ satisfying Eqs. (3.8) to (3.11) is given.

The dual semidefinite program

In this section, we introduce (D_{CS}) and prove Theorem 3.11 Item (ii). (D_{CS}) is given by:

$$\begin{aligned}
\sup \quad & \sum_{\mathbf{i}\in[n]^t} T_{\mathbf{i}} y_{0,\mathbf{i}} && (D_{\text{CS}})\\
\text{s.t.} \quad & y_0, y_0' \geq 0,\ \left(y_{\mathbf{i},\mathbf{i}'}\right)_{\mathbf{i},\mathbf{i}'\in[n]^s} \succeq 0,\ \text{for } s\in[t], && (3.13)\\
& y_0 + y_0' \leq 1, && (3.14)\\
& y_0 \geq y_{i,i}, \quad \text{for } i\in[n] && (3.15)\\
& (y_{\mathbf{j}}, y_{\mathbf{j}'})_{\mathbf{j},\mathbf{j}\in[n]^s} \geq (y_{i\mathbf{j}}, y_{i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^s} \quad \text{for } i\in[n],\ s\in[t-1], && (3.16)\\
& \left(\begin{array}{c|c} y_0' & \dots \quad (y_{0,\mathbf{i}})_{\mathbf{i}\in[n]^t}/2 \quad \dots \\ \hline \vdots & \\ \frac{(y_{0,\mathbf{i}})_{\mathbf{i}\in[n]^t}}{2} & \left(y_{\mathbf{i},\mathbf{i}'}\right)_{\mathbf{i},\mathbf{i}'\in[n]^t} \\ \vdots & \end{array}\right) \succeq 0, && (3.17)
\end{aligned}$$

Before diving into the proof, we give some intuition of why the optimal value of (D_{CS}) is $\|T\|_{\text{cb}}$. One should note that Eq. (3.17) means that the variables $y_{0,\mathbf{i}}$ can be written as $\langle u, v_{\mathbf{i}}\rangle$ for some vectors u, $v_{\mathbf{i}}$. Then, roughly speaking, Eqs. (3.15) and (3.16) encode that the $v_{\mathbf{i}}$ equal $X_1(i_1)\dots X_t(i_t)v$ for some contractions $X_1(i_1),\dots,X_t(i_t)$ and a vector v, and Eq. (3.14) encodes that u and v are bounded vectors.

Proof of Theorem 3.11 Item (ii). First, we note that Eq. (3.13) means that there exist $d\in\mathbb{N}$ and vectors $\{u, v, v_{\mathbf{i}} : \mathbf{i}\in[n]^s,\ s\in[t]\}\subset\mathbb{R}^m$ such that $y_0' = \langle u, u\rangle$, $y_0 = \langle v, v\rangle$, and $y_{\mathbf{i},\mathbf{i}'} = \langle v_{\mathbf{i}}, v_{\mathbf{i}'}\rangle$ for every $\mathbf{i}\in[n]^s$ and $s\in[t]$. Then, Eq. (3.15) means that $\langle u, u\rangle + \langle v, v\rangle \leq 1$ and Eq. (3.17) means that $y_{0,\mathbf{i}} = 2\langle u, v_{\mathbf{i}}\rangle$ for every $\mathbf{i}\in[n]^t$. Thus, we can rewrite Eq. (D_{CS}) as

$$\begin{aligned}
\sup \quad & 2\sum_{\mathbf{i}\in[n]^t} T_{\mathbf{i}}\langle u, v_{\mathbf{i}}\rangle, && (3.18)\\
\text{s.t.} \quad & m\in\mathbb{N},\ u, v, v_{\mathbf{i}}\in\mathbb{R}^m,\ \mathbf{i}\in[n]^s,\ s\in[t], \\
& \langle u,u\rangle + \langle v,v\rangle \leq 1, \\
& \langle v,v\rangle \geq \langle v_i, v_i\rangle, \quad \text{for } i\in[n] && (3.19)\\
& (\langle v_{\mathbf{j}}, v_{\mathbf{j}'}\rangle)_{\mathbf{j},\mathbf{j}\in[n]^s} \geq (\langle v_{i\mathbf{j}}, v_{i\mathbf{j}'}\rangle)_{\mathbf{j},\mathbf{j}'\in[n]^s} \quad \text{for } i\in[n],\ s\in[t-1] && (3.20)\\
& && (3.21)
\end{aligned}$$

Next, we will show that Eqs. (3.19) and (3.20) are equivalent to the existence of contractions $X_1,\dots,X_t\in M_m$ such that

$$v_{\mathbf{i}} = X_{t-s+1}(i_1)\dots X_t(i_s)v, \tag{3.22}$$

for every $\mathbf{i}\in[n]^s$ and every $s\in[t]$. Indeed, assume that Eqs. (3.19) and (3.20) hold. Then, for every $i\in[n]$ and every $s\in\{0\}\cup[t]$, we define

$$X_{t-s}(i)v_{\mathbf{j}} := v_{i\mathbf{j}}$$

for every $\mathbf{j}\in[n]^s$ and extend it by 0 on the orthogonal complement of $\text{span}\{v_{\mathbf{j}} : \mathbf{j}\in[n]^s\}$. We have to check that the $X_{t-s}(i)$ are well-defined as linear maps. Namely, that for every $\lambda\in\mathbb{R}^{n^s}$ we have

$$\sum_{\mathbf{j}\in[n]^s}\lambda_{\mathbf{j}}v_{\mathbf{j}} = 0 \implies \sum_{\mathbf{j}\in[n]^s}\lambda_{\mathbf{j}}v_{i\mathbf{j}} = 0.$$

In fact, we can prove that the $X_{t-s}(i)$ are well-defined and contractions at the same time. Indeed, for $\lambda\in\mathbb{R}^{n^s}$ we have that

$$\begin{aligned}
\left\langle \sum_{\mathbf{j}\in[n]^s}\lambda_{\mathbf{j}}v_{i\mathbf{j}}, \sum_{\mathbf{j}'\in[n]^s}\lambda_{\mathbf{j}'}v_{i\mathbf{j}'}\right\rangle &= \left\langle \lambda, \Big(\langle v_{i\mathbf{j}}, v_{i\mathbf{j}'}\rangle\Big)_{\mathbf{j},\mathbf{j}'\in[n]^s}\lambda\right\rangle \\
&\leq \left\langle \lambda, \Big(\langle v_{\mathbf{j}}, v_{\mathbf{j}'}\rangle\Big)_{\mathbf{j},\mathbf{j}'\in[n]^s}\lambda\right\rangle \\
&= \left\langle \sum_{\mathbf{j}\in[n]^s}\lambda_{\mathbf{j}}v_{\mathbf{j}}, \sum_{\mathbf{j}'\in[n]^s}\lambda_{\mathbf{j}'}v_{\mathbf{j}'}\right\rangle,
\end{aligned}$$

where we have used Eq. (3.20) (or Eq. (3.19) if $s = 0$).

On the other hand, if Eq. (3.22) holds, it is a routine check showing that Eqs. (3.19) and (3.20) hold. Putting everything together, we can rewrite (3.18) as

$$\begin{aligned} \sup \quad & 2\sum_{\mathbf{i}\in[n]^t} T_{\mathbf{i}} R_{\mathbf{i}}, \\ \text{s.t.} \quad & R \in \mathbb{R}^{n^t},\ m \in \mathbb{N},\ u, v \in \mathbb{R}^m, X_s \in M_m \text{ contractions for } s \in [t], \\ & \langle u, u\rangle + \langle v, v\rangle \leq 1, \\ & R_{\mathbf{i}} = \langle u, X_1(i_1)\dots X_t(i_t) v\rangle\,, \text{ for } \mathbf{i} \in [n]^t. \end{aligned} \tag{3.23}$$

We finally claim that the above optimization problem is equivalent to

$$\begin{aligned} \sup \quad & 2\sum_{\mathbf{i}\in[n]^t} T_{\mathbf{i}} R_{\mathbf{i}}, \\ \text{s.t.} \quad & R \in \mathbb{R}^{n^t},\ m \in \mathbb{N},\ u, v \in \mathbb{R}^m, X_s \in M_m \text{ contractions for } s \in [t], \\ & \langle u, u\rangle\,, \langle v, v\rangle \leq 1/2, \\ & R_{\mathbf{i}} = \langle u, X_1(i_1)\dots X_t(i_t) v\rangle\,, \text{ for } \mathbf{i} \in [n]^t. \end{aligned} \tag{3.24}$$

We first note that the optimum of Eq. (3.23) is greater or equal than the optimum of Eq. (3.24), because the feasible region is larger in the case of Eq. (3.23). On the other hand, if one picks a feasible instance (u, v, X) of Eq. (3.23), one can define the instance $(\tilde{u}, \tilde{v}, X)$ by

$$\tilde{u} = \frac{u\sqrt{\|u\|^2 + \|v\|^2}}{\sqrt{2}\|u\|},\ \tilde{v} = \frac{v\sqrt{\|u\|^2 + \|v\|^2}}{\sqrt{2}\|v\|},$$

which is feasible for Eq. (3.24) and attains a value greater or equal than (u, v, X), because

$$\begin{aligned} \Big|\sum T_{\mathbf{i}} \langle \tilde{u}, X_1(i_1)\dots X_t(i_t)\tilde{v}\rangle \Big| &= \frac{\|u\|^2 + \|v\|^2}{2\|u\|\|v\|}\Big|\sum T_{\mathbf{i}} \langle u, X_1(i_1)\dots X_t(i_t) v\rangle \Big| \\ &\geq \Big|\sum T_{\mathbf{i}} \langle u, X_1(i_1)\dots X_t(i_t) v\rangle \Big|. \end{aligned}$$

Now, the result follows from the fact that the optimal value of Eq. (3.24) is $\|T\|_{\mathrm{cb}}$. □

Strong duality

Finally, we prove Theorem 3.11 Item (iii).

Proof of Theorem 3.11 Item (iii). First, we show that (P_{CS}) can be expressed as in the canonical form of (P) in Eq. (2.15). To do that we introduce the slack matrix variables Z and Z' and write (P_{CS}) as

$$\begin{aligned} \inf \quad & w && (\widetilde{P}_{\mathrm{CS}}) \\ \text{s.t.} \quad & X := \begin{pmatrix} w & 0 & 0 & 0 & 0 \\ 0 & Y & 0 & 0 & 0 \\ 0 & 0 & Y' & 0 & 0 \\ 0 & 0 & 0 & Z & 0 \\ 0 & 0 & 0 & 0 & Z' \end{pmatrix} \succeq 0 \\ & Y_{0,\mathbf{i}} = T_{\mathbf{i}}, \ \mathbf{i} \in [n]^t, && (3.25) \\ & w - Y'_{0,0} = 0, && (3.26) \\ & w - \sum_{i\in[n]} Y_{i,i} = Z_{0,0}, && (3.27) \\ & \oplus_{i\in[n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^{s-1}} - \sum_{i\in[n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^s} = (Z_{\mathbf{j},\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^s}, \ s\in[t-1], && (3.28) \\ & \oplus_{i\in[n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^{t-1}} - (Y'_{\mathbf{j},\mathbf{j}'})_{\mathbf{j},\mathbf{j}'\in[n]^t} = Z', && (3.29) \end{aligned}$$

One can regard X as a positive semidefinite matrix with some entries set to 0, which can be imposed via linear constraints. Additionally, note that the objective function w is a linear function of the entries of X, and so are the restrictions Eqs. (3.25) to (3.29). Hence, (P_{CS}) has the form of (P) in Eq. (2.15).

Second, we show that (D_{CS}) can be expressed as in the canonical form of (D) in Eq. (2.15). We can rewrite (D_{CS}) as

$$\begin{aligned}
\text{sup} \quad & \sum_{\mathbf{i}\in[n]^t} T_{\mathbf{i}} R_{\mathbf{i}} && (\widetilde{D}_{\mathrm{CS}}) \\
\text{s.t.} \quad & y_0, y_0', R_{\mathbf{i}}, y_{\mathbf{i},\mathbf{i}'}, \mathbf{i}, \mathbf{i}' \in [n]^s,\ s\in[t] \\
& y_0 \geq 0,\ y_0' \geq 0,\ \sum_{\mathbf{j},\mathbf{j}'\in[n]^s} y_{\mathbf{j},\mathbf{j}'} E_{\mathbf{j},\mathbf{j}'} \succeq 0,\ \text{for } s\in[t], && (3.30) \\
& y_0 + y_0' \leq 1, && (3.31) \\
& y_0 \geq y_{i,i}, \quad \text{for } i\in[n] && (3.32) \\
& \sum_{\mathbf{j},\mathbf{j}'\in[n]^s} (y_{\mathbf{j},\mathbf{j}'} - y_{i\mathbf{j},i\mathbf{j}'}) E_{\mathbf{j},\mathbf{j}'} \succeq 0, \quad \text{for } i\in[n],\ s\in[t-1] && (3.33) \\
& y_0' E_{0,0} + \sum_{\mathbf{j}\in[n]^t} R_{\mathbf{j}} \frac{E_{0,\mathbf{j}} + E_{\mathbf{j},0}}{2} + \sum_{\mathbf{i},\mathbf{i}'\in[n]^t} y_{\mathbf{i},\mathbf{i}'} E_{\mathbf{i},\mathbf{i}'} \succeq 0. && (3.34)
\end{aligned}$$

Thus, we have written (D_{CS}) as an optimization problem $(\widetilde{D}_{\mathrm{CS}})$ on the variables $y_0, y_0', R_{\mathbf{i}}, y_{\mathbf{i},\mathbf{i}'}$. Moreover, the objective function is a linear combination of these variables. Also, the constraints are positive semidefinite constraints on matrices that are linear combinations of other matrices, where the coefficients of these linear combinations are $y_0, y_0', R_{\mathbf{i}}, y_{\mathbf{i},\mathbf{i}'}$. Putting everything together, it follows that (D_{CS}) is of the form of (D) in Eq. (2.15).

Third, we show that (D_{CS}) is the dual of (P_{CS}). Equivalently, we prove that $(\widetilde{D}_{\mathrm{CS}})$ is the dual of $(\widetilde{P}_{\mathrm{CS}})$. To take the dual of a primal semidefinite program such as $(\widetilde{P}_{\mathrm{CS}})$ it is convenient to assign a dual variable to every linear constraint. We assign $R_{\mathbf{i}}$ to the constraints in Eq. (3.25), y_0' to Eq. (3.26), y_0 to Eq. (3.27), and $y_{\mathbf{i},\mathbf{i}}$ to Eqs. (3.28) and (3.29). In addition, one should note that every variable in the primal corresponds to a restriction in the dual. With this in mind, from the definition of the dual given in Eq. (2.15), it follows that $(\widetilde{D}_{\mathrm{CS}})$ is the dual of $(\widetilde{P}_{\mathrm{CS}})$, and that the constraints of Eq. (3.30) correspond to variable Z in $(\widetilde{P}_{\mathrm{CS}})$, Eq. (3.31) to varible w, and Eqs. (3.32) to (3.34) to variable Y.

Finally, we show that the conditions of Theorem 2.20 are satisfied by $(\widetilde{P}_{\mathrm{CS}})$ and $(\widetilde{D}_{\mathrm{CS}})$, which implies that their values are equal. $(\widetilde{P}_{\mathrm{CS}})$ is feasible, as every T factors as in Eq. (3.1) for some u, v with sufficiently large norm (if this was not true, $\|T\|_{\mathrm{cb}}$ would not be a norm). In addition, we claim that the following parameters define a strictly positive feasible instance for $(\widetilde{D}_{\mathrm{CS}})$

$$
\begin{aligned}
y_0 = y_0' &= \frac{1}{3}, \\
y_{\mathbf{i},\mathbf{j}} &= \frac{\delta_{\mathbf{i},\mathbf{j}}}{3(n+1)^s}, \text{ for } \mathbf{i},\mathbf{j} \in [n]^s, \ s \in [t], \\
R_{\mathbf{i}} &= 0, \text{ for } \mathbf{i} \in [n]^t.
\end{aligned}
$$

Indeed, with these parameters Eqs. (3.30) to (3.34) read as follows:

$$
\begin{aligned}
&\frac{1}{3} \geq 0, \ \mathrm{Id} \succ 0 \\
&\frac{1}{3} + \frac{1}{3} \leq 1, \\
&\frac{1}{3} \succ \frac{n}{3(n+1)}, \\
&\frac{1}{3(n+1)^s}\mathrm{Id}_{n^s} \succ \frac{n}{3(n+1)^{s+1}}\mathrm{Id}_{n^s}, \text{ for } s \in [t-1], \\
&\begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3(n+1)^t}\mathrm{Id}_{n^t} \end{pmatrix} \succ 0,
\end{aligned}
$$

and these identities are true because $1 > n/(n+1)$. □

3.4.2 A hierarchy of SDPs to find quantum algorithms

To introduce the announced hierarchy of SDPs, we first note that by Theorem 3.6 it follows that the smallest error that can be achieved when approximating a function $f : D \subseteq \{-1,1\}^n \to \mathbb{R}$ with a t-query quantum algorithm is

$$
\begin{aligned}
\mathcal{E}(f,t) = \inf \big\{ \varepsilon \geq 0 \ | \ &\exists \ 2t\text{-linear form } T : \mathbb{R}^{2n} \times \cdots \times \times \mathbb{R}^{2n} \to \mathbb{R} \\
&|f(x) - T((x,1^n),\ldots,(x,1^n))| \leq \varepsilon \quad \forall x \in D, \\
&\|T\|_{\mathrm{cb}} \leq 1 \}.
\end{aligned}
$$

Now, an immediate corollary of Theorem 3.11 is the following formulation of $\mathcal{E}(p,t)$ as an SDP.

Corollary 3.13. *Let* $f : \{-1,1\}^n \to [-1,1]$ *and* $t \in \mathbb{N}$. *Then,*

$$\begin{aligned}
\mathcal{E}(f,t) = \inf \quad & \varepsilon \\
\text{s.t.} \quad & \varepsilon \geq 0,\ Y, Y' \succeq 0, \\
& |p(x) - \sum_{\mathbf{i} \in [n]^{2t}} Y'_{0,\mathbf{i}} y_{i_1} \dots y_{i_{2t}}| \leq \varepsilon, \qquad y = (x, 1^n), \forall\, x \in \{-1,1\}^n, \\
& Y'_{0,0} = w, \\
& \sum_{i \in [2n]} Y_{i,i} \leq w, \\
& \sum_{i \in [2n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}' \in [2n]^s} \preceq \oplus_{i \in [2n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}' \in [2n]^{s-1}},\ s \in [2t-1], \\
& (Y'_{\mathbf{j},\mathbf{j}'})_{\mathbf{j},\mathbf{j}' \in [2n]^{2t}} \preceq \oplus_{i \in [2n]} (Y_{i\mathbf{j},i\mathbf{j}'})_{\mathbf{j},\mathbf{j}' \in [2n]^{2t-1}},
\end{aligned}$$

We observe that, as a consequence of Corollary 3.13, we have that $(\mathcal{E}(f,t))_t$ determines a hierarchy of SDPs that computes quantum query complexity. Indeed, to compute $Q_\varepsilon(f)$ one can solve $\mathcal{E}(f,1),\ \mathcal{E}(f,2),\dots$ and stop at the smallest t_0 satisfying $\mathcal{E}(f,t_0) \leq \varepsilon$. Then, we will have that $t_0 = Q_\varepsilon(f)$. Additionally, from an optimal solution to $\mathcal{E}(f,t_0)$ one can obtain an optimal quantum algorithm. This can be easily (but tediously) done following the constructions in the proofs of Theorem 3.6 and Theorem 3.11 Item (i),

Comparison with other methods

There are other formulations of $\mathcal{E}(f,t)$ as a SDP: the aforementioned work by Gribling and Laurent [GL19] and by Barnum, Saks, and Szegedy [BSS03]. We will compare these three methods with ours, and also with the adversary method, which does not compute $\mathcal{E}(f,t)$, but provides a SDP that directly computes the quantum query complexity. We remark the following:

- The method of Gribling and Laurent does not provide a description of the approximating quantum algorithm, while the others method do.
- The sizes of the SDPs differ, as shown in Table 3.1. The ones of Corollary 3.13 are considerably smaller than the ones in [BSS03] and the size of the SDP of the adversary method, but they are slightly bigger than the ones in [GL19].
- The adversary method loses constant factors in the characterization of quantum query complexity, and it does not work for exact quantum query complexity. On

	# blocks	block size	# lin. ineq.	# lin. eq.
Adversary method [HLv07]	n	$\|D\|$	0	$\|f^{-1}(1)\|\|f^{-1}(0)\|$
Barnum-Saks-Szegedy [BSS03]	$nt+2$	$\|D\|$	$\|D\|$	$\Theta(t\|D\|^2)$
Gribling-Laurent [GL19]	1	$\Theta(n^t)$	$2\|D\|+1$	$\Theta(n^{2t})$
Corollary 3.13	$4t-2$	$\Theta((2n)^{2t})$	$2\|D\|+3$	$\Theta(2t(2n)^{2t})$

Table 3.1: A comparison of the sizes of the SDPs to compute quantum query complexity. We count the number of linear equalities, inequalities, and PSD blocks, keeping track of the size of the largest block.

the other hand, the other three hierarchies of SDPs do characterize quantum query complexity, including the exact case, without losing constant factors.