A Direct Reduction from the Polynomial to the Adversary Method

Aleksandrs Belovs
Faculty of Computing, University of Latvia, Riga, Latvia

Abstract

The polynomial and the adversary methods are the two main tools for proving lower bounds on query complexity of quantum algorithms. Both methods have found a large number of applications, some problems more suitable for one method, some for the other.

It is known though that the adversary method, in its general negative-weighted version, is tight for bounded-error quantum algorithms, whereas the polynomial method is not. By the tightness of the former, for any polynomial lower bound, there ought to exist a corresponding adversary lower bound. However, direct reduction was not known.

In this paper, we give a simple and direct reduction from the polynomial method (in the form of a dual polynomial) to the adversary method. This shows that any lower bound in the form of a dual polynomial is actually an adversary lower bound of a specific form.

2012 ACM Subject Classification Theory of computation → Quantum query complexity

Keywords and phrases Polynomials, Quantum Adversary Bound

Digital Object Identifier 10.4230/LIPIcs.TQC.2024.11

Funding *Aleksandrs Belovs*: This research is supported by the Latvian Quantum Initiative under European Union Recovery and Resilience Facility project no. 2.3.1.1.i.0/1/22/I/CFLA/001. Part of this work has been supported by the ERDF project number 1.1.1.5/18/A/020 "Quantum algorithms: from complexity theory to experiment."

Acknowledgements The author is thankful to Shalev Ben-David for the suggestion to apply this construction to the polynomial method, and to anonymous referees for their helpful comments on the previous versions of the paper.

1 Introduction

Proving lower bounds on quantum query complexity is a task that has attained significant attention. The reason is that it is essentially the only known way to prove limitations on the power of quantum algorithms. For instance, Bennett, Bernstein, Brassard, and Vazirani [14] proved a quantum query lower bound for the OR function using what later became known as the hybrid method. This demonstrates that there is no way to attain a better than Grover's [21] quadratic speed-up for an NP-search problem if we treat the latter as a black-box (an oracle). Powerful tools for proving quantum query lower bounds have been developed consequently: the polynomial method, and the adversary method, both in its original (positive-weighted) and improved (negative-weighted) formulations.

The polynomial method is due to Beals, Buhrman, Cleve, Mosca, and de Wolf [9], and it was inspired by a similar method used by Nisan and Szegedy [27, 28] to prove lower bounds on randomized query complexity. The method builds on the following observation: if $\mathcal{A}$ is a T-query quantum algorithm, then its acceptance probability on input x can be expressed as a degree-$2T$ multivariate polynomial in the input variables x_i. Beals et al. [9] used this method to re-prove the lower bound for the OR function from [14], and establish other results like a tight lower bound for all total symmetric Boolean functions. A landmark result obtained by this method is the lower bound for the collision problem by Aaronson and Shi [3]. Similarly

as Bennett et al.'s result [14], it shows that a black-box approach to finding a collision in a hash function by a quantum computer is doomed as well. This method has been popular ever after.

The original adversary method is due to Ambainis [4], and it is an improvement on the aforementioned hybrid method. The bound was strengthened by Ambainis himself [5] and Zhang [35] shortly afterwards. One of the appealing features of this method is its convenient combinatorial formulation, which resulted in a number of applications [7, 20, 15, 19]. However, the original formulation of the adversary bound was subject to several important limitations [35].

Partly in order to overcome these limitations, Høyer, Lee, and Špalek generalised the adversary bound in [22]. Departing from the semidefinite formulation of the original adversary bound by Barnum, Saks, and Szegedy [8], Høyer et al. showed that the same expression still yields a lower bound if one replaces non-negative entries by arbitrary real numbers. This *negative-weighted* formulation of the bound is strictly more powerful than the positive-weighted one, but it lacks the combinatorial convenience of the latter. The bound turned out to be useful for composed functions [22] and sum-problems [13, 12]. In a series of papers [31, 29, 30], Reichardt et al. surprisingly proved that the negative-weighted version of the bound is tight for bounded-error algorithms!

The polynomial method, on the other hand, is known to be non-tight. Ambainis [5] constructed a first super-linear separation between the two for total Boolean functions. This was later improved to an almost quartic separation by Aaronson, Ben-David, and Kothari [1], which is essentially tight [2]. For partial functions, the separations can be even more impressive [6].

The history of relationship between the adversary and the polynomial methods is rather interesting. For instance, the AND of ORs function allows for a very simple adversary lower bound [4], but its polynomial lower bound is more complicated and was only obtained more than a decade later. It was achieved independently by Sherstov [33], and Bun and Thaler [17] using the technique of dual polynomials [32]. The latter is the dual of an approximating polynomial in the sense of linear programming. Therefore, by strong duality, their optimal values are exactly equal, and every lower bound on polynomial degree can, in principle, be stated as a dual polynomial. The technique of dual polynomials has been used by Bun, Kothari, and Thaler [16] to prove strong lower bounds for a number of problems like k-distinctness, image size testing, and surjectivity. The first of them was later improved in [26]. Similarly strong adversary lower bounds for these problems are not known.

Since the adversary method is tight, for every polynomial lower bound, there ought to exist a similarly good adversary lower bound. However, a direct reduction was not known. In this paper, we prove a simple direct reduction, giving a mechanical way of converting every dual polynomial into an adversary lower bound of a specific form. We hope that this connection will give a better understanding of both techniques, and should enable their combined use, which could result in better lower bounds. Contrary to a large number of papers dealing with the general adversary method, all proofs in this paper are fairly elementary.

A related result is a direct reduction from the polynomial method to *multiplicative* adversary by Magnin and Roland [25], while we give a reduction to a more widely-used *additive* adversary. We also note that our construction has similarities to a recent powerful lower bound technique by Zhandry [34, 24]. It would be interesting to understand the connection between the two better.

The following result is the cornerstone of our reduction. Here, we consider the task of distinguishing whether an input $x \in [q]^n$ belongs to a set $X \subseteq [q]^n$ or $Y \subseteq [q]^n$. Informally, the result states that X and Y cannot be distinguished by a quantum query algorithm if

they are *perfectly* indistinguishable by assignments of a corresponding weight. Recall that an *assignment* is a function $\alpha\colon S \to [q]$ defined on a subset S of the set of indices $[n]$. We write $x \sim \alpha$ if $x \in [q]^n$ *agrees* with the assignment α, that is, $x_j = \alpha(j)$ for all $j \in S$. The weight $|\alpha|$ of the assignment is the size of its domain S. It is possible to have an empty assignment $\emptyset$ of zero weight, in which case, every input string agrees to it.

▶ **Theorem 1.** *Let $X, Y \subseteq [q]^n$ be sets of inputs, and μ and ν be probability distributions on X and Y, respectively. Assume that, for any assignment α of weight $\leq 2m$, we have the following perfect indistinguishability*

$$\Pr_{x\leftarrow\mu}[x \sim \alpha] = \Pr_{y\leftarrow\nu}[y \sim \alpha]. \tag{1}$$

That is, the probability that x agrees to α does not depend on whether x is sampled from μ or from ν. Then, the quantum query complexity of distinguishing X and Y is $\Omega(m)$.

The result itself is actually known. We will give two proofs in this paper. The first one in Section 2 uses the method of dual polynomials and it is purely for illustrative purposes. The second proof is new and is done using the adversary method. This is the main technical contribution of this paper. Let us give a very short outline here. The proof uses the following collection of vectors:

$$v_\alpha^X = \sum_{x\in X: x\sim\alpha} \sqrt{\mu_x}|x\rangle \qquad \text{and} \qquad v_\alpha^Y = \sum_{y\in Y: y\sim\alpha} \sqrt{\nu_y}|y\rangle,$$

where α is an assignment. By the indistinguishability, for every $k \leq m$, there exists a linear isometry $W_{\leq k}$ that maps v_α^X into v_α^Y for all α with $|\alpha| \leq k$. The adversary matrix is $\Gamma = \sum_{k=0}^{m-1} W_{\leq k}$. It is not hard to show it has norm m, and we prove that $\|\Gamma \circ \Delta_j\| \leq 1$ for all j. This proof is contained in Sections 3 and 4. In Section 3, we only consider the space $\mathbb{R}^X$, and in Section 4, we substitute $\mathbb{R}^X$ with $\mathbb{R}^Y$ using indistinguishability.

In Section 5, we show how to use this result to transform a dual polynomial into an adversary bound. The idea is that a dual polynomial gives probability distributions μ and ν on two sets $\widetilde{X}$ and $\widetilde{Y}$ that are "close" to X and Y and that satisfy the promise of Theorem 1. We first prove the lower bound in the form of the adversary for distinguishing two probability distributions from [11], as we think it is conceptually closer to the dual polynomial. Obtaining a standard worst-case adversary bound is also easy.

2 Preliminaries

For a positive integer m, let $[m]$ denote the set $\{1, 2, ..., m\}$. For a predicate P, we write 1_P to denote the indicator variable that is 1 if P is true, and 0 otherwise.

We consider partial functions $f\colon D \to \{0, 1\}$ with $D \subseteq [q]^n$. We denote $X = f^{-1}(1)$ and $Y = f^{-1}(0)$. Thus, the function f distinguishes X and Y. An element $x = (x_1, x_2, \ldots, x_n) \in [q]^n$, is called an *input*, the set $[q]$ is called the *input alphabet*, and $x_j \in [q]$ are individual input symbols.

A *measure* on a finite set X is a function μ from X to the set of non-negative real numbers. We denote the value of μ on $x \in X$ by μ_x. The measure is a *probability distribution* if $\sum_{x\in X} \mu_x = 1$. We use $x \leftarrow \mu$ to denote that x is sampled from the probability distribution μ.[1]

[1] A more standard notation is $x \sim \mu$. But since we use $x \sim \alpha$ for agreement with an assignment α, we opted to use a different piece of notation.

2.1 Linear Algebra

An $X \times Y$ matrix is a matrix with rows labelled by the elements of X and columns by the elements of Y. The element of an $X \times Y$ matrix A at the intersection of the x-th row and the y-th column is denoted by $A[\![x, y]\!]$. For $X' \subseteq X$ and $Y' \subseteq Y$, the matrix $A[\![X', Y']\!]$ is the restriction of A to the rows in X' and the columns in Y'. We use $\|A\|$ to denote spectral norm of matrices. We identify a subspace and the corresponding orthogonal projector, which we usually denote by Π with additional decorations. An *isometry* is a linear operator that preserves inner product. We need the following well-known result:

▶ **Lemma 2.** *Assume $\mathcal{H}$ and $\mathcal{K}$ are two inner-product spaces. Let $(v_i)_{i\in A} \subseteq \mathcal{H}$ and $(w_i)_{i\in A} \subseteq \mathcal{K}$ be two collections of vectors indexed by the same index set A. Assume $\langle v_i, v_j\rangle = \langle w_i, w_j\rangle$ for all $i, j \in A$. Then, there exists an isometry $T\colon \operatorname{span}_i v_i \to \operatorname{span}_i w_i$ such that $Tv_i = w_i$ for all i.*

2.2 Adversary Bound

We use two different flavours of the negative-weighted adversary bound. Here we give the canonical version from [22] and later we state the distributional version from [11].

Assume we want to distinguish two sets of inputs $X, Y \subseteq [q]^n$ as above. Let Γ be a real $X \times Y$ matrix. For $j \in [n]$, denote by $\Gamma \circ \Delta_j$ the matrix of the same dimensions as Γ whose (x, y)-th entry is given by $\Gamma[\![x, y]\!] \cdot 1_{x_j \neq y_j}$. In other words, the entries with $x_j = y_j$ are being erased (replaced by zeroes).

▶ **Theorem 3** ([22])**.** *Assume that Γ is an $X \times Y$ real matrix such that $\|\Gamma \circ \Delta_j\| \leq 1$ for all $j \in [n]$. Then, the (bounded-error) quantum query complexity of evaluating f is $\Omega(\|\Gamma\|)$.*

The matrix Γ from Theorem 3 is called the *adversary matrix*, and it is known that the bound of this theorem is tight [30].

As it can be guessed from the notation, the mapping $\Gamma \mapsto \Gamma \circ \Delta_j$ is usually expressed as an Hadamard product with a 01-matrix Δ_j of dimensions $X \times Y$. However, we find it more convenient to think of it as a mapping. In particular, we don't have to formally re-define the matrix Δ_j for matrices Γ of different dimensions, and the matrix Δ_j almost never appears by itself.

The norm of the matrix $\Gamma \circ \Delta_j$ is not always easy to estimate. The following trick from [12] based on [23, Fact 2.4] is of help here. With some stretch of notation, we write $\Gamma \overset{\Delta_j}{\longmapsto} B$ if $(\Gamma - B) \circ \Delta_j = 0$. In other words, we are allowed to arbitrary change the (x, y)-entries of Γ with $x_j = y_j$ in order to obtain B. Note that this is a relation, since B is not uniquely defined by Γ. The idea is as follows:

▶ **Proposition 4.** *For any B with $\Gamma \overset{\Delta_j}{\longmapsto} B$, we have $\|\Gamma \circ \Delta_j\| \leq 2\|B\|$. Moreover, if f is a Boolean function, i.e., $D \subseteq \{0, 1\}^n$, then $\|\Gamma \circ \Delta_j\| \leq \|B\|$.*

Hence, we can bound $\|\Gamma \circ \Delta_j\|$ from above by estimating $\|B\|$, which is often easier. We repeatedly use the following easy properties of thus defined relation Δ_j:

▶ **Proposition 5.** *For any $X \times Y$ matrices A, B, C, D and real numbers a and c, we have*

(a) *$A \overset{\Delta_j}{\longmapsto} A$ and $A \overset{\Delta_j}{\longmapsto} A \circ \Delta_j$;*

(b) *if $A \overset{\Delta_j}{\longmapsto} B$ and $C \overset{\Delta_j}{\longmapsto} D$, then $aA + cC \overset{\Delta_j}{\longmapsto} aB + cD$.*

2.3 Distributional Adversary

We also use the version of the adversary bound for distinguishing two probability distributions. This version is rather versatile as it allows the probability distributions to overlap and to have arbitrary acceptance probabilities.

▶ **Theorem 6** ([11]). *Assume $\mathcal{A}$ is a quantum algorithm that makes T queries to the input string $x = (x_1, \dots, x_n) \in [q]^n$, and performs a measurement at the end with two outcomes: "accept" or "reject". Let μ and ν be two probability distributions on $[q]^n$, and denote by s_μ and s_ν the acceptance probability of $\mathcal{A}$ when x is sampled from μ and ν, respectively. Then,*

$$T = \Omega\left(\min_{j\in[n]} \frac{\delta_\mu^* \Gamma \delta_\nu - \tau(s_\mu, s_\nu)\|\Gamma\|}{\|\Gamma \circ \Delta_j\|}\right), \tag{2}$$

for any $[q]^n \times [q]^n$ matrix Γ with real entries. Here,

$$\delta_\mu[\![x]\!] = \sqrt{\mu_x} \qquad \text{and} \qquad \delta_\nu[\![y]\!] = \sqrt{\nu_y} \tag{3}$$

are unit vectors in $\mathbb{R}^{[q]^n}$, and

$$\tau(s_\mu, s_\nu) = \sqrt{s_\mu s_\nu} + \sqrt{(1-s_\mu)(1-s_\nu)} \le 1 - \frac{|s_\mu - s_\nu|^2}{8}. \tag{4}$$

2.4 Polynomials

In the polynomial method, we have to assume that the function $f\colon D \to \{0,1\}$ is Boolean: $D \subseteq \{0,1\}^n$. If this does not hold, one has to make the function Boolean. A popular option is to introduce new variables $\widetilde{x_{i,a}}$ with $i \in [n]$ and $a \in [q]$, defined by $\widetilde{x_{i,a}} = 1_{x_i = a}$.

For $S \subseteq [n]$, the corresponding *character* is the function $\chi_S\colon \{0,1\}^n \to \{\pm 1\}$ defined by $\chi_S(x) = \prod_{i\in S}(-1)^{x_i}$. The characters form a basis of the space of functions $\mathbb{R}^{\{0,1\}^n}$. Hence, every function $f\colon \{0,1\}^n \to \mathbb{R}$ has a unique representation as a (multilinear) *polynomial*: $f = \sum_{S\subseteq[n]} \alpha_S \chi_S$. The size of the largest S with non-zero α_S is called the *degree* of f.

A degree-d *polynomial* is any function $p\colon \{0,1\}^n \to \mathbb{R}$ of degree at most d. A degree-d *dual polynomial* is a function $\phi\colon \{0,1\}^n \to \mathbb{R}$ satisfying

$$\sum_{x\in\{0,1\}^n} |\phi(x)| = 1 \qquad \text{and} \qquad \sum_{x\in\{0,1\}^n} \phi(x)\chi_S(x) = 0 \quad \text{for all } |S| \le d. \tag{5}$$

It is easy to check that the second condition above is equivalent to the following one:

$$\sum_{x\sim\alpha} \phi(x) = 0 \quad \text{for all assignments } \alpha \text{ with } |\alpha| \le d. \tag{6}$$

Dual polynomials [32] can be used to show inapproximability for real-valued total functions. We may assume $d < n$, since every function can be represented by a degree-d polynomial.

The proof of the following theorem, as well as that of Theorem 10 are based on linear programming duality, and are given in Appendix A for completeness.

▶ **Theorem 7.** *Let $d < n$. For any function $f\colon \{0,1\}^n \to \mathbb{R}$, we have*

$$\min_p \max_{x\in\{0,1\}^n} |f(x) - p(x)| = \max_\phi \sum_{x\in\{0,1\}^n} \phi(x) f(x), \tag{7}$$

where p ranges over all degree-d polynomials and ϕ ranges over all degree-d dual polynomials.

Let us now turn to the case of partial functions $f\colon D \to \{0,1\}$ with $D \subseteq \{0,1\}^n$. Again, we let $X = f^{-1}(1)$ and $Y = f^{-1}(0)$.

▶ **Definition 8.** We say that a polynomial $p\colon \{0,1\}^n \to \mathbb{R}$ ε-approximates a partial function $f\colon D \to \{0,1\}$ with $D \subseteq \{0,1\}^n$ if

- for every $x \in D$, we have $|p(x) - f(x)| \leq \varepsilon$;
- for every $x \in \{0,1\}^n$, we have $0 \leq p(x) \leq 1$.

The importance of this definition stems from the following result:

▶ **Theorem 9** ([9])**.** *If a partial function $f\colon D \to \{0,1\}$ with $D \subseteq \{0,1\}^n$ can be evaluated by a T-query quantum algorithm with error at most ε, then f can be ε-approximated by a polynomial of degree at most $2T$.*

The corresponding analogue of Theorem 7 is slightly more involved. A similar result previously appeared in [18].

▶ **Theorem 10.** *The best approximation distance ε as in Definition 8 of the function f by a degree-d polynomial is given by*

$$\max\left\{\max_{\phi}\left(\sum_{x\in X}\phi^+(x) - \sum_{x\notin Y}\phi^-(x)\right), 0\right\}, \tag{8}$$

where the maximisation is over functions $\phi\colon \{0,1\}^n \to \mathbb{R}$ satisfying

$$\sum_{x\in X}\phi^+(x) + \sum_{x\in Y}\phi^-(x) = 1 \qquad \text{and} \qquad \sum_{x\in\{0,1\}^n}\phi(x)\chi_S(x) = 0 \quad \text{for all } |S| \leq d. \tag{9}$$

Here $\phi^+(x) = \max\{0, \phi(x)\}$ and $\phi^-(x) = \max\{0, -\phi(x)\}$ are the positive and the negative parts of ϕ, respectively. We will still call ϕ a degree-d dual polynomial in this case, although it need not satisfy the first (normalisation) condition of (5).

Proof of Theorem 1 using Dual Polynomials. We may assume the function f is Boolean. It suffices to show that it cannot be approximated by a polynomial of degree less than $2m$. Let

$$\phi(x) = \begin{cases} \mu_x/2, & \text{if } x \in X; \\ -\nu_x/2, & \text{if } x \in Y; \\ 0, & \text{otherwise.} \end{cases}$$

This function satisfies (9) with $d = 2m$. Indeed, the first condition follows from μ and ν being probability distributions, and the second one follows from (6) since

$$\sum_{x\sim\alpha}\phi(x) = \frac{1}{2}\Pr_{x\leftarrow\mu}[x \sim \alpha] - \frac{1}{2}\Pr_{y\leftarrow\nu}[y \sim \alpha] = 0$$

by (1). The value of (8) is $1/2$, hence, by Theorem 10, it is impossible to get a better than trivial approximation. ◀

3 Δ-decomposition of $\mathbb{R}^X$

Let $X \subseteq [q]^n$ be a set of inputs, and let μ be some measure on X. We assume in this section that X is the support of μ, i.e., $\mu_x > 0$ for all $x \in X$. The goal of this section is to develop a decomposition of the space $\mathbb{R}^X$ convenient for the Δ_j operation and that takes into account the measure μ. Let us remind that we use the same notation, like $\Pi_{\leq k}$, to denote *both* the subspace and the corresponding orthogonal projector.

3.1 Definition of subspaces

For each assignment α, define the following vector in $\mathbb{R}^X$:

$$v_\alpha = \sum_{x\sim\alpha} \sqrt{\mu_x}|x\rangle. \tag{10}$$

Based on these vectors, we define a number of subspaces. First, for $k \in \{0,1,\dots,n\}$:

$$\Pi_{\le k} = \operatorname*{span}_{\alpha\colon |\alpha|=k} v_\alpha. \tag{11}$$

▷ Claim 11. We have $\Pi_{\le k-1} \subseteq \Pi_{\le k}$ and $\Pi_{\le n} = \mathbb{R}^X$.

Proof. Let α be an assignment of weight $k-1$, and i be an element of $[n]$ outside the domain of α. Then,

$$v_\alpha = \sum_{a\in[q]} v_{\alpha\cup\{i\mapsto a\}},$$

proving the first claim.

For the second claim, note that an assignment α of weight n defines an individual input. ◁

This gives an orthogonal decomposition of $\mathbb{R}^X$ into subspaces

$$\Pi_k = \Pi_{\le k} \cap \Pi_{\le k-1}^\perp = \Pi_{\le k} - \Pi_{\le k-1}.$$

3.2 Example

A simple example is $X = [q]^n$ with the uniform distribution μ_x. Define two orthogonal projectors on $\mathbb{R}^q$: $E_0 = J_q/q$ and $E_1 = I_q - E_0$, where J_q is the all-1 matrix. Then,

$$\Pi_k = \sum_{s\in\{0,1\}^n\colon |s|=k} E_{s_1}\otimes E_{s_2}\otimes\cdots\otimes E_{s_n},$$

where $|s|$ is the Hamming weight. These operators are similar to the ones used in the construction of the adversary lower bound for element distinctness [10] and sum-problems [12].

Note that while the vectors v_α in (10) only have non-negative entries, the projectors $\Pi_{\le k}$ can have negative entries. For instance, such are matrices $\Pi_{\le 1}$ in the above example.

3.3 Action of Δ_j

Let us consider the action of Δ_j with $j \in [n]$. For that, we define the following variant of the above subspaces $\Pi_{\le k}$:

$$\Xi_{\le k,\circ j} = \operatorname*{span}_{\alpha\colon |\alpha|=k,\ \alpha \text{ defined on } j} v_\alpha.$$

In particular, we again have $\Xi_{\le n,\circ j} = \mathbb{R}^X$. This time, however, $\Xi_{\le 0,\circ j}$ is the empty subspace.

▷ Claim 12. We have the following:

(a) $\Xi_{\le k-1,\circ j} \subseteq \Xi_{\le k,\circ j}$;

(b) $\Pi_{\le k-1} \subseteq \Xi_{\le k,\circ j} \subseteq \Pi_{\le k}$;

(c) $\Delta_j \circ \Xi_{\le k,\circ j} = 0$.

Proof. The proof of (a) is analogous to the proof of Claim 11.

The second inclusion of (b) holds because $\Xi_{\leq k,\circ j}$ is a span of a subset of vectors of $\Pi_{\leq k}$. To prove the first inclusion of (b), it suffices to show that an arbitrary v_α with $|\alpha| = k-1$ is contained in $\Xi_{\leq k,\circ j}$. The proof of that is analogous to the proof of Claim 11. However, this time we take $i = j$ if α is not defined on j (and an arbitrary i as before, otherwise).

Now let us prove (c). Note that $\Xi_{\leq k,\circ j}$ can be written as a direct sum

$$\Xi_{\leq k,\circ j} = \bigoplus_{b\in[q]} \Xi_{\leq k,j\mapsto b} \tag{12}$$

of orthogonal projectors

$$\Xi_{\leq k,j\mapsto b} = \operatorname*{span}_{\alpha\colon |\alpha|=k,\ \alpha(j)=b} v_\alpha.$$

Each $\Xi_{\leq k,j\mapsto b}$ acts on the subspace spanned by $x \in X$ with $x_j = b$. Hence, $\Delta_j \circ \Xi_{\leq k,j\mapsto b} = 0$. By linearity, $\Delta_j \circ \Xi_{\leq k,\circ j} = 0$. ◁

3.4 Standard Form of Adversary

As a warm-up for the next sections, we describe the following "standard" form of the "adversary" matrix on $\mathbb{R}^X$:

$$\sum_{k=0}^{m-1} \Pi_{\leq k} = \sum_{k=0}^{m} (m-k)\Pi_k. \tag{13}$$

Clearly, the norm of this matrix is m. The action of Δ_j is defined as

$$\sum_{k=0}^{m-1} \Pi_{\leq k} \overset{\Delta_j}{\longmapsto} \sum_{k=0}^{m-1} \big(\Pi_{\leq k} - \Xi_{\leq k,\circ j}\big), \tag{14}$$

where we use Proposition 5 and point (c) of Claim 12.

▷ Claim 13. The norm of the operator on the right-hand side of (14) is 1.

Proof. The operator in question is the sum of projectors $\Pi_{\leq k} - \Xi_{\leq k,\circ j}$. By point (b) of Claim 12, we know that $\Pi_{\leq k}$ is contained in $\Xi_{\leq k+1,\circ j}$. Hence, these projectors are pairwise orthogonal, and the norm of the operator is 1. ◁

In the following section, we will transfer this construction for $X \times Y$-matrices.

4 Second Proof of Theorem 1

Here, we give a proof of Theorem 1, which is based on the adversary method. In this section, we use upper indices X and Y in the following way. If the upper index X is used, the corresponding object is equal to the one without the upper index as defined in Section 3. If the upper index Y is used, we use the same object but with the probability distribution ν on Y instead of μ on X. For example:

$$v^X_\alpha = \sum_{x\in X\colon x\sim\alpha} \sqrt{\mu_x}|x\rangle \in \mathbb{R}^X \qquad \text{and} \qquad v^Y_\alpha = \sum_{y\in Y\colon y\sim\alpha} \sqrt{\nu_y}|y\rangle \in \mathbb{R}^Y.$$

Similarly, $\Pi^X_{\leq k}$ and $\Xi^X_{\leq k,\circ j}$ are projectors in $\mathbb{R}^X$, and $\Pi^Y_{\leq k}$ and $\Xi^Y_{\leq k,\circ j}$ are projectors in $\mathbb{R}^Y$.

▶ **Lemma 14.** *In the assumptions of Theorem 1, there exists a linear isometry* $W\colon \Pi^Y_{\leq m} \to \Pi^X_{\leq m}$ *that maps* v^Y_α *into* v^X_α *for each* $|\alpha| \leq m$.

Proof. Let α and β be assignments of weight at most m. Note that

$$\left\langle v^X_\alpha, v^X_\beta \right\rangle = \Pr_{x\sim\mu}[x \sim \alpha \wedge x \sim \beta] = \Pr_{y\sim\nu}[y \sim \alpha \wedge y \sim \beta] = \left\langle v^Y_\alpha, v^Y_\beta \right\rangle.$$

Indeed, either α and β contradict each other, in which case the both sides of the above equality are zero, or they can be merged into one assignment of weight at most $2m$, in which case (1) applies. Hence, by Lemma 2, there exists a linear isometry W that maps v^Y_α into v^X_α for each $|\alpha| \leq m$. ◀

The following theorem defines the adversary matrix Γ which, when plugged into Theorem 3, gives Theorem 1. This matrix will be important in the next section for the reduction from the polynomial method.

▶ **Theorem 15.** *In the assumptions of Theorem 1, the following* $X \times Y$ *matrix*

$$\Gamma = W\left(\sum_{k=0}^{m-1} \Pi^Y_{\leq k}\right). \tag{15}$$

has the following properties:

(a) *its norm is* m*, as witnessed by* $\|\Gamma\| = \delta^*_\mu \Gamma \delta_\nu = m$ *with* δ_μ *and* δ_ν *as defined in (3); and*

(b) *the action of* Δ_j *is given by*

$$\Gamma \overset{\Delta_j}{\longmapsto} W\left(\sum_{k=0}^{m-1} \left(\Pi^Y_{\leq k} - \Xi^Y_{\leq k,\circ j}\right)\right), \tag{16}$$

where the norm of the matrix on the right-hand side is 1.

Proof. Eq. (13) gives the decomposition of $\sum_{k=0}^{m-1} \Pi^Y_{\leq k}$ into eigenspaces since the subspaces Π^Y_k are pairwise orthogonal. The maximal eigenvalue m is achieved on Π^Y_0, which is spanned by $v^Y_\emptyset = \delta_\nu$, where $\emptyset$ denotes the empty assignment. Also, W is an isometry that maps $v^Y_\emptyset$ into $v^X_\emptyset = \delta_\mu$, which proves point (a) of the theorem.

The validity of the action of Δ_j in (16) follows from the claim that $\Delta_j \circ (W\Xi^Y_{\leq k,\circ j}) = 0$. The proof of this claim is similar to the point (c) of Claim 12. We use decomposition (12) for $\Xi^Y_{\leq k,\circ j}$, and observe that the range of $W\Xi^Y_{\leq k,j\mapsto b}$ is $\Xi^X_{\leq k,j\mapsto b}$. Hence, $\Delta_j \circ (W\Xi^Y_{\leq k,j\mapsto b}) = 0$, and the first half of point (b) follows by linearity.

The second half of point (b) follows from Claim 13 and the fact that W is an isometry. ◀

5 Reduction from Dual Polynomial to Adversary

In this section, we demonstrate a direct conversion of a polynomial lower bound into an adversary lower bound. We do so by taking a dual polynomial that witnesses degree at least d and convert it into an adversary bound of value $\Omega(d)$.

For warm-up, we consider the case of total functions in Section 5.1, and then the general case of partial functions in Section 5.2. In both cases, we use the distributional version of the adversary bound, Theorem 6, which we find conceptually more appropriate in this case. However, it is not hard to reduce to the usual version of the bound, Theorem 3, as well, which we do in Section 5.3.

5.1 Total Functions

We start with the case when $f\colon \{0,1\}^n \to \{0,1\}$ is a total Boolean function. Assume it cannot be 1/3-approximated by a polynomial of degree d. In this case, we can use Theorem 7. Let ϕ be a degree-d dual polynomial attaining the maximum in (7). Thus,

$$\sum_{x\in\{0,1\}^n} \phi(x)f(x) \geq 1/3. \tag{17}$$

Our goal is to prove an adversary lower bound of $\Omega(d)$.

Let us define

$$\widetilde{X} = \{x \in \{0,1\}^n \mid \phi(x) \geq 0\} \qquad \text{and} \qquad \widetilde{Y} = \{y \in \{0,1\}^n \mid \phi(y) < 0\}, \tag{18}$$

and two measures

$$\mu\colon \widetilde{X} \to \mathbb{R},\ x \mapsto 2\phi(x) \qquad \text{and} \qquad \nu\colon \widetilde{Y} \to \mathbb{R},\ y \mapsto -2\phi(y).$$

From (6) applied to empty α, we get that $\sum_x \phi(x) = 0$. Also, $\sum_x |\phi(x)| = 1$. Hence,

$$\sum_{x\in\widetilde{X}} \mu_x = \sum_{y\in\widetilde{Y}} \nu_y = 1, \tag{19}$$

that is, both μ and ν are probability distributions.

Using (6) again, we get that for each assignment α of weight at most d, we have

$$\Pr_{x\leftarrow\mu}[x \sim \alpha] = \Pr_{y\leftarrow\nu}[y \sim \alpha]. \tag{20}$$

Thus, by Theorem 1, the quantum query complexity of distinguishing $\widetilde{X}$ and $\widetilde{Y}$ is $\Omega(d)$. This is a nice development, but we would really like to prove the same result for the sets $X = f^{-1}(1)$ and $Y = f^{-1}(0)$. Luckily, by condition (17), these sets are sufficiently well correlated.

Define the $\widetilde{X} \times \widetilde{Y}$ matrix $\widetilde{\Gamma}$ as in (15) with the sets $\widetilde{X}$ and $\widetilde{Y}$, the distributions μ and ν, and $m = d/2$. By Theorem 15, we have

$$\|\widetilde{\Gamma}\| = \delta_\mu^* \widetilde{\Gamma} \delta_\nu = d/2, \qquad \text{and} \qquad \|\widetilde{\Gamma} \circ \Delta_j\| \leq 1 \quad \text{for all } j \in [n]. \tag{21}$$

Since Theorem 6 requires an $\{0,1\}^n \times \{0,1\}^n$ adversary matrix, we extend $\widetilde{\Gamma}$ with zeroes to fit this requirement.

▶ **Proposition 16.** *In the above notations, Theorem 6 with the adversary matrix $\widetilde{\Gamma}$ and the distributions μ and ν gives an $\Omega(d)$ lower bound on the number of queries made by any quantum query algorithm $\mathcal{A}$ that distinguishes X and Y with error probability at most $1/6$.*

Proof. Note that (17) is equivalent to

$$\sum_{x\in f^{-1}(1)} \mu_x - \sum_{y\in f^{-1}(1)} \nu_y \geq 2/3.$$

This is the difference between the "ideal" acceptance probabilities of $\mathcal{A}$ on μ and ν, i.e, in the hypothetical case when the algorithm never errs. Since the actual error of the algorithm $\mathcal{A}$ is at most $1/6$, we get

$$s_\mu - s_\nu \geq 1/3$$

in notations of Theorem 6. From (4), we get that $\tau(s_\mu, s_\nu) \leq 1 - \Omega(1)$. Pluging this and (21) into (2), we get that the query complexity of $\mathcal{A}$ is $\Omega(d)$. ◀

5.2 Partial Functions

Now let us consider the case of partial functions $f\colon D \to \{0,1\}$ with $D \subseteq \{0,1\}^n$. Again, we assume that the best optimisation distance by a degree-d polynomial is more than $1/3$. Let ϕ be the optimal degree-d dual polynomial from Theorem 10. Then we have from (8):

$$\sum_{x\in X} \phi^+(x) - \sum_{x\notin Y} \phi^-(x) \geq 1/3. \tag{22}$$

The sets $\widetilde{X}$ and $\widetilde{Y}$ are still defined as in (18). By (9), we still have that

$$\sum_{x\in\{0,1\}^n} \phi^+(x) = \sum_{x\in\{0,1\}^n} \phi^-(x).$$

But, in order to define μ and ν, we have to choose a different scaling factor. We have

$$\sum_{x\in\{0,1\}^n} \phi^-(x) = \sum_{x\in Y} \phi^-(x) + \sum_{x\notin Y} \phi^-(x) \leq \sum_{x\in Y} \phi^-(x) + \sum_{x\in X} \phi^+(x) - 1/3 = 2/3,$$

where we used (22) and the first condition from (9). Let us denote the left-hand side of the above inequality by M. Then, we can define probability distributions

$$\mu\colon \widetilde{X} \to \mathbb{R},\ x \mapsto \phi(x)/M \qquad \text{and} \qquad \nu\colon \widetilde{Y} \to \mathbb{R},\ y \mapsto -\phi(y)/M. \tag{23}$$

So that (22) becomes

$$\sum_{x\in X} \mu_x - \sum_{y\notin Y} \nu_y \geq 1/2. \tag{24}$$

The equation (20) still holds, and we use the same construction of $\widetilde{\Gamma}$, which still satisfies (21).

Let $\mathcal{A}$ be an algorithm that evaluates f with error ε. Denote by p_x the acceptance probability of the algorithm on an input $x \in \{0,1\}^n$. So, we have $p_x \geq 1-\varepsilon$ for $x \in X$, $p_x \leq \varepsilon$ for $x \in Y$, and $0 \leq p_x \leq 1$ for all x. Thus,

$$\begin{aligned} s_\mu - s_\nu &= \sum_{x\in X} \mu_x p_x + \sum_{x\notin X} \mu_x p_x - \sum_{y\in Y} \nu_y p_y - \sum_{y\notin Y} \nu_y p_y \\ &\geq (1-\varepsilon)\sum_{x\in X} \mu_x - \varepsilon \sum_{y\in Y} \nu_y - \sum_{y\notin Y} \nu_y \geq \sum_{x\in X} \mu_x - \sum_{y\notin Y} \nu_y - 2\varepsilon \geq 1/2 - 2\varepsilon \geq 1/4, \end{aligned}$$

assuming $\varepsilon \leq 1/8$.

In the same way as in Section 5.1, Theorem 6 implies that the query complexity of $\mathcal{A}$ is $\Omega(d)$.

5.3 Usual Version of the Adversary

In this section, we obtain a usual version of the adversary bound from a dual polynomial. Let us recap the construction.

We assume $f\colon D \to \{0,1\}$ with $D \subseteq \{0,1\}^n$ is a partial Boolean function, where we define $X = f^{-1}(1)$ and $Y = f^{-1}(0)$. Assume ϕ is a degree-d dual polynomial that satisfies (9) of Theorem 10 and attains value at least $1/3$ in (8).

Let $\widetilde{X}, \widetilde{Y} \subseteq \{0,1\}^n$ be as in (18), and μ and ν be probability distributions in (23). They satisfy (20), therefore, we can apply Theorem 15 with $m = d/2$, and obtain an $\widetilde{X} \times \widetilde{Y}$-matrix $\widetilde{\Gamma}$ as in (15) for the sets $\widetilde{X}$ and $\widetilde{Y}$ with the probability distributions μ and ν on them. This matrix satisfies (21). We extend it with zeroes to form an $\{0,1\}^n \times \{0,1\}^n$-matrix, which we still denote $\widetilde{\Gamma}$.

▶ **Theorem 17.** *In the above assumptions, the $X \times Y$-matrix*

$$\Gamma = \widetilde{\Gamma}[\![X, Y]\!]$$

satisfies $\|\Gamma\| = \Omega(d)$ and $\|\Gamma \circ \Delta_j\| \leq 1$ for all $j \in [n]$.

Proof. As Γ is a sub-matrix of $\widetilde{\Gamma}$, we get $\|\Gamma \circ \Delta_j\| \leq 1$ for all j from (21). By the same equation (21), it suffices to show that $\|\Gamma\| = \Omega\big(\|\widetilde{\Gamma}\|\big)$.

We know that $\widetilde{\Gamma}\delta_\nu = \|\widetilde{\Gamma}\|\delta_\mu$ by point (a) of Theorem 15. This gives us

$$\Big\|\widetilde{\Gamma}[\![X, \{0,1\}^n]\!]\ \delta_\nu\Big\| = \|\widetilde{\Gamma}\| \cdot \|\delta_\mu[\![X]\!]\|.$$

On the other hand,

$$\Big\|\widetilde{\Gamma}[\![X, \{0,1\}^n]\!]\ \delta_\nu\Big\| \leq \Big\|\widetilde{\Gamma}[\![X, Y]\!]\ \delta_\nu[\![Y]\!]\Big\| + \Big\|\widetilde{\Gamma}[\![X, \overline{Y}]\!]\ \delta_\nu[\![\overline{Y}]\!]\Big\| \leq \|\widetilde{\Gamma}[\![X, Y]\!]\| + \|\widetilde{\Gamma}\| \cdot \|\delta_\nu[\![\overline{Y}]\!]\|,$$

where $\overline{Y} = \{0,1\}^n \setminus Y$. Thus,

$$\|\widetilde{\Gamma}[\![X, Y]\!]\| \geq \|\widetilde{\Gamma}\|\Big(\|\delta_\mu[\![X]\!]\| - \|\delta_\nu[\![\overline{Y}]\!]\|\Big) = \|\widetilde{\Gamma}\|\frac{\|\delta_\mu[\![X]\!]\|^2 - \|\delta_\nu[\![\overline{Y}]\!]\|^2}{\|\delta_\mu[\![X]\!]\| + \|\delta_\nu[\![\overline{Y}]\!]\|}.$$

From (24), we get that

$$\|\delta_\mu[\![X]\!]\|^2 - \|\delta_\nu[\![\overline{Y}]\!]\|^2 = \sum_{x \in X} \mu_x - \sum_{y \notin Y} \nu_y \geq 1/2.$$

Also, $\|\delta_\mu[\![X]\!]\| + \|\delta_\nu[\![\overline{Y}]\!]\| \leq 2$, hence, we obtain

$$\|\widetilde{\Gamma}[\![X, Y]\!]\| \geq \frac{1}{4}\|\widetilde{\Gamma}\|,$$

as required. ◀

References

1. Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *Proc. of 48th ACM STOC*, pages 863–876, 2016. doi:10.1145/2897518.2897644.
2. Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. Degree vs. approximate degree and quantum implications of Huang's sensitivity theorem. In *Proc. of 53rd ACM STOC*, pages 1330–1342, 2021. doi:10.1145/3406325.3451047.
3. Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004. doi:10.1145/1008731.1008735.
4. Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. doi:10.1006/jcss.2002.1826.
5. Andris Ambainis. Polynomial degree vs. quantum query complexity. In *Proc. of 44th IEEE FOCS*, pages 230–239, 2003. doi:10.1109/SFCS.2003.1238197.
6. Andris Ambainis and Aleksandrs Belovs. An exponential separation between quantum query complexity and the polynomial degree. In *Proc. of 38th IEEE CCC*, volume 264 of *LIPIcs*, pages 24:1–24:13. Dagstuhl, 2023. doi:10.4230/LIPIcs.CCC.2023.24.
7. Howard Barnum and Michael Saks. A lower bound on the quantum query complexity of read-once functions. *Journal of Computer and System Sciences*, 69(2):244–258, 2004. doi:10.1016/j.jcss.2004.02.002.

8 Howard Barnum, Michael Saks, and Mario Szegedy. Quantum decision trees and semi-definite programming. In *Proc. of 18th IEEE CCC*, pages 179–193, 2003. doi:10.1109/CCC.2003.1214419.

9 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. doi:10.1145/502090.502097.

10 Aleksandrs Belovs. Adversary lower bound for element distinctness, 2012. arXiv:1204.5074.

11 Aleksandrs Belovs, Gilles Brassard, Peter Høyer, Marc Kaplan, Sophie Laplante, and Louis Salvail. Provably secure key establishment against quantum adversaries. In *Proc. of 12th TQC*, volume 73 of *LIPIcs*, pages 3:1–3:17. Dagstuhl, 2018. doi:10.4230/LIPIcs.TQC.2017.3.

12 Aleksandrs Belovs and Ansis Rosmanis. On the power of non-adaptive learning graphs. *Computational Complexity*, 23(2):323–354, 2014. doi:10.1007/s00037-014-0084-1.

13 Aleksandrs Belovs and Robert Špalek. Adversary lower bound for the k-sum problem. In *Proc. of 4th ACM ITCS*, pages 323–328, 2013. doi:10.1145/2422436.2422474.

14 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. doi:10.1137/S0097539796300933.

15 Harry Buhrman and Robert Špalek. Quantum verification of matrix products. In *Proc. of 17th ACM-SIAM SODA*, pages 880–889, 2006. arXiv:quant-ph/0409035.

16 Mark Bun, Robin Kothari, and Justin Thaler. The polynomial method strikes back: Tight quantum query bounds via dual polynomials. In *Proc. of 50th ACM STOC*, pages 297–310, 2018. doi:10.1145/3188745.3188784.

17 Mark Bun and Justin Thaler. Dual lower bounds for approximate degree and Markov-Bernstein inequalities. *Information and Computation*, 243:2–25, 2015. doi:10.1016/j.ic.2014.12.003.

18 Mark Bun and Justin Thaler. Dual polynomials for collision and element distinctness. *Theory of Computing*, 12(16):1–34, 2016. doi:10.4086/toc.2016.v012a016.

19 Sebastian Dörn and Thomas Thierauf. The quantum query complexity of algebraic properties. In *Proc. of 16th FCT*, volume 4639 of *LNCS*, pages 250–260. Springer, 2007. doi:10.1007/978-3-540-74240-1_22.

20 Christoph Dürr, Mark Heiligman, Peter Høyer, and Mehdi Mhalla. Quantum query complexity of some graph problems. In *Proc. of 31st ICALP*, volume 3142 of *LNCS*, pages 481–493. Springer, 2004. doi:10.1007/978-3-540-27836-8_42.

21 Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. of 28th ACM STOC*, pages 212–219, 1996. doi:10.1145/237814.237866.

22 Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proc. of 39th ACM STOC*, pages 526–535, 2007. doi:10.1145/1250790.1250867.

23 Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proc. of 52nd IEEE FOCS*, pages 344–353, 2011. doi:10.1109/FOCS.2011.75.

24 Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In *Proc. of 38th EUROCRYPT*, volume 11478 of *LNCS*, pages 189–218, 2019. doi:10.1007/978-3-030-17659-4_7.

25 Loïck Magnin and Jérémie Roland. Explicit relation between all lower bound techniques for quantum query complexity. In *Proc. of 30th STACS*, volume 20 of *LIPIcs*, pages 434–445. Dagstuhl, 2013. doi:10.4230/LIPIcs.STACS.2013.434.

26 Nikhil S. Mande, Justin Thaler, and Shuchen Zhu. Improved approximate degree bounds for k-distinctness. In *Proc. of 15th TQC*, volume 158 of *LIPIcs*, pages 2:1–2:22, 2020. doi:10.4230/LIPIcs.TQC.2020.2.

27 Noam Nisan. CREW PRAMs and decision trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991. doi:10.1137/0220062.

28 Noam Nisan and Mario Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994. doi:10.1007/BF01263419.

29 Ben W. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function. In *Proc. of 50th IEEE FOCS*, pages 544–551, 2009. doi:10.1109/FOCS.2009.55.

30 Ben W. Reichardt. Reflections for quantum query algorithms. In *Proc. of 22nd ACM-SIAM SODA*, pages 560–569, 2011. doi:10.1137/1.9781611973082.44.

31 Ben W. Reichardt and Robert Špalek. Span-program-based quantum algorithm for evaluating formulas. *Theory of Computing*, 8:291–319, 2012. doi:10.4086/toc.2012.v008a013.

32 Alexander A Sherstov. The pattern matrix method. *SIAM Journal on Computing*, 40(6):1969, 2011. doi:10.1137/080733644.

33 Alexander A. Sherstov. Approximating the and-or tree. *Theory of Computing*, 9(20):653–663, 2013. doi:10.4086/toc.2013.v009a020.

34 Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Proc. of 39th CRYPTO*, volume 11693 of *LNCS*, pages 239–268, 2019. doi:10.1007/978-3-030-26951-7_9.

35 Shengyu Zhang. On the power of Ambainis lower bounds. *Theoretical Computer Science*, 339(2):241–256, 2005. doi:10.1016/j.tcs.2005.01.019.

A Linear Programming for Dual Polynomials

A.1 Proof of Theorem 7

The left-hand side of (7) is equal to the optimal value of the following linear optimisation problem:

$$\begin{aligned}
&\text{minimise} && \varepsilon && \\
&\text{subject to} && f(x) - \sum_S \alpha_S \chi_S(x) \le \varepsilon && \text{for all } x \in \{0,1\}^n; && \text{(25a)}\\
& && f(x) - \sum_S \alpha_S \chi_S(x) \ge -\varepsilon && \text{for all } x \in \{0,1\}^n; && \text{(25b)}\\
& && \alpha_S \in \mathbb{R} && \text{for all } S \subseteq [n],\ |S| \le d; && \\
& && \varepsilon \in \mathbb{R}. && &&
\end{aligned}$$

Let us write the Lagrangian with the dual variables $a_x \ge 0$ for (25a) and $b_x \ge 0$ for (25b):

$$\varepsilon - \sum_x a_x \left(\varepsilon - f(x) + \sum_S \alpha_S \chi_S(x) \right) - \sum_x b_x \left(\varepsilon + f(x) - \sum_S \alpha_S \chi_S(x) \right) \tag{26}$$

Let us denote $\phi(x) = a_x - b_x$, so that we can rewrite the last expression as

$$\sum_x \phi(x) f(x) + \varepsilon \left(1 - \sum_x a_x - \sum_x b_x \right) - \sum_S \alpha_S \left(\sum_x \phi(x) \chi_S(x) \right). \tag{27}$$

In the dual optimisation problem, all of the brackets in (27) must be zero.

We can turn any dual polynomial into a feasible solution to the dual (27) by taking $a_x = \phi^+(x)$ and $b_x = \phi^-(x)$.

For the opposite direction, consider optimal primal and dual solutions, whose values are equal due to strong duality. If $\varepsilon > 0$, then, by complementary slackness, at most one of a_x and b_x is non-zero for each x, therefore, $|\phi(x)| = a_x + b_x$. Hence, ϕ is a dual polynomial satisfying $\sum_x \phi(x) f(x) = \varepsilon$. If $\varepsilon = 0$, we can take ϕ equal to the normalised parity function.

A.2 Proof of Theorem 10

In this case, we have the following linear programming problem:

$$\begin{aligned}
&\text{minimise} && \varepsilon && \\
&\text{subject to} && \sum_S \alpha_S \chi_S(x) \geq 1-\varepsilon && \text{for all } x \in X; && \text{(28a)}\\
&&& \sum_S \alpha_S \chi_S(x) \leq \varepsilon && \text{for all } x \in Y; && \text{(28b)}\\
&&& \sum_S \alpha_S \chi_S(x) \geq 0 && \text{for all } x \notin X; && \text{(28c)}\\
&&& \sum_S \alpha_S \chi_S(x) \leq 1 && \text{for all } x \notin Y; && \text{(28d)}\\
&&& \alpha_S \in \mathbb{R} && \text{for all } S \subseteq [n],\ |S| \leq d; &&\\
&&& \varepsilon \in \mathbb{R}. &&&&
\end{aligned}$$

Let us write the Lagrangian with the dual variables $a_x, b_x, c_x, d_x \geq 0$ for (28a) – (28d), respectively:

$$\begin{aligned}
\varepsilon \;-\; &\sum_{x\in X} a_x \Big(\sum_S \alpha_S \chi_S(x) - 1 + \varepsilon\Big) \;-\; \sum_{x\in Y} b_x \Big(\varepsilon - \sum_S \alpha_S \chi_S(x)\Big) \\
-\; &\sum_{x\notin X} c_x \Big(\sum_S \alpha_S \chi_S(x)\Big) \;-\; \sum_{x\notin Y} d_x \Big(1 - \sum_S \alpha_S \chi_S(x)\Big)
\end{aligned} \tag{29}$$

Let us define

$$\phi(x) = \begin{cases} a_x - d_x & \text{if } x \in X; \\ c_x - b_x & \text{if } x \in Y; \\ c_x - d_x & \text{if } x \notin X \cup Y. \end{cases}$$

Then, we can rewrite (29) as

$$\sum_{x\in X} a_x - \sum_{x\notin Y} d_x + \varepsilon\Big(1 - \sum_{x\in X} a_x - \sum_{x\in Y} b_x\Big) - \sum_S \alpha_S \Big(\sum_{x\in\{0,1\}^n} \phi(x)\chi_S(x)\Big). \tag{30}$$

Again, in the dual optimisation problem, all the brackets in (30) must be zero.

If ϕ satisfies (9), then we can take $a_x = \phi^+(x)$ for $x \in X$, $b_x = \phi^-(x)$ for $x \in Y$, $c_x = \phi^+(x)$ for $x \notin X$, and $d_x = \phi^-(x)$ for $x \notin Y$, and get a feasible solution to the dual.

For the opposite direction, consider optimal primal and dual solutions, whose values are equal by strong duality. We may assume $\varepsilon > 0$. By complementary slackness, for each x, at most one of the dual variables is non-zero, except for the case when $\varepsilon = 1/2$, in which case both a_x and b_x can be non-zero. Either way, we get $a_x = \phi^+(x)$ for $x \in X$, $b_x = \phi^-(x)$ for $x \in Y$, and $d_x = \phi^-(x)$ for $x \notin Y$. Thus we obtain the required dual formulation of Theorem 10.

Let us note that maximisation with 0 is required in (8). For example, consider the case $d = n-1$, and X and Y are of size 1. The function can be approximated by a polynomial of degree at most 1, thus $\varepsilon = 0$. On the other hand, by the second condition of (9), ϕ must be equal to a multiple of the parity function. It is easy to see that $\sum_{x\in X} \phi^+(x) - \sum_{x\notin Y} \phi^-(x)$ is actually negative in this case.